A Distributed Parameter Identification Problem
in Neuronal Cable Theory Models

Jonathan Bell* and Gheorghe Craciun†

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 nonlinear model that has closed formed solutions. Then the numerical method is shown to be applicable to nonlinear 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.

Keywords: cable theory, inverse problems, ion channel density, distributed parameters

Classification: 35, 92

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

\[ C \frac{\partial v}{\partial t} + I_{\text{ion}} = \frac{\partial^2 v}{\partial x^2}. \]

*Department of Mathematics and Statistics, University of Maryland, Baltimore County.
†Mathematical Biosciences Institute, Ohio State University. This author was supported by the National Science Foundation under Agreement No. 0112050.
The current-voltage relation $I_{\text{ion}}(v, \ldots)$, which is a sum of current densities (sodium, calcium, potassium, chloride, etc.), usually depends not only on the transmembrane potential $v$, but also on other "gating" variables. Through pharmacological, environmental, and electrical means individual ionic currents can be isolated and measured. If a particular ionic current is written as $I_A = g_A(v - v_A)$, $g_A$ is the conductance, and $v_A$ is the reversal potential. From the Nernst equation [1], $v_A$ is proportional to the logarithm of the ratio of steady external to internal concentrations of the specific ion. In this paper we assume these reversal potentials are known quantities. The conductance has the form $g = \overline{g} a^p q^q$, where $\overline{g}$ is the maximum conductance (conductance per channel times the channel density), $p$ and $q$ are nonnegative numbers, and $a$ and $b$ are activation and inactivation variables, respectively.

Since the time Hodgkin and Huxley developed their famous model [2], 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. Some work on measuring spatial variations in channel densities, hence nonconstant maximal conductances, has been done; see, for example, [3], [4].

In this paper we assume an individual ionic current can be isolated in which there is a nonuniform 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 as our first example a linear neural model with boundary conditions, then nondimensionalize the model and discuss relevant previous work for estimating parameters in such models. In the third section we give a uniqueness result and a transform of the model we found useful in generating nontrivial 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 nonlinear 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 switch gears a little and consider a model of a dendrite with active spines. This model was originally formulated by Baer and Rinzel [5]. This is a higher dimensional nonlinear 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 model situations, and the need for more theoretical foundations. The point we want to emphasize is that we have an effective method that avoids an expensive optimization procedure.
2 Formulation of the Inverse Problem for a Linear Nerve Cable Model

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

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

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 \( u \) satisfies a first order ordinary differential equation in \( t \). In the simplest cases 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 fact, through pharmacological use of channel blockers as mentioned earlier, the dynamics of a single channel can be studied. These assumptions lead to a significant reduction of the above model system to the scalar equation

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

Considering the domain to be \( 0 < x < L, \ 0 < t < T \), the appropriate boundary conditions of interest to us are

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

The first condition expresses our injecting 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 can assume that the fiber terminates at \( x = L \), or experimentally we set up experimental conditions to block current flow at that end. For a discussion of cable modeling see, e.g., Tuckwell [6] or Jack et al [7]. 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.

There has been a fair amount of work in this setting in estimating model parameters. Rall and colleagues developed "peeling" methods for single fibers and "constrained optimization" methods for coupled compartmental models for multiple neuron cases (see Rall et al [8]; Vanier and Bower [9]; see also Jack et al [7], D’Aguanno et al [10], White et al [11], Holmes and Rall [12], Tawfik and Durand [13]). These papers deal only with estimating constant parameters. Cox ([14], [15],[16],[17]) developed a moment method he calls an "input impedance" method to recover (uniquely) constant parameters \( R_i, C_m, g, L \), and hence the cell’s membrane time constant and electrotonic length (see [17]). That is, along with the boundary conditions above, and the initial condition, it is sufficient
to have $v(0, t)$ as the overspecified data for the inverse problem (constant-
parameter case), and to compute a certain number of integrals of $v(0, t)$ and $i_o(t)$. With little extra work the approach can be used on a somatic shunt model
where the current injection is performed at the cell soma, which is represented
by a couple extra parameters. Hence, in this case the boundary condition at
$x = 0$ above is replaced by a boundary condition of the form

$$
\frac{1}{R_i} \frac{\partial v}{\partial x}(0, t) - g_s v(0, t) - C_s \frac{\partial v}{\partial t}(0, t) = -i_o(t).
$$

Cox applied his input impedance method to the soma case (Cox and Ji [14]),
tapered and branched cells (Cox [17]), as well as considering other inverse
problems such as estimating current-voltage relations (Cox and Ji [16], Cox and
Griffith [15]). The same boundary data plays a crucial role for our methodology
below.

