



## A distributed parameter identification problem in neuronal cable theory models

Jonathan Bell <sup>a</sup>, Gheorghe Craciun <sup>b,\*</sup>

<sup>a</sup> *Department of Mathematics and Statistics, University of Maryland, Baltimore County,  
Baltimore MD 21250, United States*

<sup>b</sup> *Mathematical Biosciences Institute, Ohio State University, 100 Mathematics Tower, 231 West 18th Avenue,  
Columbus OH 43210, United States*

Received 19 September 2003; received in revised form 22 June 2004; accepted 7 July 2004

---

### Abstract

Dendritic and axonal processes of nerve cells, along with the soma itself, have membranes with spatially distributed densities of ionic channels of various kinds. These ionic channels play a major role in characterizing the types of excitable responses expected of the cell type. These densities are usually represented as constant parameters in neural models because of the difficulty in experimentally estimating them. However, through microelectrode measurements and selective ion staining techniques, it is known that ion channels are non-uniformly spatially distributed. This paper presents a non-optimization approach to recovering a single spatially non-uniform ion density through use of temporal data that can be gotten from recording microelectrode measurements at the ends of a neural fiber segment of interest. The numerical approach is first applied to a linear cable model and a transformed version of the linear model that has closed-form solutions. Then the numerical method is shown to be applicable to non-linear nerve models by showing it can recover the potassium conductance in the Morris–Lecar model for barnacle muscle, and recover the spine density in a continuous dendritic spine model by Baer and Rinzel.

© 2004 Published by Elsevier Inc.

---

\* Corresponding author. Tel.: +1 614 688 0427; fax: +1 614 247 6643.  
E-mail address: [gcraciun@mbi.ohio-state.edu](mailto:gcraciun@mbi.ohio-state.edu) (G. Craciun).

MSC: 35; 92

*Keywords:* Cable theory; Inverse problems; Ion channel density; Distributed parameters; Variable conductance; Morris-Lecar model

---

## 1. Introduction

The excitable membrane of nerve and muscle cells have large numbers of ion channels whose gating (opening and closing) is typically voltage-dependent. These channels are ion specific and the dynamic interaction between these ionic channels plays a dominant role in the electrical activity of the cell. The transmission of signals through neuronal processes like dendrites and axons is classically modeled by cable theory. Conductance-based modeling through cable theory is just a current conservation equation of the form

$$C \frac{\partial v}{\partial t} + I_{\text{ion}} = \frac{\partial^2 v}{\partial x^2} \quad (1)$$

Since the time that Hodgkin and Huxley developed their famous model [1], this model formalism has been applied to a large number of excitable cells. This naturally leads to developing methods to estimate the model parameters. Due to the difficulty of taking a significant number of measurements along a neuronal fiber, uniform channel densities are assumed, and conductance parameters are estimated as constants. In this context there has been a fair amount of work to estimate model parameters, starting with the work of Rall in the 1960s [2,3]. Rall and colleagues developed ‘peeling’ methods for single fibers and ‘constrained optimization’ methods for coupled compartmental models for multiple neural cases [4–6]. Other methods have subsequently been described, among them are [7–13]. Also, [14] discusses the effect of noisy data on the estimates and proposes a technique that improves the robustness of the inverse mapping problem. Cox [13,15–17] developed a moment method he calls an ‘input impedance’ method to recover uniquely the (constant) membrane capacitance, conductance, fiber length, etc. Cox applied his input impedance method to the soma case [15], tapered and branched cells [13], as well as considering other inverse problems such as estimating current–voltage relations [16,17].

All the work referenced above is for estimating constant parameters. There is very little work, either experimental or theoretical, that obtains good estimates of spatially distributed cable parameters. Nevertheless, some experimental work motivates considering the problem of estimating spatially distributed ion channel densities, the subject of this paper. Johnston et al. [18] discussed studies of low-voltage activated and high-voltage activated conductance calcium channel types along the pyramidal dendrites in the mammalian hippocampus. Magee [19] investigated hyperpolarization-activated ( $I_h$ ) channels in CA1 hippocampal dendrites. Magee found that the density of these  $I_h$  channels increases six-fold from soma to distal dendrites. Traub et al. [20], based on experimental work from several labs, modeled the CA3 pyramidal cell dendrites using 19 compartments with six different active ionic channels having varying spatial distributions. See Johnston et al. [21] for a review of the non-uniform distributions of sodium and various calcium channel subtypes in some pyramidal cells, and Safronov [22] for a review of sodium and potassium channel distributions in spinal dorsal horn neurons.

In this paper we assume an individual ionic current can be isolated, with a non-uniform distribution of ionic channels along the fiber segment. By specifying physiologically reasonable extra boundary conditions, our objective is to efficiently recover an estimate for the variable conductance coefficient.

In the next section we formulate our first example of linear neural model with boundary conditions, then non-dimensionalize the model. In the third section we give a transformation of the model, useful in generating non-trivial examples with closed-form solutions. In section four we give our algorithm and discuss a number of results, first on a transformed version of our linear cable model, then on the nerve model itself. This will include recovering a continuous channel density function from an exact solution to our equation, along with considering a discontinuous channel density function. In section five we introduce a well-known non-linear model, the Morris–Lecar model for barnacle giant muscle, and show our algorithmic approach can reasonably recover the model’s potassium channel density, assuming it is spatially distributed. This can be done even when the density is discontinuous, as is shown. Then we consider a model of a dendrite with active spines. This model was originally formulated by Baer and Rinzel [23]. This is a higher dimensional non-linear model than the Morris–Lecar model, and our interest in section six is to use our algorithmic approach to recover the spatial distribution of spines. Finally, in the last section we discuss strengths and weaknesses of the methodology, its applicability to other situations, and the need for more theoretical foundations. We also give a uniqueness result in [Appendix A](#).

## 2. Formulation of the inverse problem for a linear cable model

First consider the classical case of Hodgkin–Huxley formulation of current conservation equation as given above, namely

$$C_m \frac{\partial v}{\partial t} + I_{\text{ion}}(v, \mathbf{u}) = \frac{a}{2R_i} \frac{\partial^2 v}{\partial x^2}. \tag{2}$$

Here  $R_i$ ,  $C_m$ ,  $a$  are, respectively, positive constants representing axoplasmic resistivity, membrane capacitance per unit area, and fiber radius. In the ionic current density  $I_{\text{ion}}$  each component of  $\mathbf{u}$  satisfies a first order ordinary differential equation in  $t$ . In the simplest case, the coefficients are voltage dependent, but other dependencies are possible. The exact form of the conductances are not of concern to us in this section since we impose sufficiently small stimuli to assume a linearized model for the current–voltage relation. In general,  $I_{\text{ion}}$  is a sum of current densities (sodium, calcium, chloride, etc.). Through pharmacological use of channel blockers, the dynamics of channels corresponding to a single ionic species can be studied. These assumptions lead to a significant reduction of the above model system to the scalar equation

$$C_m \frac{\partial v}{\partial t} + g(v - v_a) = \frac{a}{2R_i} \frac{\partial^2 v}{\partial x^2}. \tag{3}$$

Here  $v_a$  is a reversal potential, determined from the Nernst equation [24], but is treated below as a known, fixed parameter. Considering the domain to be  $0 < x < L$ ,  $0 < t < T$ , the appropriate boundary conditions of interest to us are

$$\frac{\partial v}{\partial x}(0, t) = -\frac{R_i}{\pi a^2} i_0(t), \quad \frac{\partial v}{\partial x}(L, t) = 0. \quad (4)$$

The first condition expresses the injection of a controlling current into the cell, and the second condition (a sealed-end condition) expresses that no longitudinal current escapes at  $x = L$ . We assume that the fiber terminates at  $x = L$ , or experimentally we set up conditions to block current flow at that end. For a discussion of cable modeling see, e.g., Tuckwell [25] or Jack et al. [26]. With an initial condition specified, and certain compatibility conditions given, the problem is solved by elementary means. This assumes all parameters are known and specified.

In this and the next section we consider the linear model above with known parameters, except for the single distributed channel density parameter  $N = N(x)$ . Using the notation of Hodgkin [27] we represent  $g = g_0 N$ ,  $C_m = C_0 N + C_1$ , where  $g_0$ ,  $C_0$ ,  $C_1$  are, respectively, the appropriate conductance per channel, the capacitance per channel, and the capacitance per unit area of membrane in the absence of channels. We will not be concerned with recovery or estimation of these parameters in this paper, but rather focus on the methodology for estimating  $N(x)$ . With a recording electrode, we may expect to measure the voltage at  $x = 0$ , even though the electrode is used to input the current stimulus. Is this enough information to estimate the channel density  $N$ ? How accurately can one determine  $N$ , given such boundary data? We address these questions below.

