LETTER Communicated by Anthony Burkitt

Estimating Parameters of Generalized Integrate-and-Fire Neurons from the Maximum Likelihood of Spike Trains

Yi Dong
doyen@pha.jhu.edu
Stefan Mihalas
mihalas@jhu.edu
Department of Neuroscience and Mind/Brain Institute, Johns Hopkins University, Baltimore, MD 21218, U.S.A.

Alexander Russell
alexrussell@jhu.edu
Ralph Etienne-Cummings
retienne@jhu.edu
Department of Electrical and Computer Engineering, Johns Hopkins University, Baltimore, MD 21218, U.S.A.

Ernst Niebur
niebur@jhu.edu
Department of Neuroscience and Mind/Brain Institute, Johns Hopkins University, Baltimore, MD 21218, U.S.A.

When a neuronal spike train is observed, what can we deduce from it about the properties of the neuron that generated it? A natural way to answer this question is to make an assumption about the type of neuron, select an appropriate model for this type, and then choose the model parameters as those that are most likely to generate the observed spike train. This is the maximum likelihood method. If the neuron obeys simple integrate-and-fire dynamics, Paninski, Pillow, and Simoncelli (2004) showed that its negative log-likelihood function is convex and that, at least in principle, its unique global minimum can thus be found by gradient descent techniques. Many biological neurons are, however, known to generate a richer repertoire of spiking behaviors than can be explained in a simple integrate-and-fire model. For instance, such a model retains only an implicit (through spike-induced currents), not an explicit, memory of its input; an example of a physiological situation that cannot be explained is the absence of firing if the input current is increased very slowly. Therefore, we use an expanded model (Mihalas & Niebur, 2009), which is capable of generating a large number of complex firing patterns while still being linear. Linearity is important because it maintains the distribution of the random variables and still allows maximum likelihood

methods to be used. In this study, we show that although convexity of the negative log-likelihood function is not guaranteed for this model, the minimum of this function yields a good estimate for the model parameters, in particular if the noise level is treated as a free parameter. Furthermore, we show that a nonlinear function minimization method (r-algorithm with space dilation) usually reaches the global minimum.

1 Introduction

Lapicque's (1907) leaky integrate-and-fire (LIF) neuron model became very popular for the study of networks of spiking neurons. The model neglects the details of the rapid spike generation process while capturing subthreshold integration dynamics (Burkitt, 2006). LIF neurons are thus more realistic than the simplest neural models, like McCulloch-Pitts neurons or coincidence detectors (McCulloch & Pitts, 1943; Mikula & Niebur, 2003, 2004, 2008). However, in the traditional LIF neuron, the membrane voltage is reset to a predetermined value after each spike, making the model incapable of explaining many behaviors that are observed in biological neurons. To a limited extent, history dependence of spike times can be introduced by adding spike-induced currents (Hille, 1992). The model can then explain some more complex behaviors, like spike frequency adaptation and some bursting, and it can predict spiking times recorded in some physiological experiments reasonably well (Kim et al., 2009).

As a matter of principle, the introduction of spike-induced currents cannot explain phenomena that are rooted in the history of the input to the neuron. Some of these phenomena are anode break spiking, threshold adaptation, and the absence of firing when the membrane voltage is increased very slowly. Several extensions of the LIF model generate more biophysically realistic firing patterns (Izhikevich, 2003; Brette & Gerstner, 2005; Tonnelier, Belmabrouk, & Martinez, 2007; Mihalas and Niebur, 2009). In this letter, we focus on the model introduced by Mihalas and Niebur (2009), who extended the LIF neuron model by adding a variable threshold, together with an arbitrary number of spike-induced currents, all with linear dynamics. The reason is that this is the only one of the cited extensions of the LIF models that maintains linearity of the state-variable dynamics and thus allows easy application of the maximum likelihood methods that we will use. Among the rich behavioral repertoire of this model are spiking and bursting; tonic, phasic, or adapting responses; and depolarizing or hyperpolarizing afterpotentials and others (see Figure 1 of Mihalas & Niebur, 2009). The equations governing the time evolution of the membrane potential are solved analytically between spiking, allowing (but not requiring) event-based implementations of the model (Mihalas, Dong, von der Heydt, & Niebur, 2011). Furthermore, the analytical solution is obtained very efficiently since the model dynamics can be written as a diagonalizable set of linear differential equations.