In this and the next section we consider the linear model above with known
parameters except for the single distributed channel density parameter $N = N(x)$. Then we can represent $g = g_o N, C_m = C_o N + C_1,$ where $g_o, C_o, C_1$
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 can expect to measure the voltage at $x = 0$ though the
electrode is used to input the current stimulus. Is that enough to estimate $N$?
How good of a job can be done with such boundary data? We are interested in
answering these questions below.

We substitute the above expressions for $g$ and $C_m$ into the equation, and
let $\tilde{v}(\tilde{x}, \tilde{t}) = v(x, t)/v_n - 1, \tilde{x} = x \sqrt{\frac{2 R_i C_1}{g_o}}, \tilde{t} = t g_o / C_o, q(\tilde{x}) = C_o N(x) / C_1.$
Then (1), (2) become

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

$$v(0, \tilde{t}) = 0, \quad \frac{\partial \tilde{v}}{\partial \tilde{x}}(0, \tilde{t}) = -i(\tilde{t}), \quad \frac{\partial \tilde{v}}{\partial \tilde{x}}(l, \tilde{t}) = 0,$$

where now $l = L \sqrt{\frac{2 R_i C_1}{g_o}}, \tilde{T} = T g_o / C_o, i(\tilde{t}) = \frac{1}{\pi a v_n} \sqrt{\frac{2 R_i C_1}{g_o}} i_o(t C_o / g_o)$.

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

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

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

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

(3)

\[ v(x, 0) = 0, \quad 0 < x < l, \]  

(4)

\[ \frac{\partial v}{\partial x}(0,t) = -i(t), \quad \frac{\partial v}{\partial x}(l,t) = 0, \quad 0 < t < T, \]  

(5)

with the extra measurements given by

\[ v(0,t) = f(t). \]  

(6)

Now the mathematical problem is to recover \( q(x) \) on \([0,l]\), given \( i(t) \) and \( f(t) \) on \((0,T)\). Further motivation for considering such a problem in this context comes, for example, from Johnston et al [3], [4], Magee [18], and Traub et al [19]. Johnston and Magee [3] found low-voltage activated and high-voltage, moderate conductance \( Ca^{2+} \) channel types along the pyramidal dendrites. In Traub et al [19], based on experimental work from several labs, they model the CA3 pyramidal cell dendrites by 19 compartments with 6 different active ionic channels having varying spatial distributions.

While there is a considerable literature now on inverse problems for parabolic equations (see, e.g., England and Rundell [20]), 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 above problem (3)-(5) is to recover a spatially distributed coefficient from overspecified temporal data.

### 3 Uniqueness and a Transformation of the Problem

From the theory of parabolic initial boundary value problems it can be shown that there exists a unique solution to (3)-(5) for a given \( q \in L^2[0,1] \). What we want for our problem is that, for a given \( i(t) \), we have uniqueness of \( q \) for given \( v(0,t) \).

**Lemma 1.** Let \( v_1, v_2 \) be solutions to (3)-(5) corresponding to \( q_1, q_2 \in L^2[0,1] \), 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 \( q_1(x) = q_2(x) \) a.e. 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 [21], so we omit writing out the details here.