We substitute the above expressions for  $g$  and  $C_m$  into the equation, and let

$$\begin{aligned} \tilde{x} &= x \sqrt{\frac{2R_i C_1 g_0}{a C_0}}, & \tilde{t} &= t g_0 / C_0, \\ \tilde{v}(\tilde{x}, \tilde{t}) &= v(x, t) / v_a - 1, & q(\tilde{x}) &= C_0 N(x) / C_1. \end{aligned} \quad (5)$$

Then (3), (4) become

$$\begin{aligned} (1 + q) \frac{\partial \tilde{v}}{\partial \tilde{t}} + q \tilde{v} &= \frac{\partial^2 \tilde{v}}{\partial \tilde{x}^2}, & 0 < \tilde{x} < l, & \quad 0 < \tilde{t} < \tilde{T}, \\ v(0, \tilde{t}) &= 0, & \frac{\partial \tilde{v}}{\partial \tilde{x}}(0, \tilde{t}) &= -i(\tilde{t}), & \frac{\partial \tilde{v}}{\partial \tilde{x}}(l, \tilde{t}) &= 0, \end{aligned}$$

where now

$$l = L \sqrt{\frac{2a R_i C_0}{2C_1 g_0}}, \quad \tilde{T} = T g_0 / C_0, \quad i(\tilde{t}) = \frac{1}{\pi a^2 v_a} \sqrt{\frac{a R_i C_0}{2C_1 g_0}} i_0(\tilde{t} C_0 / g_0).$$

We assume that the original initial condition was the equilibrium state  $v(x, 0) = v_a$ , for all  $0 < x < L$ . Measuring the potential at the origin yields

$$\tilde{v}(0, \tilde{t}) = \frac{v(0, t)}{v_a} - 1 = \tilde{f}(\tilde{t}).$$

Dropping all the tildes in the notation, the inverse problem we consider is

$$(1 + q(x)) \frac{\partial v}{\partial t} = \frac{\partial^2 v}{\partial x^2} - q(x)v, \quad 0 < x < l, \quad 0 < t < T, \quad (6)$$

$$v(x, 0) = 0, \quad 0 < x < l, \tag{7}$$

$$\frac{\partial v}{\partial x}(0, t) = -i(t), \quad \frac{\partial v}{\partial x}(l, t) = 0, \quad 0 < t < T \tag{8}$$

with the extra measurements given by

$$v(0, t) = f(t). \tag{9}$$

While there is a considerable literature on inverse problems for parabolic equations (see, e.g., [28–30]), much of the work deals with recovery of source terms, or diffusion coefficients, or time-dependent only coefficients (and in some cases, initial conditions). The challenge of the problem (6)–(8) is to recover a spatially distributed coefficient from overspecified temporal data.

### 3. A transformation of the problem

A uniqueness result for the problem (6)–(8) is given in the [Appendix A](#). For the time being let us consider the more general case, namely

$$(1 + q(x)) \frac{\partial v}{\partial t} = \frac{\partial^2 v}{\partial x^2} - q(x)v, \quad 0 < x < l, \quad 0 < t < T, \tag{6}$$

$$v(x, 0) = v_0(x), \quad 0 < x < l, \tag{10}$$

$$v(0, t) = f_0(t), \quad \frac{\partial v}{\partial x}(0, t) = f_1(t), \quad 0 < t < T, \tag{11}$$

$$v(l, t) = g_0(t), \quad \frac{\partial v}{\partial x}(l, t) = g_1(t), \quad 0 < t < T. \tag{12}$$

If we let  $p(x) = 1 + q(x)$  and  $v(x, t) = p^{-1/4}u(s, t)$ , where  $\frac{ds}{dx} = p^{1/2}$ ,  $s(0) = 0$ , and substitute these into (6), (10)–(12), we obtain

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial s^2} - Q(s)u, \quad 0 < s < s_r, \quad 0 < t < T, \tag{13}$$

$$u(s, 0) = p^{1/4}(x(s))v_0(x(s)), \quad 0 < s < s_r, \tag{14}$$

$$u(0, t) = \eta_0(t) = f_0(t)p_0^{1/4}, \quad \frac{\partial u}{\partial s}(0, t) = \eta_1(t) = f_1(t)p_0^{-1/4} - \gamma_0 f_0(t), \tag{15}$$

$$u(s_r, t) = \mu_0(t) = g_0(t)p_l^{1/4}, \quad \frac{\partial u}{\partial s}(s_r, t) = \mu_1(t) = g_1(t)p_l^{-1/4} - \gamma_l g_0(t), \tag{16}$$

where  $p_0 = p(0)$  and  $\gamma_0 = -p'(0)/4p_0^{3/2}$  with analogous expressions for  $p_l, \gamma_l$ .

Here  $s_r = \int_0^l \sqrt{p(x)}dx$ , which can only be determined by obtaining  $p$ . In (13)  $Q$  needs to satisfy  $Q = (\alpha q - \frac{\partial^2 q}{\partial x^2})/\alpha\beta^2$ , where  $\alpha = p^{-1/4}$ ,  $\beta = p^{1/2}$ . An equivalent way to write this expression in terms of  $p$  is

$$p \frac{d^2 p}{ds^2} - \frac{3}{4} \left( \frac{dp}{ds} \right)^2 + 4p(p-1) - 4p^2 Q(s) = 0 \quad (17)$$

along with boundary (or initial) conditions. This implies we need some information about the channel density at the ends of our neural segment, i.e., at  $x=0, l$ . We will assume even for our method developed below that we have such information.

We are actually interested in a method that will also be applicable to more experimentally relevant, non-linear versions of these equations, as we describe in Sections 5 and 6. In general, previous numerical methods of solving (13)–(16) are tailored specifically for the linear case, and do not seem to have any immediate applications for more general, non-linear versions of these equations.

Employing a trick used in Danilaev [31], since  $Q(s) = (\frac{\partial^2 u}{\partial s^2} - \frac{\partial u}{\partial t})/u$  is independent of  $t$ , then  $\frac{\partial Q}{\partial t} = 0$ , which implies  $\frac{\partial}{\partial t} \left\{ \left( \frac{\partial^2 u}{\partial s^2} - \frac{\partial u}{\partial t} \right) / u \right\} = 0$ , or

$$u \frac{\partial^3 u}{\partial s^2 \partial t} - u \frac{\partial^2 u}{\partial t^2} - \frac{\partial u}{\partial t} \frac{\partial^2 u}{\partial s^2} + \left( \frac{\partial u}{\partial t} \right)^2 = 0. \quad (18)$$

This calls for a number of observations. Eq. (17) has at least one two-parameter family of closed-form solutions. This can be seen by writing (17) as

$$p \frac{d^2 p}{ds^2} - \frac{3}{4} \left( \frac{dp}{ds} \right)^2 + kp^2 = 4p^2 Q - 4p(p-1) + kp^2 \quad (19)$$

for any constant  $k > 0$ . If the right-hand side of (19) is zero, then the homogeneous equation is solvable, and this gives  $p(s) = p_0 \cos^4(-\frac{\sqrt{k}}{2}s + C_1)$  for some constants of integration  $p_0, C_1$ . The zero right-hand side gives  $Q = 1 - k/4 - 1/p$ . For our purposes let  $k=4$  so that  $p(s) = p_0 \cos^4(C_1 - s)$  and  $Q(s) = -1/P_0 \cos^4(C_1 - s)$ . As an example to be used in a numerical simulation below, on the interval  $0 < s < \pi/2$ , we can model increasing (respectively, decreasing) density,  $p/p_0$ , by setting  $C_1 = 1$  (respectively,  $C_1 = 0$ ). If, for example, we let  $C_1 = 0$  and  $p_0 = 4$ , so that  $p(s) = 4 \cos^4(s)$ , then  $\frac{dx}{ds} = 2 \cos^2(s)$  or  $s = \tan^{-1}(2x)$  and therefore  $p(x) = 4 \cos^4(\tan^{-1}(2x))$  on  $0 < x < \frac{1}{2}$ . This gives  $p(x) \in (1, 4)$ , guaranteeing the physical situation of  $q(x) \geq 0$  on  $[0, 1/2]$ . Then (13) becomes

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial s^2} + \frac{1}{4} \sec^4(s) u. \quad (20)$$

Note that we have two particular classes of solutions to (20) worth mentioning. If one separates variables, e.g.,  $u(s, t) = e^{-t} y(s)$ , then  $y$  must satisfy

$$\frac{d^2 y}{ds^2} + \left( \frac{1}{4} \sec^4(s) + 1 \right) y = 0, \quad (21)$$

which has solutions  $y(s) = y_1(s) = \cos(s) \sin(\frac{1}{2} \tan(s))$  and  $y(s) = y_2(s) = \cos(s) \cos(\frac{1}{2} \tan(s))$ . In fact, for  $p(s) = p_0 \cos^4(\frac{\sqrt{k}}{2}s - C_1)$ , the function

$$u(s, t) = e^{-t} \cos \left( \frac{\sqrt{k}}{2} s - C_1 \right) \sin \left( \frac{2}{\sqrt{k p_0}} \tan \left( \frac{\sqrt{k}}{2} s - C_1 \right) \right)$$

is a solution of Eq. (13), along with its companion expression with sin replaced by cos. Notice that  $u_1 = e^{-t}y_1$  satisfies  $u_1(0, t) = 0$ ,  $\frac{\partial u_1}{\partial s}(0, t) = \frac{1}{2}e^{-t}$ , while  $u_2 = e^{-t}y_2$  satisfies  $u_2(0, t) = e^{-t}$ ,  $\frac{\partial u_2}{\partial s}(0, t) = 0$ . In fact, for any  $C^2$  function  $\varphi(s)$ , and constant  $K$ ,  $u(s, t) = e^{Kt}\varphi(s)$  is a classical solution to (18).