An important problem in computational neuroscience is the determination of the parameters of a neuronal model given the available experimental data (Prinz, 2007). We assume that only the input to a neuron (the total injected current) and its output, the spike train (i.e., the sequence of time points when a neuron spikes), are available, not its transmembrane voltage. This corresponds to a current clamp experiment where the injected current is known and the neuron is free to spike, or to other situations where the input current is known. Parameter optimization is then accomplished by minimizing a suitable cost function. Choosing a cost function is not easy; for instance, using the benchmark for spike train prediction proposed by the INCF Quantitative Single-Neuron Modeling 2009 Competition, the so-called Γ-factor (Kistler, Gerstner, & van Hemmen, 1997; Jolivet et al., 2008), as the cost function results in large numbers of local minima. Finding the global minimum then requires substantial computational resources (Rossant, Goodman, Platkiewicz, & Brette, 2010).

A natural cost function for parameter fitting is the maximum likelihood estimator of the observed sequence of spike times (Brown, Barbieri, Eden,& Frank, 2003). Paninski, Pillow, and Simoncelli (2004) showed that the negative log-likelihood function of the stochastic LIF neuron model is concave and that its unique global minimum can thus be reached using gradient descent techniques. Application of this method to understanding biological neural activity is, however, limited by the small behavioral repertoire of this model. Assuming that noise appears only in the threshold, we have shown that the likelihood function of the generalized LIF model (Mihalas & Niebur, 2009) can be computed by solving a one-dimensional Fokker-Planck equation with the threshold as stochastic variable (Russell et al., 2010). In this study, we first generalize that result by deriving the Fokker-Planck equation of the likelihood function with noise in both membrane voltage and threshold. For computational efficiency, we then derive a reduced one-dimensional Fokker-Planck equation for the likelihood function, in this case assuming noise only in the membrane voltage. Subsequently we show that numerical accuracy is substantially improved by computing the likelihood function of the generalized LIF model by a Volterra integral equation method (Buonocore, Nobile, & Ricciardi, 1987; Plesser & Tanaka, 1997; Paninski, Haith, & Szirtes, 2008). Convexity of the negative log-likelihood function is not guaranteed after inclusion of the variable threshold, but we show that in practice, common optimization methods converge for the generalized LIF neuron.

2 Methods

2.1 Neuronal Model. The membrane voltage V in the stochastic LIF model evolves according to the dynamics

$$CV'(t) = -g(V(t) - V_{rest}) + I_{sti}(t) + I_{ind}(t) + \sigma\varepsilon(t), \tag{2.1}$$

where $I_{sti}(t)$ is the input current and $I_{ind}(t)$ is the sum of all spike-induced currents. The gaussian white noise, $\varepsilon(t)$, is scaled by the constant σ, and C, g, and V_{rest} are constants representing the capacitance, membrane conductance, and equilibrium membrane voltage, respectively. A spike is generated and the voltage is reset when $V(t)$ exceeds the threshold Θ, a constant. Paninski et al. (2004) showed that the negative log-likelihood function of generating a given (observed) spike train by this model is convex and that its global minimum can therefore be found using gradient descent techniques.

Considering the threshold $\Theta(t)$ to be dependent on the instantaneous membrane potential is important for the realization of a rich spiking behavior. Mihalas and Niebur (2009) introduced the following threshold behavior,

$$\Theta'(t) = a(V(t) - V_{rest}) - b(\Theta(t) - \Theta_\infty), \tag{2.2}$$

where a, b are threshold adaptation and threshold rebound constants, respectively, and Θ_∞ is the threshold at equilibrium. The dynamics of the threshold introduces history dependence, that is, correlations between interspike intervals. The model also allows an arbitrary number N of spike-induced currents, given by

$$\begin{aligned} I_{ind}(t) &= \sum_{i=1}^{N} I_i(t), \\ I_i'(t) &= -I_i(t)/\tau_i. \end{aligned} \tag{2.3}$$

When V reaches threshold, $V(t) = \Theta(t)$, three types of events occur. First, a spike is generated. Second, voltage and threshold are set according to the following reset conditions,

$$\begin{aligned} V(t) &\to V_{reset}, \\ \Theta(t) &\to \max(\Theta(t), \Theta_\infty), \end{aligned} \tag{2.4}$$

ensuring that after the update, the membrane potential $V(t)$ is below the instantaneous threshold $\Theta(t)$. Third, the spike-induced currents $I_i(t)$ are updated by the following rules,

$$I_i(t) \to R_i I_i(t) + A_i, \tag{2.5}$$

where A_i is the magnitude of the added current. In this study, we use a simplified version, always choosing $R_i = 1$. This model is capable of generating a much larger variety of neuronal behaviors than the traditional LIF model (Mihalas & Niebur, 2009).