For the time being let us consider the notationally more general case, namely

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

(3)

\[ v(x, 0) = v_0(x), \quad 0 < x < l, \]  

(7)

\[ v(0,t) = f_0(t), \quad \frac{\partial v}{\partial x}(0,t) = f_1(t), \quad 0 < t < T, \]  

(8)

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

(9)
If we let $p(x) = 1 + q(x)$ and $v(x, t) = p^{-1/4} u(s, t)$, where \( \frac{du}{dx} = p^{1/2}, s(0) = 0 \), and substitute these into (3), (7), (8), we obtain

\[
\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial s^2} - Q(s)u, \quad 0 < s < s_r, 0 < t < T, \quad (10)
\]

\[
u(s, 0) = p^{1/4}(x(s))v_0(x(s)), \quad 0 < s < s_r, \quad (11)
\]

\[
u(0, t) = \eta_0(t) = f_0(t)p_0^{1/4}, \quad \frac{\partial \nu}{\partial s}(0, t) = \eta_1(t) = f_1(t)p_0^{-1/4} - \gamma_0 f_0(t), \quad (12)
\]

\[
u(s_r, t) = \mu_0(t) = g_0(t)p_1^{1/4}, \quad \frac{\partial \nu}{\partial s}(s_r, t) = \mu_1(t) = g_1(t)p_1^{-1/4} - \gamma_1 g_0(t), \quad (13)
\]

where \( p_0 = p(0) \) and \( \gamma_0 = -\beta / 4p_0^{3/2} \), with analogous expressions for \( p_1, \gamma_1 \).

Here \( s_r = \int_0^l \sqrt{p(x)} \, dx \), which can only be determined by obtaining \( p \). But in (10) \( Q \) needs to satisfy \( Q = (\alpha q - \alpha^2 \beta^2) / \alpha \beta^2 \) where \( \alpha = p^{-1/4}, \beta = p^{1/2} \). An equivalent way to write this expression in terms of \( p \) is

\[
\frac{d^2 p}{ds^2} - \frac{3}{4} \left( \frac{dp}{ds} \right)^2 + 4p(p - 1) - 4p^2 Q(s) = 0 \quad (14)
\]

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.

For the problem of finding \( Q(s) \) in (10), i.e., depending only on the space variable, there is literature on this more standard form. In this form the scalar variable \( u \) can be interpreted in terms of the local temperature in heat conduction problems, or light intensity in optical tomography. A couple of approaches to mention about the spatially distributed parameter problem include optimization methods (e.g., Tadi [22]), and methods based on quasireversibility (Lattes and Lions [23], Tadi, Klibanov, and Cai [24], Klibanov and Lucas [25]; Daniiliev [26]). These methods leave a number of theoretical questions unresolved, as does our method below.

Isakov [27] and Klibanov [28] have some detailed uniqueness results. These results require the left and right Dirichlet data to belong to the range of a Laplace-like transform. That is, there should be functions \( b_1 \) and \( b_2 \) such that \( \eta_0 = H[b_1] \) and \( \mu_0 = H[b_2] \), where

\[
H[b](t) = \frac{2}{\sqrt{2\pi t}} \int_0^\infty e^{-\tau^2 / 4t} b(\tau) d\tau.
\]

The above inverse problem can be transformed to a similar problem with hyperbolic equation \( \frac{\partial^2 u}{\partial s^2} = \frac{\partial^2 u}{\partial x^2} - Q(s)u \). Although the inversion process is unstable, there are stronger methods for uniqueness for hyperbolic problems than for parabolic problems so the approach can be useful in inverse problems like the linear problems in this paper. Tadi, Klibanov and Cai [24] also discuss this transformation and mention two cases when the uniqueness result for the hyperbolic case can be used to the case of (10)-(13). Also, Klibanov [28] approaches uniqueness through the use of Carleman estimates. A related approach
is to study the Neumann-Dirichlet map, relying on corresponding results for the above hyperbolic equation (see, e.g., [29]). There is also a literature on overdetermined final-time data. While interesting mathematically, the results are not useful for the biological application discussed here.

We are actually interested in a method that would be also applicable to more experimentally relevant, nonlinear versions of these equations, as we describe in Sections 5 and 6. In general, previous numerical methods of solving (10)-(13) are tailored specifically for these equations, and do not seem to have any immediate applications for more general, nonlinear versions of these equations.