Another approach is to make the reduction-of-order type change of variable  $w = \frac{\partial u}{\partial t} / u$  to Eq. (18) to obtain the (non-local) convection-diffusion equation

$$\frac{\partial w}{\partial t} = \frac{\partial^2 w}{\partial s^2} + 2 \left( \int_0^t \frac{\partial w}{\partial s} dt \right) \frac{\partial w}{\partial s}. \tag{22}$$

This also gives

$$Q(s) = \frac{\frac{\partial^2 u}{\partial s^2} - \frac{\partial u}{\partial t}}{u} = \left( \int_0^t \frac{\partial w}{\partial s} dt \right)^2 + \int_0^t \frac{\partial^2 w}{\partial s^2} dt - w(s, t). \tag{23}$$

A challenge to transforming the original problem is determining the right hand boundary,  $s_r$ , which introduces an extra error into the numerical procedures, and having to deal with the unknown initial condition, if the original initial condition is not zero everywhere. Hence, it is best to develop a method that is applicable to the original problem (6)–(9), and can be extended to non-linear problems.

#### 4. The algorithm for $q(x)$ and numerical results

We propose a new numerical procedure to solve the inverse problems described in the previous section. We will first describe our numerical procedure for the simplest type of inverse problem discussed previously, namely

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} - qu, \tag{24}$$

where  $u = u(x, t)$  and  $q = q(x)$ .

We suppose we know the initial condition  $u(x, 0)$  for  $0 < x < L$  and boundary conditions  $u(0, t)$  and  $\frac{\partial u}{\partial t}(0, t)$  for  $0 < t < t_{\max}$ . We want to compute  $q(x)$  for  $0 < x < L$ . Consider some positive integer  $n$ , and an  $n \times n$  uniform grid for the domain  $[0, L] \times [0, t_{\max}]$ . Denote by  $u_i^m$  and  $q_i$ , the values of  $u(0, t)$  and  $q(x)$  on the discrete grid, where we have  $1 \leq m, i \leq n$ ,  $\Delta x = L/n$ , and  $\Delta t = t_{\max}/n$ .

Consider the usual explicit finite difference scheme for Eq. (24)

$$\frac{u_i^{m+1} - u_i^m}{\Delta t} = \frac{u_{i+1}^m + u_{i-1}^m - 2u_i^m}{(\Delta x)^2} - q_i u_i^m. \tag{25}$$

Then, according to (25) we have

$$q_i = \frac{1}{u_i^m} \left( -\frac{u_i^{m+1} - u_i^m}{\Delta t} + \frac{u_{i+1}^m + u_{i-1}^m - 2u_i^m}{(\Delta x)^2} \right), \tag{26}$$

assuming  $u_i^m \neq 0$ . Also according to (25) we have

$$u_{i+1}^m = -u_{i-1}^m + 2u_i^m + (\Delta x)^2 \left( \frac{u_i^{m+1} - u_i^m}{\Delta t} + q_i u_i^m \right). \tag{27}$$

Since we know  $u(x, 0)$ ,  $u(0, t)$  and  $\frac{\partial u}{\partial x}(0, t)$ , we shall suppose we know  $u_i^1$ ,  $u_1^m$  and  $u_2^m$ , for all  $1 \leq m$ ,  $i \leq n$ .

The algorithm we use to compute  $q_2, q_3, \dots, q_{n-1}$  is the following:

*Step 1.* First, compute  $q_2$  from (26), since we know  $u_1^1$ ,  $u_2^1$ ,  $u_3^1$ , and  $u_2^2$ ; then, use (27) and the value of  $q_2$  just computed, to compute  $u_3^m$  for  $2 \leq m \leq n - 1$ .

*Step 2.* First, compute  $q_3$  from (26), since we know  $u_2^1$ ,  $u_3^1$ ,  $u_4^1$ , and  $u_3^2$ ; then, use (27) and the value of  $q_3$  just computed, to compute  $u_4^m$  for  $2 \leq m \leq n - 2$ .

...

*Step i.* First, compute  $q_{i+1}$  from (26), since we know  $u_i^1$ ,  $u_{i+1}^1$ ,  $u_{i+2}^1$ , and  $u_{i+1}^2$ ; then, use (27) and the value of  $q_{i+1}$  just computed, to compute  $u_{i+2}^m$  for  $2 \leq m \leq n - i$ .

...

*Step n - 2.* First, compute  $q_{n-1}$  from (26), since we know  $u_{n-2}^1$ ,  $u_{n-1}^1$ ,  $u_n^1$ , and  $u_{n-1}^2$ ; then, use (27) and the value of  $q_{n-1}$  just computed, to compute  $u_n^m$  for  $2 \leq m \leq n - (n - 2)$ .

Note that, in the process of computing  $q_2, q_3, \dots, q_{n-1}$  using the algorithm above, we end up also knowing  $u_i^m$  for all  $1 \leq m \leq n - i + 1$  and all  $1 \leq i \leq n$ . We will now present some concrete examples for the use of this algorithm.

**Example 1.** As we saw in the previous section, the function

$$u(x, t) = e^{-t} \cos(x) \sin\left(\frac{1}{2} \tan(x)\right)$$

is a solution of (24) for

$$q(x) = -\frac{1}{4} \sec^4(x).$$

Consider a  $n \times n$  grid of the  $(x, t)$  domain  $[0, 1] \times [0, 1]$ . The  $i$ th grid point on the  $x$  axis is  $x = \frac{i-1}{n}$ , and the  $m$ th grid point on the  $t$ -axis is  $t = \frac{m-1}{n}$ .

The algorithm described above can be used to compute an approximation of  $q_i = -\frac{1}{4} \sec^4(\frac{i-1}{n})$ , for  $2 \leq i \leq n-1$ . The only input data are the initial condition  $u_i^1 = \cos(\frac{i-1}{n}) \sin(\frac{1}{2} \tan(\frac{i-1}{n}))$ , and the boundary conditions  $u_1^m = e^{-\frac{m-1}{n}}$ ,  $u_2^m = u_1^m + \frac{1}{2} e^{-\frac{m-1}{n}}$ . The expressions for  $u_i^1$  and  $u_1^m$  are obtained by plugging into the formula for  $u(x, t)$  above, and the expression for  $u_2^m$  is just  $u_i^m + \frac{\partial u}{\partial x}(0, \frac{m-1}{n}) \Delta t$ . We would like to investigate how robust our algorithm is if we add random noise terms to the input data. In Fig. 1 we show the logarithmic scale error obtained using our algorithm on a  $100 \times 100$  grid (i.e.,  $n = 100$ ). We initialized  $u_i^1$ ,  $u_1^m$  and  $u_2^m$ , for all  $1 \leq i, m \leq 100$ , using the formula above, and then added some random noise terms. We use *relative* noise in all our computations (i.e., a percentage of the exact value), to make the noise value independent of scaling.