2.2 Likelihood of a Spike Train. Assume that a neuron receives a known stimulus (input) current and in response generates a spike train. Assume further that we decide to model it as a generalized LIF neuron. How do we choose the parameters of the model? A natural estimator of the parameters is given by selecting them such that the observed spike train is the most likely for the given input current. We thus need to maximize the likelihood of observing the spike train or, equivalently (but technically more convenient), minimize its negative log likelihood. In this section, we compute the likelihood for the whole spike train as a function of the individual interspike intervals (note that this computation would be trivial if the interspike intervals were independent, which is not so in the general case we are concerned with here). In the following section, we compute the likelihood for individual interspike intervals.

The stimulus current starts at time t_0 and continues throughout the trial. The spike train is composed of a total of n spike time intervals, with the ith of them starting at time t_{i-1} and ending at time t_i. Define the conditional probability that the neuron fires at time t, which is in the ith interval, given the previous spike times intervals, as

$$f_i(t|t_k, k = 1, \ldots, i-1), t_{i-1} \le t < t_i. \tag{2.6}$$

Equivalently, the function f_i is the probability density function of the first passage time in the ith spike time interval. We define E_i for $i > 1$ as the event that the neuron fires at times t_{i-1} and t_i (and not in between); event E_1 consists of the stimulus onset at time t_0 and the first spike at t_1. The likelihood of the observed spike train is that all events $E_1, \ldots, E_n$ happen, which is $P(E_1 E_2, \ldots, E_n)$. From the multiplication rule in probability theory, we know that

$$P(E_1 E_2, \ldots, E_n) = P(E_1)P(E_2|E_1)P(E_3|E_1 E_2) \ldots P(E_n|E_1 \ldots E_{n-1}) \tag{2.7}$$

where

$$P(E_i|E_1, \ldots, E_{i-1}) = \frac{P(E_1 E_2, \ldots, E_i)}{P(E_1 E_2, \ldots, E_{i-1})} = \frac{P(\bigcap_{k=1,2,\ldots,i}\{S_k(t) < 0\} \cap \{S_k(t_k) = 0\} \cap \{S_k(t_{k-1}) = V_{reset} - \Theta(t_{k-1})\})}{P(\bigcap_{k=1,2,\ldots,i-1}\{S_k(t) < 0\} \cap \{S_k(t_k) = 0\} \cap \{S_k(t_{k-1}) = V_{reset} - \Theta(t_{k-1})\})} \tag{2.8}$$

The first and second equalities in equations 2.8 derive, respectively, from the definitions of conditional probabilities and the events E_i. The variable $S_k(t) := V(t) - \Theta(t)$ is the difference between voltage and threshold, and the

variable t in this equation is understood to be in the kth spike time interval, $t \in [t_{k-1}, t_k]$. The likelihood that the neuron fires at time t_i given its history is obtained by evaluating the conditional probability in equation 2.6 at time t_i, yielding

$$P(E_i|E_1, \ldots, E_{i-1}) = f_i(t_i|t_k, k = 1, \ldots, i-1). \tag{2.9}$$

We show in section 2.3 how $f_i(t_i|t_k, k = 1, \ldots, i-1)$ is computed. Using equation 2.9 in equation 2.7, the likelihood of the train of n spikes is then

$$P(E_1 E_2, \ldots, E_n) = \prod_{i=1}^{n} f_i(t_i|t_k, k = 1, \ldots, i-1), \tag{2.10}$$

which is the cost function used in the likelihood estimation.

2.3 Likelihood of a Single Spike Time Interval. Having obtained the likelihood for the spike train as a function of the likelihoods of the individual interspike intervals, we now compute the latter—the functions $f_i(t_i|t_k, k = 1, \ldots, i-1)$. This can be achieved by two methods[1] proposed by Paninski et al. (2004, 2008). The first, discussed in section 2.3.1, is by solving the Fokker-Planck equation for the probability density function of the state variables (voltage and threshold; the time dependence of the threshold requires a slight generalization of the approach of Paninski et al., 2004). It is expensive to compute, and we therefore introduce, in section 2.3.2, a more efficient one-dimensional approximation for the Fokker-Planck equation. The second method, discussed in section 2.3.3, is by solving a Volterra integral equation directly for the likelihood of the interspike interval.