Employing a trick used in Danilov [26], since \( Q(s) = (\frac{\partial^2 u}{\partial x^2} - \frac{\partial u}{\partial t}) / u \) is independent of \( t \), then \( \frac{\partial Q}{\partial t} = 0 \), which implies \( \frac{\partial u}{\partial t} \left( \frac{\partial^2 u}{\partial x^2} - \frac{\partial u}{\partial t} \right) / u = 0 \), or

\[
u \frac{\partial^2 u}{\partial x^2 \partial t} - u \frac{\partial u}{\partial t} \frac{\partial^2 u}{\partial t^2} - \frac{\partial u}{\partial t} \frac{\partial^2 u}{\partial x^2} + \left( \frac{\partial u}{\partial t} \right)^2 = 0. \tag{15}
\]

This calls for a number of observations. Equation (14) has at least one two-parameter family of closed-form solutions. This can be seen by writing (14) as

\[
p \frac{d^2 p}{ds^2} - \frac{3}{4} \left( \frac{dp}{ds} \right)^2 + kp^2 = 4p^2 Q - 4p(p - 1) + kp^2,
\]

for any constant \( k > 0 \). If the right-hand side is zero, then the homogeneous equation is solvable, and this gives \( p(s) = p_0 \cos^4 \left( -\frac{\sqrt{k}}{2} s + C_1 \right) \) 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 \( \frac{dp}{ds} = 2 \cos^2 (s) \) 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 (10) becomes

\[
\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{1}{4} \sec^4 (s) u. \tag{16}
\]

Note that we have two particular classes of solutions to (16) worth mentioning. If one separates variables, e.g., \( u(s, t) = e^{-t} y(s) \), then \( y \) must satisfy

\[
p \frac{d^2 y}{ds^2} + \left( \frac{1}{4} \sec^4 (s) + 1 \right) y = 0,
\]

which has solutions \( y(s) = y_1 (s) = \cos (s) \sin \left( \frac{1}{2} \tan (s) \right) \) and \( y(s) = y_2 (s) = \cos (s) \cos \left( \frac{1}{2} \tan (s) \right) \). In fact, for \( p(s) = p_0 \cos^4 \left( \frac{\sqrt{k}}{2} s - C_1 \right) \), 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 equation (10), along with its companion expression with sin replaced by cos.

Notice that $u_1 = e^{-t} y_1$ satisfies $u_1(0, t) = 0$, $\frac{\partial u_1}{\partial x}(0, t) = \frac{1}{2} e^{-t}$, while $u_2 = e^{-t} y_2$ satisfies $u_2(0, t) = e^{-t}$, $\frac{\partial u_2}{\partial x}(0, t) = 0$. In fact, for any $C^2$ function $\varphi(s)$, and constant $K$, $u(s, t) = e^{Kt} \varphi(s)$ is a classical solution to (15).

Another observation is to make the reduction-of-order type change of variable $w = \frac{\partial \varphi}{\partial s} / u$ to equation (15) to obtain the (non-local) convection-diffusion equation

$$\frac{\partial w}{\partial t} = \frac{\partial^2 w}{\partial s^2} + 2 \left( \int_0^t \frac{\partial w}{\partial s} dt \right) w_s. \quad (17)$$

This also gives

$$Q(s) = \frac{\partial^2 u}{\partial s^2} - \frac{\partial u}{\partial s} = \left( \int_0^t \frac{\partial w}{\partial s} dt \right)^2 + \int_0^t \frac{\partial^2 w}{\partial s^2} dt - w(s, t). \quad (18)$$

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 workable on the original problem (3)-(6), and can be extended to nonlinear problems.

## 4 The Algorithm for $q(x)$ and Numerical Results

We propose a new numerical procedure to solve the type of inverse problems we described in the previous section. We will first describe our numerical procedure for the simplest type of inverse problem we are interested in:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} - q u, \quad (19)$$

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 $\frac{\partial u}{\partial x}(0, t)$ for $0 < t < t_{max}$. We want to compute $q(x)$ for $0 < x < L$.

Consider some positive integer $n$, and consider a $n \times n$ uniform grid for the domain $[0, L] \times [0, t_{max}]$.