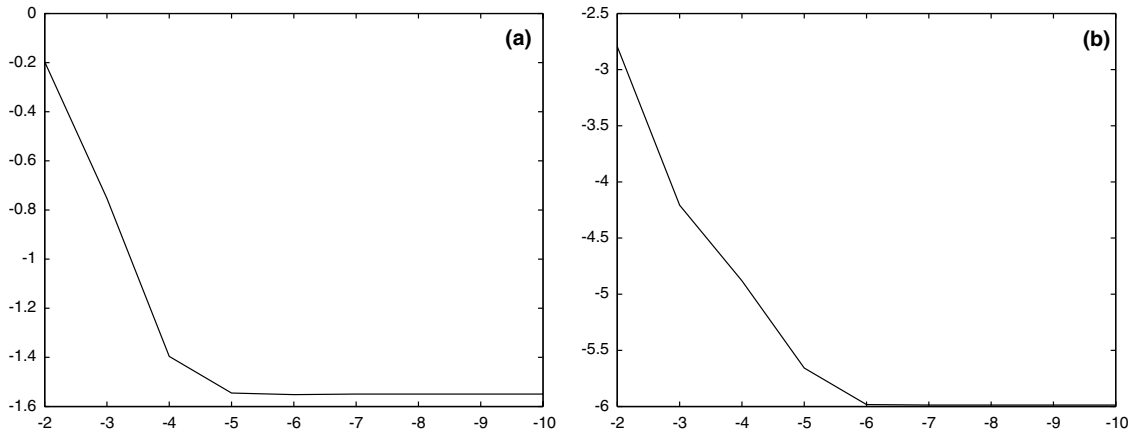


Fig. 1. Logarithmic scale error for solutions of the inverse problem (24): (a) for  $q(x)$ , (b) for  $u(x, t)$ . The horizontal axis shows  $\log_{10}$  of the amplitude of the noise, and the vertical axis shows  $\log_{10}$  of the maximum error in the solution, as a fraction of the true value.

More exactly, we replaced each  $u_i^1$  by  $(1 + \varepsilon_1)u_i^1$ , each  $u_1^m$  by  $(1 + \varepsilon_2)u_1^m$ . Then we updated each  $u_2^m$  to keep the difference  $u_2^m - u_1^m$  noise-free, since this difference corresponds to the stimulus current  $i(t)$ , and we have good control on  $i(t)$  in an experimental setting. The noise terms  $\varepsilon_1, \varepsilon_2$  are independent and uniformly distributed on some interval  $[-a, a]$ . We refer to  $a$  as the *amplitude* of the noise.

We did experiment with more general types of noise, e.g., distributed noise: independent noise terms  $\varepsilon_i^1, \varepsilon_1^m, \varepsilon_2^m$  for each  $u_i^1, u_1^m, u_2^m$ . Note that in this case we do not impose that the difference  $u_2^m - u_1^m$  is noise-free. The shape of the logarithmic scale error plot in the case of distributed noise is similar to Fig. 1, i.e., as we decrease the noise level, the error decreases at about the same rate as in the case of constant noise, and to the same plateau value. On the other hand, for large noise level, the error in the case of distributed noise is significantly larger than in the case of constant noise, especially if  $n$  is also large. For example, for large noise level and  $n = 20$ , the error is about one order of magnitude larger in the case of distributed noise than in the case of constant noise, while for large noise level and  $n = 100$  the error is about two orders of magnitude larger. For large distributed noise level the total error can be reduced by averaging over many runs. For example, by averaging over 20 runs of the algorithm, we can consistently compute a good approximation of  $q$  for 0.1% distributed noise level. Also, the error level does not increase as  $x$  increases, although, as it marches to the right, the algorithm is exposed to more and more noise in this case.

Note that the approximation results for  $u(x, t)$  scale very well with the error rate (see Fig. 1(b)). The approximation results for  $q(x)$  in Fig. 1(a) also scale well with the error rate; note though that the error curve for  $q$  becomes flat for error less than  $10^{-5}$  (becomes about  $-1.549$ ), since we reached the level of the error of approximating  $q(x)$  by using the *exact* input values. Indeed, the error of approximating  $q(x)$  by the  $q_i$ , obtained from formula (26), if we use the exact input values, for  $n = 100$ , is about 2.82%; on the other hand  $10^{-1.549} \approx 0.0282$ . This plateau value can be lowered by increasing  $n$ .

If we look at the case where the input data has large noise level, but more than one input dataset is available, we can basically cancel out the error by averaging the results. In Fig. 2 we show

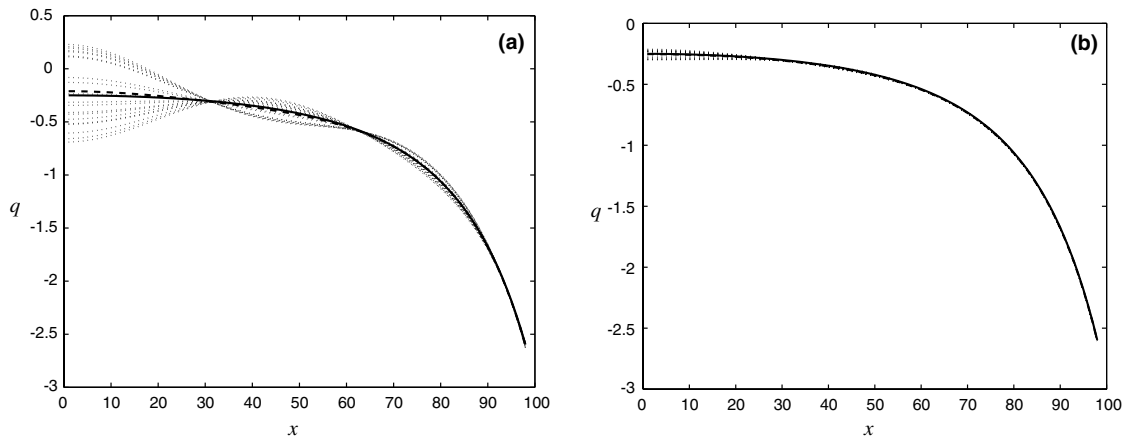


Fig. 2. The dotted curves are 20 computed versions of  $q$  for the inverse problem (24), for: (a) 1% and (b) 0.1% random input noise. The thick dashed line is their average, and the thick solid line is the exact graph of  $q$ .

results for 1% and 0.1% input noise. In Fig. 2(a) there are 20 curves that approximate  $q$  for 1% input error, as explained before, obtained by applying our algorithm 20 times, for new error terms  $\varepsilon_1, \varepsilon_2$  each time. Although most of these approximations are not very accurate, their average (the thick dashed curve) is a good approximation of the correct  $q$  (the solid curve). It makes little difference if we apply the algorithm more than 20 times, and averaging just 10 times gives good error canceling results. Note also that for 0.1% input noise the approximation is good even without averaging (see Fig. 2(b)).

**Example 2.** We compare results of our method with results of Tadi et al. [32] for discontinuous  $q(x)$ . Given the sparse conductance measurements of nerve fiber studies which lead to histograms of the channel density distributions, considering a piecewise constant  $q$  is appropriate. We do not have any analytic solutions for discontinuous  $q(x)$ . Therefore, for testing purposes, we first use classical numerical procedures (e.g., an explicit scheme, or Crank–Nicholson) to solve the direct problem, i.e., to compute  $u_i^m$  for a given  $q_i$ , and for given initial and boundary values. For simplicity, consider again the domain  $(x,t) \in [0, 1] \times [0, 1]$ . Let us define  $u(x,t) \equiv 1$  if  $t = 0$  or  $x = 0$  or  $x = 1$ . Also, let us divide the  $x$  domain into six equal subintervals, and define  $q$  to be piecewise constant and taking values 0, 3, 1, 2, 0, on these six intervals, in this order ( $q$  is the piecewise constant function in Fig. 3(a)).

Given this  $q$  and these initial and boundary conditions for  $u$  we can solve Eq. (24) on a uniform  $n \times n$  grid, using some classical numerical procedure, e.g., the first-order explicit scheme, or the Crank–Nicholson scheme. We end up knowing  $u_i^m$  for all  $1 \leq i, m \leq n$ . Then we can input only  $u_i^1, u_1^m$  and  $u_2^m$  (and no information about  $q$ ) into the algorithm we described above to recover  $q_i$ , for  $2 \leq i \leq n - 1$ . Note that when we use the explicit scheme we might have to subdivide into more time intervals than space intervals to ensure convergence. Suppose we have  $k$  times as many time intervals as space intervals. Then, in order to input the results into our algorithm, we keep only every  $k$ th column, since our algorithm requires that we work with a square matrix  $u_i^m$ . This is not an issue if we use unconditionally convergent schemes, e.g., the Crank–Nicholson scheme.

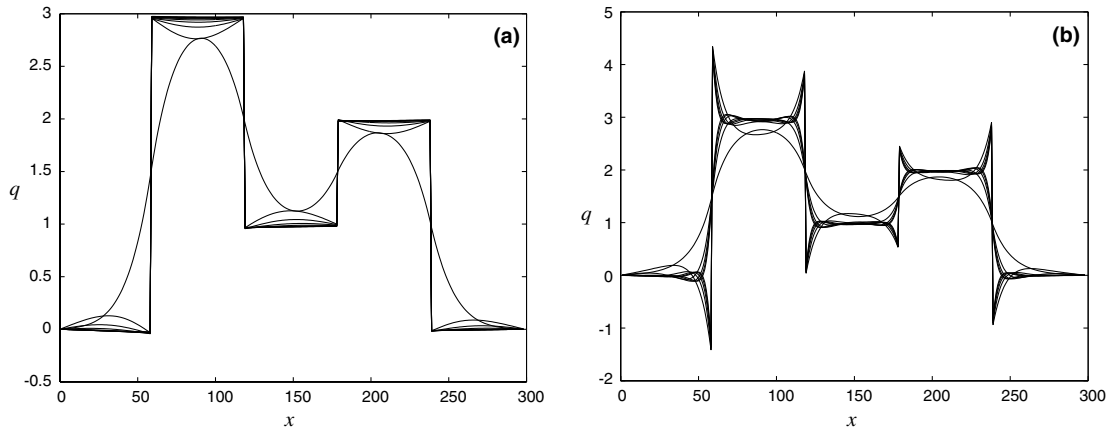


Fig. 3. Error introduced by classical numerical procedures for the direct problem (24), in the region of the domain near  $t = 0$ : (a) by the explicit scheme and (b) by the Crank–Nicholson scheme. The function  $q$  is piecewise constant.

Before we go on and use our algorithm to compute an approximation of the piecewise constant function  $q$ , we should consider what is the best possible result we can hope for, based on input data  $u_i^1, u_1^m, u_2^m$  obtained from e.g., the explicit scheme, or the Crank–Nicholson scheme. Recall that even in Example 1, where we had an exact solution, there was a plateau in the approximation error, due not to the performance of our algorithm, but to the discretization of the data.

We use  $n = 300$  for all computations in this example. In Fig. 3, we show that the explicit scheme and the Crank–Nicholson scheme do indeed introduce a significant error, before we use our algorithm. For example, to obtain Fig. 3(a) we did the following: First, we applied the explicit scheme for the boundary and initial conditions described above, and obtained  $u_i^m$  for all  $1 \leq i, m \leq n$ . Then, we applied formula (26) to these values  $u_i^m$  for each fixed  $m = 1, 2, \dots, n$ . For each fixed  $m$  we obtained one curve that approximates  $q$ . For  $m = 1$  we actually obtained the worse such approximation, the smooth curve in Fig. 3(a). Then, the approximation became better as we increased  $m$ . On the other hand, it is exactly the value  $m = 1$  that we use in our algorithm whenever we compute  $q_i$ .

Fig. 3(b) is completely analogous, but for the Crank–Nicholson scheme. Note that, overall, the approximation is much better in Fig. 3(a). This suggests that input data obtained using the explicit scheme is a better input for our algorithm.

Of course, we cannot expect to get a better approximation of  $q(x)$  from our algorithm than there is in the input data. But, if we use variable  $\Delta t$ , i.e., smaller near  $t = 0$  than near  $t_{\max}$ , and use the explicit scheme output as input into our algorithm, we obtain the results in Fig. 4. We also added 0.01% noise to the input data, in a manner similar to Example 1. While the methods of Tadi et al. [32] also can localize the ‘bumps’ of  $q$ , the approximations in [32] are more diffusive, and the height of the approximation ‘bumps’ is often only about 75% of the correct value (see Figs. 4, 5, 8 and 9 in [32]). Also, the algorithm described in [32] requires nested iterations of solving a transformed problem, which is a higher order integral PDE with complicated boundary conditions, thus more difficult to implement and much more time intensive than our algorithm.

If we avoid the region near  $t = 0$ , e.g., we use input values  $u_i^{20}, u_{21}^m, u_{22}^m$  instead of  $u_i^1, u_1^m, u_2^m$ , we get good approximations for  $q(x)$ , even for multiple ‘bumps’, as we can see in Fig. 5.

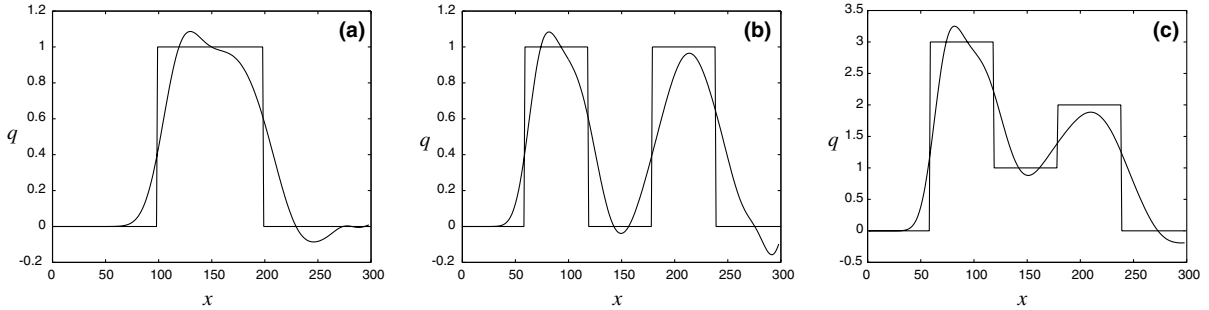


Fig. 4. Results for variable  $\Delta t$  for the inverse problem (24): (a)  $q(x)$  has one ‘bump’; (b)  $q(x)$  has two ‘bumps’ at some distance from each other and (c)  $q(x)$  has three ‘bumps’ of different sizes.

Note that the results we get in this case are very accurate approximations of the best possible results we can hope for, according to Fig. 3.

**Example 3.** Recall that Eq. (24) is a simplification of the type of equations derived in Section 2. From neural applications we would actually like to be able to solve the inverse problem for Eq. (6). On the other hand, it is easy to see that the same explicit finite difference scheme we described for Eq. (24) at the beginning of this section can be used for Eq. (6), and exactly the same algorithm can be applied to compute  $q_2, q_3, \dots, q_{n-1}$  for Eq. (6).

To illustrate the use of our algorithm in this case, we proceed as in Example 2: we first use some classical numerical procedure to solve Eq. (6) for the same initial and boundary conditions as in Example 2. Then, we input the computed  $u_i^1, u_i^m,$  and  $u_i^m$  into our algorithm to recover  $q_i$  from the explicit finite difference scheme for Eq. (6). The results are shown in Fig. 6. Again, we avoided the region of the domain near  $t = 0$ , where the classical numerical procedures used to produce input data introduce large error.

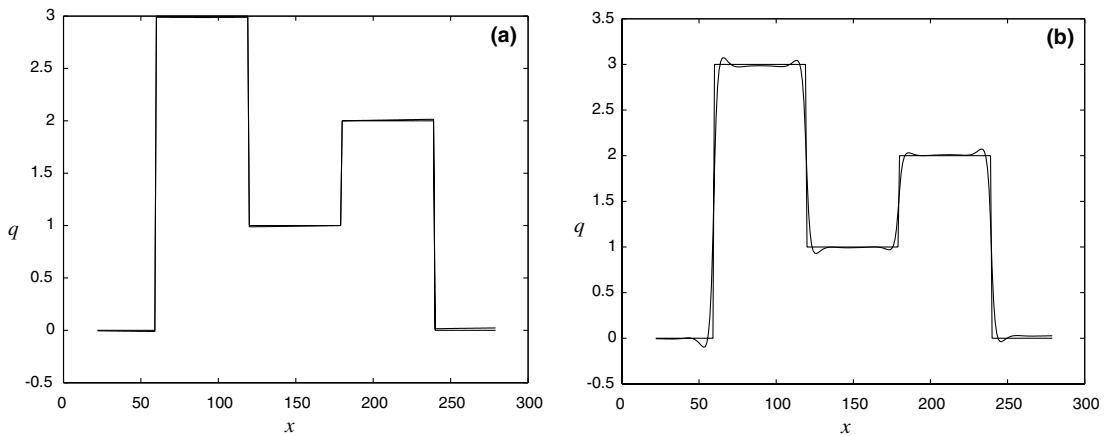


Fig. 5. Our numerical procedure gives accurate results for input obtained with classical numerical procedures for Eq. (24), if we avoid the region of the domain near  $t = 0$ , where the classical numerical procedures introduce error. We show 20 computed versions of  $q$ , each with 0.01% random noise, for input data obtained using: (a) the explicit scheme and (b) the Crank–Nicholson scheme.