Let us denote by $u_i^m$ and $q_i$ the values of $u(0, t)$ and $q(x)$ on the discrete grid. 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 the equation (19):

$$\frac{u_i^{m+1} - u_i^m}{\Delta t} = \frac{u_{i+1}^m + u_{i-1}^m - 2 u_i^m}{(\Delta x)^2} - q_i u_i^m. \quad (20)$$

According to (20) we have

$$q_i = \frac{1}{u_i^n} \left( -\frac{u_i^{m+1} - u_i^m}{\Delta t} + \frac{u_{i+1}^m + u_{i-1}^m - 2 u_i^m}{(\Delta x)^2} \right), \quad (21)$$

8
assuming \( u_i^m \neq 0 \). Also according to (20) we have

\[
    u_i^{m+1} = -u_i^m + 2u_i^m + (\Delta x)^2 \left( \frac{u_i^{m+1} - u_i^m}{\Delta t} + qu_i^m \right).
\]  \hspace{1cm} (22)

Since we know \( u(x,0), u(0,t) \) and \( u_x(0,t) \), we shall suppose we know \( u_1^1, u_1^m \) and \( u_2^m \), for all \( 1 \leq m,i \leq n \).

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

**Step 1**: first, compute \( q_2 \) from (21), since we know \( u_1^1, u_1^2, u_2^2 \); then, use (22) 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 (21), since we know \( u_2^1, u_2^3, u_3^3 \); then, use (22) 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 (21), since we know \( u_i^1, u_i^1, u_i^1, u_i^1 \), and \( u_{i+1}^2 \); then, use (22) 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 (21), since we know \( u_{n-2}^1, u_{n-1}^m, u_n^1, \) and \( u_{n-1}^n \); then, use (22) 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 \( q_2, q_3, ..., q_{n-1} \) 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

\[
    u(x,t) = e^{-t} \cos(x) \sin(\frac{1}{2} \tan(x))
\]

is a solution of (19) for

\[
    q(x) = -\frac{1}{4} \sec^2(x).
\]

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 \( q_i = -\frac{1}{4} \sec^2(\frac{i-1}{n}) \), 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 \( u_1^m = e^{-\frac{m-1}{n}}, u_n^m = u_1^m + \frac{e^{-\frac{m-1}{n}}}{n} \). The expressions for \( u_1^m \) and \( u_n^m \) are obtained by plugging into the formula for \( u(x,t) \) above, and the expression for \( u_2^m \) is just \( u_2^m + \frac{w_2}{n}(0, \frac{m-1}{n}) \Delta t \). We would like to see how robust is our algorithm if we add random noise terms to the input data. In Figure 1 we show the logarithmic scale error obtained using our algorithm on a \( 100 \times 100 \) grid (i.e., \( n = 100 \)). We initialized \( u_1^1, u_1^m \) 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. More exactly, we replaced each \( u^i_t \) by \((1 + \epsilon_1)u^i_t \), each \( u^m_t \) by \((1 + \epsilon_2)u^m_t \). Then we updated each \( u^m_t \) to keep the difference \( u^m_t - u^1_t \) noise-free, since their difference corresponds to the stimulus current \( i(t) \), and we have good control on \( i(t) \) in an experimental setting. The noise terms \( \epsilon_1, \epsilon_2 \) are independent and uniformly distributed on some interval \([-a, a]\). We refer to \( a \) as the amplitude of the noise.

On the other hand, we did experiment with more general types of noise (e.g., independent noise terms \( \epsilon_1^1, \epsilon_1^m, \epsilon_2^m \) for each \( u^1_t, u^1_m, u^2_m \)), and the overall results were similar to the ones in Figure 1.

![Figure 1: Logarithmic scale error for solutions of the inverse problem (19): (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.](image-url)

Note that the approximation results for \( u(x, t) \) scale very well with the error rate (see Figure 1(b)). The approximation results for \( q(x) \) in Figure 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). This is simply due to the fact that we reached the level of the error of approximating \( q(x) \) by using the exact input values. The error of approximating \( q(x) \) by the \( q_i \) obtained from formula (21), 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 Figure 2 we show results for 1% and 0.1% input noise. In Figure 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 \( \epsilon_1, \epsilon_2 \) 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 Figure 2(b)).

Figure 2: The dotted curves are 20 computed versions of $q$ for the inverse problem (19), 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 $q$.

**Example 2.** We compare results of our method with results of Tadi, Klibanov, and Cai [24] for discontinuous $q(x)$. Piecewise constant $q$ is reasonable to consider in the nerve fiber problem because, given sparse measurements, that is the type of representation we are faced with.

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 $u^m_i$ given $q_i$, and 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 intervals, 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 Figure 3(a)).

Given this $q$ and these initial and boundary conditions for $u$ we can solve the equation (19) on an 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 $u^m_i$ for all $1 \leq i, m \leq n$.

Then, we can input only $u^1_i$, $u^m_i$, 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 end up with $k$ times as many time than space intervals in $u^m$. 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 $u^m$. This is not an issue if we use unconditionally convergent schemes, e.g., the Crank-Nicholson scheme.

Before we go on and use our algorithm to compute an approximation of the piecewise constant function $q$, we should ask ourselves what is the best possible result we can hope for, based on input data $u^1_i$, $u^m_i$, $u^2_m$ 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 Figure 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 Figure 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 (21) to these values $u_i^m$ for each fixed $m = 1, 2, \ldots, n$. 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 Figure 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 $q_i$.

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

![Figure 3](image-url)

**Figure 3**: *Error introduced by classical numerical procedures for the direct problem (19), in the region of the domain near $t = 0$: (a) by the explicit scheme (b) by the Crank-Nicholson scheme. The function $q$ is piecewise constant.*

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 $t = t_{\text{max}}$, and use the explicit scheme output as input into our algorithm, we get the results in Figure 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 [24] also can localize the “bumps” of $q$, the approximations in [24] are more diffusive, and the height of the approximation “bumps” is often only about 75% of the correct value (see Figures 4, 5, 8 and 9 in [24]). Also, the algorithm described in [24] is 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 $u_i^1, u_{11}^m, u_{22}^m$, we get very good approximation for $q(x)$, even for multiple
Figure 4: Results for variable $\Delta t$ for the inverse problem (19): (a) $q(x)$ has one “bump” (b) $q(x)$ has two “bumps” at some distance from each other (c) $q(x)$ has three “bumps” of different sizes.

“bumps”, as we can see in Figure 5.

Figure 5: Our numerical procedure gives accurate results for input obtained with classical numerical procedures for equation (19), 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 (b) the Crank-Nicholson scheme.

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

**Example 3.** Recall that the equation (19) is just a simplification of the type of equations that we derived in Section 2. From the point of view of the applications we are interested in, we would actually like to be able to solve the inverse problem for equation (3).

On the other hand, it is easy to see that the same explicit finite difference scheme that we described for equation (19) at the beginning of this section can be used for equation (3), and exactly the same algorithm can be applied to compute $q_2, q_3, ..., q_{n-1}$ for equation (3).

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 the equation (3) for the same initial and boundary conditions as in Example 2. Then, we input the
computed $u^1$, $u^m$, and $u^q$ into our algorithm to recover $q$, from the explicit finite difference scheme for equation (3). The results are shown in Figure 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.

![Image](image_url)

Figure 6: Results for input obtained with classical numerical procedures, if we avoid the region of the domain near $t = 0$, for equation (3). We show 20 computed versions of $q$, each with 0.01% random noise, for input data obtained using: (a) the explicit scheme (b) the Crank-Nicholson scheme.

5 Nonlinear Example: the Morris-Lecar Model

Morris and Lecar [30] developed a model for barnacle giant muscle which has become a popular two-variable nonlinear model to compute certain prototypical behavior, like regular trains of impulses. Their model incorporates a nonactivating fast calcium current, nonactivating slower potassium current, and a small leak current. The simulations of their model provided reasonable agreement with their experimental measurements. The form of their model is

$$C_m \frac{\partial v}{\partial t} + g_{Ca} m_\infty (v - v_{Ca}) + g_K w (v - v_K) + g_L (v - v_L) = \frac{a}{3 R} \frac{\partial^2 v}{\partial x^2}$$

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

where the calcium activation variable, $m$, is fast enough to set it to the quasisteady 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 [1].