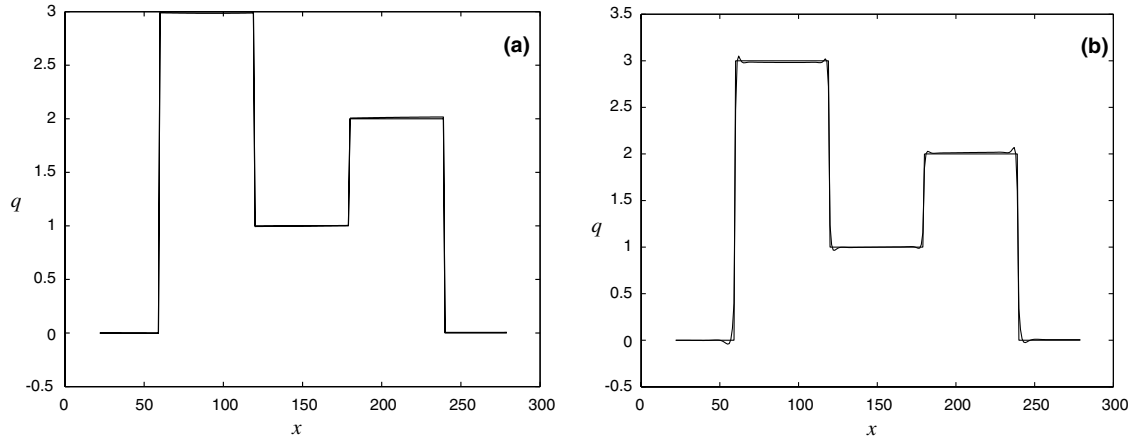


Fig. 6. Results for input obtained with classical numerical procedures, if we avoid the region of the domain near  $t = 0$ , for Eq. (6). We show 20 computed versions of  $q$ , each with 0.01% random noise, for input data obtained using: (a) the explicit scheme and (b) the Crank–Nicholson scheme.

### 5. Non-linear example: the Morris–Lecar model

Morris and Lecar [33] developed a model for barnacle giant muscle which has become a popular two-variable non-linear model to compute certain prototypical behavior, like regular trains of impulses. Their model incorporates a non-inactivating fast calcium current, non-inactivating slower potassium current, and a small leak current. The simulations of their model provided reasonable agreement with experimental measurements. The form of their model is

$$C_m \frac{\partial v}{\partial t} + \bar{g}_{Ca} m_\infty(v)(v - v_{Ca}) + \bar{g}_K w(v - v_K) + \bar{g}_L(v - v_L) = \frac{a}{2R} \frac{\partial^2 v}{\partial x^2}, \tag{28}$$

$$\frac{\partial w}{\partial t} = \varphi \{w_\infty(v) - w\} / \tau_w(v), \tag{29}$$

where the calcium activation variable,  $m$ , is fast enough to set it to the quasi-steady state function  $m_\infty(v) = 1 / \{1 + \exp[-(v + 1.2)/9]\}$ . For the potassium activation variable,  $w_\infty(v) = 1 / \{1 + \exp[-(v - 2)/15]\}$  and  $\tau_w(v) = 1 / \cosh[(v - 2)/30]$ . For other parameter values we pulled values from Table 2.4, p. 35 of Fall et al. [24].

For numerical simulations we non-dimensionalize the model analogous to the previously studied linear model. We arbitrarily pick the potassium conductance to be spatially variable, and take the characteristic voltage  $V_{char} = 100$  [mV]. We let  $v = v_K + V_{char} \tilde{v}$ ,  $\tilde{x} = x/\lambda$ ,  $\tilde{t} = t/\tau$ , and  $C_m = C_0 N + C_1$ ,  $g_k = g_k^* N$ . Then  $\tau = C_0 / g_k^*$ ,  $\lambda^2 = \frac{a}{2R} \frac{C_0}{C_1 g_k^*}$ , and  $q = C_0 N / C_1$ . This scaling also leads to definitions of the voltage-dependent functions of the form  $\tilde{m}_\infty(\tilde{v}) = m_\infty(v_K + V_{char} \tilde{v})$ , and analogous expressions for  $\tilde{w}_\infty$ ,  $\tilde{\tau}_w$ . With appropriate definitions for the scaled reversal potentials and maximum conductance values, the non-dimensionalized Morris–Lecar model (after dropping the tildes to make the notation manageable) becomes

$$(1 + q(x)) \frac{\partial v}{\partial t} + \bar{g}_{Ca} m_{\infty}(v)(v - v_{Ca}) + q(x)w(v - v_K) + \bar{g}_L(v - v_L) = \frac{\partial^2 v}{\partial x^2}, \quad (30)$$

$$\frac{\partial w}{\partial t} = \varphi_0 \{(w_{\infty}(v) - w)/\tau_w(v)\}, \quad (31)$$

where  $\varphi_0 = \varphi\tau$ . Finally, the boundary conditions take the form

$$v(x, 0) = v_0(x), \quad \frac{\partial v}{\partial x}(0, t) = -\alpha i(t), \quad \frac{\partial v}{\partial x}(l, t) = 0, \quad \left( \alpha = \frac{\lambda R}{\pi a^2 V_{\text{char}}} \right) \quad (32)$$

$$v(0, t) = f(t). \quad (33)$$

So now the problem is to recover a reasonable estimate of  $q$  in (30) and (31) by applying (32) and the measured data (33), and using the methodology developed in Section 4. Again we assume all parameters are known except  $q(x)$ .

Note that one can easily express  $q$  in terms of  $v$ ,  $\frac{\partial v}{\partial t}$ ,  $\frac{\partial^2 v}{\partial x^2}$  and  $w$ , using Eq. (30). Let us denote by  $q_i$ ,  $v_i^m$ ,  $w_i^m$  the discretized versions of  $q(x)$ ,  $v(x, t)$ ,  $w(x, t)$  for the same type of uniform  $n \times n$  numerical grid as in Section 4. We can use Eq. (30) to compute  $q_2$ , given the initial conditions for  $v$  and  $w$ , and given boundary conditions (32) for  $v$  at  $x = 0$ . Then, we can use Eq. (31) to compute  $w_2^m$ , and Eq. (30) to compute  $v_3^m$ . After that we can use Eq. (30) to compute  $q_3$ , and so on. In other words, we can use exactly the same idea as in Section 4 to solve for  $q_i$ ,  $v_i^m$ ,  $w_i^m$ . Some results of applying this algorithm appear in Fig. 7. As in a previous example, we used simple initial and boundary conditions  $u(x, t) \equiv 1$  if  $t = 0$ , or  $x = 0$ , or  $x = l$ , where  $l = 8$ . The way we add noise to the data is similar to Example 1 in Section 4: we add independent noise to all the input terms  $v_i^1$ ,  $w_i^1$ ,  $v_1^m$ ,  $v_2^m$ , except that we keep the difference  $v_2^m - v_1^m$  noise-free (since this difference corresponds to the stimulus current that we control in an experimental setting). On the other hand, as we mentioned before, in none of our numerical experiments is this special restriction necessary in order to get the type of numerical results described.

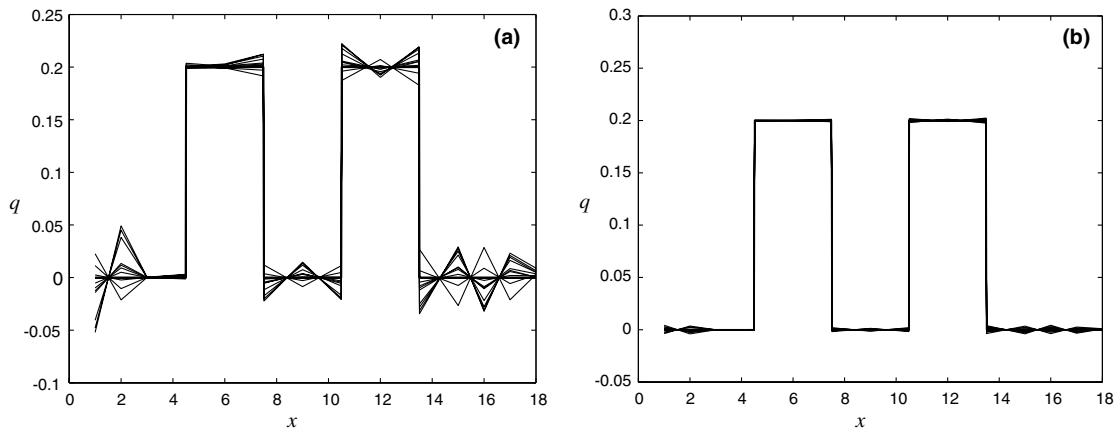


Fig. 7. Results for the Morris–Lecar model: (a) for 1% noise level and (b) for 0.1% noise level. The thin lines are the computed solutions, the thick line is the exact solution.