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

$$(1 + q(x))\frac{\partial v}{\partial t} + g_{Ca}m_\infty(v - v_{Ca}) + q(x)w(v - v_K) + g_L(v - v_L) = \frac{\partial^2 v}{\partial x^2}$$

$$\frac{\partial w}{\partial t} = \varphi_o \{(w_\infty(v) - w)/\tau_w(v)\}$$

where $\varphi_o = \varphi_T$. Finally, the boundary conditions take the form

$$v(x, 0) = v_o(x), \quad \frac{\partial v}{\partial x}(0, t) = -\alpha(t), \quad \frac{\partial v}{\partial x}(L, t) = 0,$$  \quad \left(\alpha = \frac{\lambda_B}{\pi \sigma^2 V_{char}}\right)$$

$$v(0, t) = f(t).$$

So now the problem is to recover a reasonable estimate of $q$ in (23)-(24) by applying (23) and the measured data (26), and using the methodology developed in Section 4. Again we assume all parameters are known except $q(x)$.

![Graphs showing Morris-Lecar model results](image)

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

Note that one can easily express $q$ in terms of $v, \frac{\partial v}{\partial x}, \frac{\partial^2 v}{\partial x^2}$, and $w$, using equation (23). Let us denote by $q_i, v_i^n, w_i^n$ 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 equation (23) to compute $q_2$, given the initial conditions for $v$ and $w$, and given boundary conditions (25) for $v$ at $x = 0$. Then, we can use equation (24) to compute $w_i^n$, and equation (23) to compute $v_i^n$. After that we can use equation (23) to compute $q_3$, and so on. In other words, we can use exactly the same idea as in Section 4 to solve for $q_4, v_4^n, w_4^n$. Some results of applying this algorithm appear in Figure 7. As in a previous example, we used simple initial and boundary conditions $u(x, t) = 1$ if $t = 0, x = 0$, 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 $v_1^i$, $w_1^i$, $v_1^m$, $v_1^n$, except that we keep the difference $v_1^m - v_1^n$ 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 we describe.

6 Recovery of Dendritic Spine Distribution: the Baer-Rinzel Model

Baer and Rinzel [5] formulated a (continuous dendritic spine) cable theory to investigate electrical interactions between many excitatory 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 basically Hodgkin-Huxley kinetics. Given that the area of the spine head is estimated in [5] to be $1 \mu m^2$, and conductance is assumed to be $12 pS$ for a single sodium channel, this would imply $250 Na^+$ channels per spine head. A similar calculation could be done for the potassium channel. Rather than using this model to recover some specific spatially distributed ionic channel density, below we demonstrate our algorithm on the model to recover the 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-ladened dendrite. Alternative possibilities can be explored in a future study. The Baer-Rinzel model is then

$$C_m \frac{\partial v_d}{\partial t} = \frac{a}{2R_i} \frac{\partial^2 v_d}{\partial x^2} - \frac{v_d}{R_m} + \frac{N}{2 \pi a} \frac{v_{sh} - v_d}{R_{sa}}$$

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

Again these are current conservation equations for the dendritic potential, $v_d$, and the spine head potential, $v_{sh}$. Here $m, n, h$ 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: $R_{sa}$ is the spine stem resistance, $R_{sh}$ is the spine head membrane resistance, $R_i, R_m, a$ are parameters given previously, and $N$ is the spine density (the number of spines per unit physical length of the dendrite). This is the parameter of interest to use here, and the one we will consider can be spatially dependent. For the current-voltage relation associated with the active spine heads, Baer and Rinzel used

$$I_{ion} = \gamma A_{sh} \left( \mathcal{I}_{Na} m^3 h(v - v_{Na}) + \mathcal{I}_{K} n^4 (v - v_{K}) + \mathcal{I}_{L} (v - v_l) \right).$$
The constant $A_{sh} = 1 \mu m^2$ is the spine-head area mentioned above, and $\gamma$ is taken to be a speed-up parameter which can be taken to be 2.5. Some typical values for the parameters are: $R_{sg} = 2000M\Omega$, $R_{sh} = 10^6 M\Omega$, $C_m = C_{sh} = 1 \mu F/cm^2$, $a = 20 \mu m$, $g_{Na} = 120 mS/cm^2$, $g_K = 36 mS/cm^2$, $g_l = 0.3 mS/cm^2$, $v_{Na} = 115$, $v_K = -12$, $v_f = 10.6$. For the dynamics of the gating variables, each $y_\infty = \alpha/(\alpha + \beta)$, and each $\tau = 1/(\alpha + \beta)$, where