### 6. Recovery of dendritic spine distribution: the Baer–Rinzel model

Baer and Rinzel [23] formulated a (continuous dendritic spine) cable theory to investigate electrical interactions between many excitable dendritic spines. In their model the spine density, membrane potential in the spine heads, and spine stem current vary continuously in space and time, but the spines interact only indirectly by voltage spread along a passive dendritic cable. The active membrane in the spine heads is modeled using Hodgkin–Huxley kinetics. The area of the spine head is estimated in [23] to be  $1.31 \mu\text{m}^2$ , and the conductance is assumed to be 12 pS for a single sodium channel, which implies 328  $\text{Na}^+$  channels per spine head. A similar calculation could be done for the potassium channels. Rather than using this model to recover some specific spatially distributed ionic channel density, we demonstrate our algorithm on the model to recover a non-uniform distribution of spines. We will use the same overdetermined left and right boundary conditions for this model though this is a bit artificial way of stimulating a spine-laden dendrite. Alternative possibilities can be explored in a future study. The Baer–Rinzel model is then

$$C_m \frac{\partial v_d}{\partial t} = \frac{a}{2R_i} \frac{\partial^2 v_d}{\partial x^2} - \frac{v_d}{R_m} + \frac{\bar{N}}{2\pi a} \frac{v_{sh} - v_d}{R_{ss}}, \tag{34}$$

$$C_{sh} \frac{\partial v_{sh}}{\partial t} + I_{ion}(v_{sh}, m, n, h) = \frac{v_d - v_{sh}}{R_{ss}}.$$

Again, these are current conservation equations for the dendritic potential,  $v_d$ , and the spine head potential,  $v_{sh}$ . The quantities  $m, n, h$  in (34) are gating variables satisfying dynamics of the form

$$\frac{\partial y}{\partial t} = (y_\infty(v_{sh}) - y) / \tau_y(v_{sh}), \quad y = m, n, h.$$

Other parameters appearing in the model are:  $R_{ss}$  is the spine stem resistance,  $R_{sh}$  is the spine head membrane resistance,  $R_i, R_m, a$  are parameters given previously, and  $\bar{N}$  is the spine density (the number of spines per unit physical length of the dendrite). This is the parameter of interest here, and the one we will consider to be spatially dependent. For the current–voltage relation associated with the active spine heads, Baer and Rinzel used

$$I_{ion} = \gamma A_{sh} \{ \bar{g}_{Na} m^3 h (v - v_{Na}) + \bar{g}_K n^4 (v - v_K) + \bar{g}_l (v - v_l) \}.$$

The constant  $A_{sh} = 1 \mu\text{m}^2$  is the spine-head area mentioned above, and  $\gamma$  is a parameter controlling channel density with respect to temperature, but here is fixed at the value 2.5, consistent with the parameter choice in [23]. Some typical values for the parameters are:  $R_{ss} = 2000 \text{M}\Omega$ ,  $R_{sh} = 10^6 \text{M}\Omega$ ,  $C_m = C_{sh} = 1 \mu\text{F}/\text{cm}^2$ ,  $a = 20 \mu\text{m}$ ,  $\bar{g}_{Na} = 120 \text{mS}/\text{cm}^2$ ,  $\bar{g}_K = 36 \text{mS}/\text{cm}^2$ ,  $\bar{g}_l = 0.3 \text{mS}/\text{cm}^2$ ,  $v_{Na} = 115$ ,  $v_K = -12$ ,  $v_l = 10.6$ . For the dynamics of the gating variables, each  $y_\infty = \alpha / (\alpha + \beta)$ , and each  $\tau = 1 / (\alpha + \beta)$ , where

$$\alpha_m = 0.1 \frac{25 - v}{\exp\left[\frac{25-v}{10}\right] - 1}, \quad \beta_m = 4 \exp[-v/18],$$

$$\alpha_h = 0.07 \exp[-v/20], \quad \beta_h = \frac{1}{\exp\left[\frac{30-v}{10}\right] + 1} \tag{35}$$

$$\alpha_n = 0.01 \frac{10 - v}{\exp\left[\frac{10-v}{10}\right] - 1}, \quad \beta_n = 0.125 \exp[-v/80].$$

We scale the two current conservation equations (with the appropriate changes to the coefficients in the  $m, n, h$  equations), by introducing the dimensionless variables  $\tilde{t} = t/\tau_m$ ,  $\tilde{x} = x/\lambda$ ,  $\tilde{v} = v/V_{\text{char}}$  (for both  $v_d$  and  $v_{\text{sh}}$ , with  $V_{\text{char}}$  being some characteristic potential), with  $\tau_m = R_m C_m$ ,  $\lambda = \sqrt{aR_m/2R_i}$ , and the model becomes (after dropping the tilde notation)

$$\frac{\partial v_d}{\partial t} + v_d = \frac{\partial^2 v_d}{\partial x^2} + \rho \bar{n}(x)(v_{\text{sh}} - v_d),$$

$$\delta \frac{\partial v_{\text{sh}}}{\partial t} + i_{\text{ion}}(v_{\text{sh}}, m, n, h) = \rho(v_d - v_{\text{sh}}),$$

where  $\delta$  is the ratio of time scales  $R_{\text{sh}}C_{\text{sh}}/\tau_m$ ,  $i_{\text{ion}} = R_{\text{sh}}I_{\text{ion}}/V_{\text{char}}$ ,  $\rho = R_{\text{sh}}/R_{\text{ss}}$ , and  $\bar{n} = R_m \bar{N}/2\pi a R_{\text{sh}}$ . Then the problem is: given boundary conditions analogous to the Morris–Lecar example, recover the (scaled) spine density function  $\bar{n} = \bar{n}(x)$ . We use the same idea as in the previous sections and, given initial conditions  $v_d(x, 0)$ ,  $v_{\text{sh}}(x, 0)$ ,  $m(x, 0)$ ,  $n(x, 0)$ ,  $h(x, 0)$ , and boundary conditions  $v_d(0, t)$ ,  $\frac{\partial v_d}{\partial x}(0, t)$ , we solve for  $\bar{n}_2, \bar{n}_3, \dots, \bar{n}_{k-1}$ .

To illustrate the use of the algorithm in this case we first have to produce input data for it. As in the the previous examples, we do this by solving the direct problem for some initial and boundary conditions, given  $q(x)$ . In this case we choose:  $v_d(x, 0) = 1$ ,  $v_{\text{sh}}(x, 0) = -10$ ,  $m(x, 0) = 0.01$ ,  $n(x, 0) = 0.01$ ,  $h(x, 0) = 0.01$ ,  $v_d(0, t) = 1$ ,  $v_d(L, t) = 1$ , for  $L = 2$ , and piecewise constant  $q(x)$  as shown in Fig. 8. Then, we input the initial data  $v_{d,i}^1$ ,  $v_{\text{sh},i}^1$ ,  $m_i^1$ ,  $n_i^1$ ,  $h_i^1$ , and boundary data  $v_{d,1}^j$ ,  $v_{d,2}^j$ , into our algorithm, to recover  $\bar{n}_2, \bar{n}_3, \dots, \bar{n}_{k-1}$ . The final results are shown in Fig. 8. Note that we add independent noise terms to all the input data, with the only restriction that we keep the difference  $v_{d,2}^j - v_{d,1}^j$  noise-free (see Example 1 in Section 4 for a more detailed explanation of the noise terms).

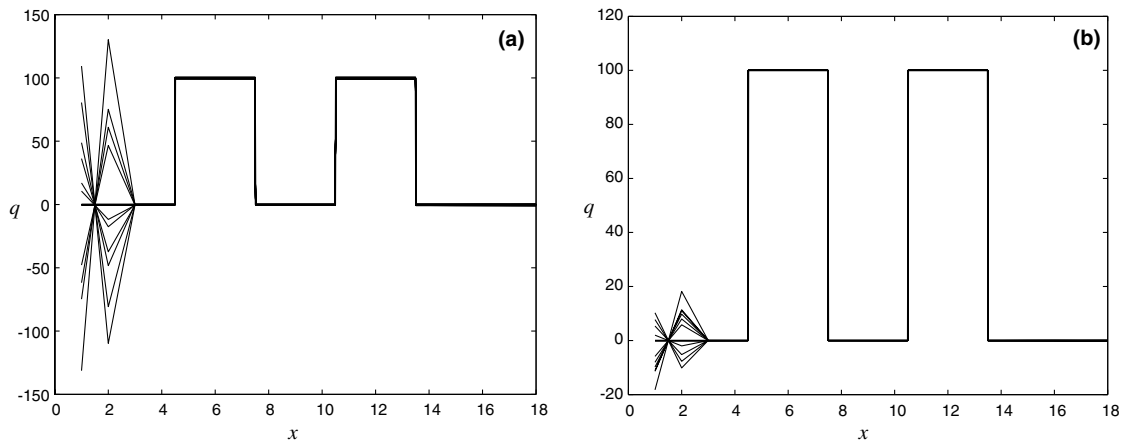


Fig. 8. Results for finding the spine density  $n(x)$  for the Baer–Rinzel model: (a) for 1% noise level and (b) for 0.1% noise level. The thin lines are the computed solutions, the thick line is the exact solution.