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

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

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

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

$$\frac{\partial v_d}{\partial \tilde{t}} + v_d = \frac{\partial^2 v_d}{\partial \tilde{x}^2} + \rho \mathcal{P}(x)(v_{sh} - v_d)$$

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

where $\delta$ is the ratio of time scales $R_{sh} C_{sh}/\tau_m$, $i_{ion} = R_{sh} I_{ion}/V_{char}$, $\rho = R_{sh}/R_{sg}$, and $\mathcal{P} = R_m N/2 \pi a R_{sh}$.

So the problem here is, given the analogous boundary conditions as we did in the Morris-Lecar example, recover the (scaled) spine density function $\mathcal{P} = \mathcal{P}(x)$.

We can take what we learned in the previous sections and apply the same algorithm to solve for $\pi_2, \pi_3, ..., \pi_{k-1}$, given initial conditions $v_d(x, 0), v_{sh}(x, 0), m(x, 0), n(x, 0), h(x, 0)$, and boundary conditions $v_d(0, t), v_{sh}(0, t)$.

To illustrate the use of our 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: $v_d(x, 0) = 1$, $v_{sh}(x, 0) = -10$, $m(x, 0) = 0.01$, $n(x, 0) = 0.01$, $h(x, 0) = 0.01$, $v_d(0, t) = 1$, $v_{sh}(L, t) = 1$, for $L = 2$, and piecewise constant $q(x)$ as shown in Figure 8. Then, we input the initial data $v_{d, 0}^{1, i}, v_{sh, 0}^{1, i}, m_i^1, n_i^1, h_i^1$, and boundary data $v_{d, 2}^{i}, v_{d, 2}^{j}$ into our algorithm, to recover $\pi_2, \pi_3, ..., \pi_{k-1}$. The final results are shown in Figure 8. Note that we add independent noise terms to all the input data, with the only restriction that we keep the difference $\nu_{d, 2}^{j} - \nu_{d, 1}^{j}$ noise-free (see Example 1 in Section 4 for a more detailed explanation of the noise terms).
Figure 8: Results for finding the spine density \( n(x) \) for the Baer-Rinzel model: (a) for 1\% noise level, (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. 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 to obtain even reliable mean estimates for conductance parameters, it is worth exploring ways to implement approaches to recovering 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 many other circumstances, but we have not explored all the possibilities yet. We did demonstrate 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 unchanged to a fiber with variable, but known, diameter, adjusted to handle the spatially variable diffusion term. This would be a reasonable consideration since dendrites have taper and other non-uniformities independent of branching. However, this does not fundamentally change the algorithmic ideas, and so this generalization was not pursued in this paper.

Our method marches from the left (where we impose overdetermined boundary data) to the right in its construction of scaled density \( q(x) \). The right-hand boundary condition is of little use to us in our scheme even though the method’s accuracy seems to be maintained as we close in on the right-hand boundary. This needs to be explored further, but perhaps there is at work a diffusive Saint-Venant type decay of the influence of the right-hand boundary conditions into the domain. We can show from constructing a total energy function \( E(x, t) \) for equation (3), and given \( q \), that \( E \) drops off exponentially as one moves away from the boundary, i.e. as \( l - x \) increases. This does not provide for a completely
satisfactory answer for our inverse problem, however.

We have not carried out a complete sensitivity analysis yet, but we know the results are somewhat sensitive to the choice of initial conditions. We lack convergence results and other analytic error bounds that could guide us to the applicability of the method. We hope to make progress on these and other numerical and theoretical issues in future studies.

References