## 7. Conclusions

We now know through various microelectrode and ion specific staining experimental techniques that ion channels in excitable cells are not distributed uniformly across the membrane [18,19,21,22]. It is likely that, to fully understand what firing pattern capabilities a given cell type has, we need to understand more about its ion channel distributions. While it is still a major challenge experimentally to obtain even reliable mean estimates for conductance parameters, it is worth exploring ways to implement approaches to recover the main features of these distributions.

In this paper we developed a non-optimization approach for using overdetermined (temporal) boundary data to recover a distributed parameter representing an ion channel density function. In general, the method shows promise in other circumstances, but we have not explored all the possibilities yet. We demonstrated the method's capability to recover the parameter in non-linear systems as well as in linear scalar equations, even if significant noise is present in the input data. In principle we could apply our numerical approach to a fiber with variable, but known, diameter, adjusted to handle the spatially variable diffusion term. This is a reasonable consideration since dendrites have taper and other non-uniformities independent of branching. However, these properties do not fundamentally change the algorithm, so this generalization was not pursued here.

Our method marches from the left (where we impose the overdetermined boundary data) to the right in its construction of scaled density  $q(x)$ . The right-side boundary condition is of little use in our scheme even though the method's accuracy seems to be maintained as we approach the right-side boundary. This needs to be explored further, but the decay of the influence of the right-side boundary conditions into the domain might be a diffusive Saint-Venant type decay. We can show from constructing a total energy function  $E(x, t)$  for Eq. (6), and given  $q$ , that  $E$  drops off exponentially as one moves away from the boundary, i.e., as  $l - x$  increases. This does not, however, provide for a completely satisfactory answer for the inverse problem.

We have not carried out a complete sensitivity analysis, and more analysis of the model is anticipated to make the approach more robust. However, the method is straightforward, uses the type of data that can be measured, and though it only uses the boundary data, it successfully recovers the features of the ionic channel and spine density distributed parameters.

## Acknowledgments

The first author would like to thank Thomas Seidman for suggesting transforming Eq. (6) to Eq. (13). The second author was supported by the National Science Foundation upon Agreement No. 0112050.

## Appendix A

From the theory of parabolic initial boundary value problems it can be shown that there exists a unique solution to (6)–(8) for a given  $q \in L^2[0, 1]$ . What we want for our problem is that we have uniqueness of  $q$ , given  $i(t)$ ,  $v(0, t)$ .

**Lemma 1.** *Let  $v_1, v_2$  be solutions to (6)–(8) corresponding to  $q_1, q_2 \in L^2[0, l]$ , for the same stimulus  $i(t)$ , where  $i \in C^2(0, T)$  with  $i(0) = 0$ ,  $i'(0) \neq 0$ . If  $v_1(0, t) = v_2(0, t)$  on  $(0, T)$ , then  $q_1(x) = q_2(x)$  on  $[0, l]$ .*

This lemma is an application of inverse Sturm-Liouville eigenvalue problem theory. The proof can be adapted from results in Chapter 4 of Kirsch's book [34], so we omit writing out the details here.

## References

- [1] A. Hodgkin, A. Huxley, A quantitative description of membrane current and its applications to conduction and excitation in nerve, *J. Physiol.* 117 (1952) 500.
- [2] W. Rall, Membrane potential transients and membrane time constants of motoneurons, *Exp. Neurol.* 2 (1960) 503.
- [3] W. Rall, Theory of physiological properties of dendrites, *Ann. NY Acad. Sci.* 96 (1962) 1071.
- [4] W. Rall, Core conductor theory and cable properties of neurons, in: *Handbook of Physiology. The Nervous System*, vol. 1, 1977 p. 39.
- [5] W. Rall, R.E. Burke, W.R. Holmes, J.J.B. Jack, S.J. Redman, I. Segev, Matching dendritic neuron models to experimental data, *Physiol. Rev.* 172 (1992) S159.
- [6] W.R. Holmes, W. Rall, Estimating the electrotonic structure of neurons with compartmental models, *J. Neurophysiol.* 68 (4) (1992) 1438.
- [7] J.J.B. Jack, S.J. Redman, An electrical description of a motoneurone, and its application to the analysis of synaptic potentials, *J. Physiol.* 215 (1971) 321.
- [8] T.H. Brown, R.A. Friske, D.H. Perkel, Passive electrical constants in three classes of hippocampal neurons, *J. Neurophysiol.* 46 (1981) 812.
- [9] D.M. Durand, P.L. Carlen, N. Gurevich, A. Ho, H. Kunov, Electrotonic parameters of rat dentate granule cells measured using short current pulses and HRP staining, *J. Neurophysiol.* 50 (1983) 1080.
- [10] M. Kawato, Cable properties of a neuron model with non-uniform membrane resistivity, *J. Theor. Biol.* 111 (1984) 149.
- [11] A. D'Aguzzo, B.J. Bardakjian, P.L. Carlen, Passive neuronal membrane parameters: comparison of optimization and peeling methods, *IEEE Trans. Biomed. Eng.* 33 (1986) 1188.
- [12] A.K. Schierwagen, Identification problems in distributed parameter neuron models, *Automatica* 26 (1990) 739.
- [13] S.J. Cox, A new method for extracting cable parameters from input impedance data, *Math. Biosci.* 153 (1998) 1.
- [14] J.A. White, P.B. Manis, E.D. Young, The parameter identification problem for the somatic shunt model, *Biol. Cybern.* 66 (1992) 307.
- [15] S.J. Cox, L. Ji, Identification of the cable parameters in the somatic shunt model, *Biol. Cybern.* 83 (2000) 151.
- [16] S.J. Cox, B. Griffith, Recovering quasi-active properties of dendrites from dual potential recordings, *J. Comput. Neurosci.* 11 (2) (2001) 95.
- [17] S.J. Cox, L. Ji, Discerning ionic currents and their kinetics from input impedance data, *Bull. Math. Biol.* 63 (2001) 909.
- [18] D. Johnston, J.C. Magee, Characterization of single voltage-gated  $\text{Na}^+$  and  $\text{Ca}^{2+}$  channels in apical dendrites of rat CA1 pyramidal neurons, *J. Physiol.* 487 (1995) 67.
- [19] J.C. Magee, Dendritic hyperpolarization-activated currents modify the integrative properties of hippocampal CA1 pyramidal neurons, *J. Neurosci.* 18 (19) (1998) 7613.
- [20] R.D. Traub, R.K.S. Wong, R. Miles, H. Michelson, A Model of CA3 Hippocampal pyramidal neuron incorporating voltage-clamp data on intrinsic conductances, *J. Neurophysiol.* 66 (1991) 635.
- [21] D. Johnston, J.C. Magee, C.M. Colbert, B.R. Christie, Active properties of neuronal dendrites, *Annu. Rev. Neurosci.* 19 (1996) 165.
- [22] B.V. Safronov, Spatial distribution of Na and K channels in spinal dorsal horn neurones: role of the soma, axon and dendrites in spike generation, *Prog. Neurobiol.* 59 (1999) 217.
- [23] S.M. Baer, J. Rinzel, Propagation of dendritic spikes mediated by excitable spines: a continuum theory, *J. Neurophysiol.* 65 (4) (1991) 874.
- [24] C.P. Fall, E.S. Marland, J.M. Wagner, J.J. Tyson, *Computational Cell Biology*, Springer, New York, 2002.
- [25] H.C. Tuckwell, *Introduction to Theoretical Neurobiology*, vol. 1, Cambridge University, Cambridge, 1988.

- [26] J.J.B. Jack, D. Noble, R.W. Tsien, *Electric Current Flow in Excitable Cells*, Clarendon, Oxford, 1975.
- [27] A.L. Hodgkin, The optimal density of sodium channels in an unmyelinated nerve, *Philos. Trans. R. Soc. Lond. B* 270 (1975) 297.
- [28] H.W. Engl, W. Rundell (Eds.), *Inverse Problems in Diffusion Processes*, SIAM, Philadelphia, PA, 1995.
- [29] G. Anger, *Inverse Problems in Differential Equations*, Plenum, New York, 1990.
- [30] V. Isakov, *Inverse Problems for Partial Differential Equations*, Springer, New York, 1998.
- [31] P.G. Danilaev, *Coefficient Inverse Problems for Parabolic Type Equations and Their ApVSP, VSP*, Utrecht, 2001.
- [32] M. Tadi, M.V. Klibanov, W. Cai, An inversion method for parabolic equations based on quasi-reversibility, *Computers and Mathematics with Applications* 43 (2002) 927.
- [33] C. Morris, H. Lecar, Voltage oscillations in the barnacle giant muscle fiber, *Biophys. J.* 35 (1981) 193.
- [34] A. Kirsch, *An Introduction to the Mathematical Theory of Inverse Problems*, Springer, New York, 1995.