2.3.1 Two-Dimensional Fokker-Planck Equation. For simplicity, we introduce the change of variables $V(t) \to CV(t)$, $\Theta(t) \to C\Theta(t)$ and $g \to g/C$. Defining as before $S = V - \Theta$ (the difference between voltage and threshold) and

$$I(t) = (g + a)V_{rest} - b\Theta_\infty + I_{sti}(t) + I_{ind}(t),$$

we substitute $V = S + \Theta$ in equations 2.1 and 2.2 and obtain

$$\begin{aligned} S'(t) &= -(g + a)S + (b - g - a)\,\Theta(t) + I(t) + \sigma\varepsilon(t), \\ \Theta'(t) &= a(S - V_{rest}) + (a - b)\,\Theta(t) + b\Theta_\infty. \end{aligned} \tag{2.11}$$

[1]Paninski et al. (2004) proposed a third method: directly evaluating the gaussian integrals occurring in the likelihood functions. This method, however, provides only a crude approximation to the likelihood and furthermore is not efficient to calculate. We will not discuss it any further in this letter.

Now define $P(S, \Theta, t)$ as the probability density function for the variables S and Θ at time t. The evolution of $P(S, \Theta, t)$ is governed by the two-dimensional Fokker-Planck equation:

$$\frac{\partial P(S,\Theta,t)}{\partial t} = -\frac{\partial}{\partial S}[(-(g+a)S + (b-g-a)\Theta(t) + I(t))P] + \frac{\sigma^2}{2}\frac{\partial^2 P}{\partial S^2} - \frac{\partial}{\partial \Theta}[(a(S - V_{rest}) + (a-b)\Theta(t) + b\Theta_\infty)P]. \tag{2.12}$$

There are two boundary conditions. One is absorbing,

$$P(0, \Theta, t) = 0, \tag{2.13}$$

since the membrane voltage can cross the threshold only once. The other is the resetting condition at spike time t_i. We show in appendix A that this condition is

$$P(S,\Theta,t_i) = \frac{1}{Z}\delta(V_{reset} - \Theta - S)R\left(-\left.\frac{\partial P(S,\Theta,t_i^-)}{\partial S}\right|_{S=0} \left((b-g-a)\Theta(t_i^-) + I(t_i^-)\right) + \frac{1}{2}\left.\frac{\partial^2 P(S,\Theta,t_i^-)}{\partial S^2}\right|_{S=0}\right),$$

where $\delta()$ is the delta function, t_i^- is the time immediately before the reset happens, and R is a linear rectifier:

$$R(u) = \begin{cases} 0 & \text{if u} \le 0 \\ u & \text{if u} > 0 \end{cases}. \tag{2.14}$$

Finally, Z is defined in appendix A to normalize the probability. Given $P(S, \Theta, t)$ for the ith spike time interval, the likelihood of firing at time t_i is then computed from P as

$$f_i(t_i | t_k, k = 1, \ldots, i-1) = -\left[\frac{\partial}{\partial t}\int_{-\infty}^{\infty} d\Theta \int_{-\infty}^{0} dS\, P(S,\Theta,t)\right]_{t=t_i}.$$

In practice, the 2D Fokker-Planck equation method computes the likelihood function accurately, but it is not very efficient and therefore of limited use. In the following section, we introduce a lower-dimensional Fokker-Planck equation that results in a more efficient solution; it is the latter that will we use in section 3.2.

2.3.2 One-Dimensional Fokker-Planck Equation. The numerical computation of the partial differential equation, equation 2.12, is time-consuming since it requires calculating the probability density in the two-dimensional (S, Θ) space. One way to reduce the complexity of the problem is to use a deterministic threshold $\Theta_d(t)$, which is computed from the deterministic version of equation 2.11:

$$\begin{aligned} S_d'(t) &= -(g+a)S_d(t) + (b-g-a)\Theta_d(t) + I(t), \\ \Theta_d'(t) &= a(S_d(t) - V_{rest}) + (a-b)\Theta_d(t) + b\Theta_\infty. \end{aligned} \tag{2.15}$$

The variable S remains a stochastic process, governed by

$$S'(t) = -(g+a)S + (b-g-a)\Theta_d(t) + I(t) + \sigma_1 \varepsilon(t), \tag{2.16}$$

where σ_1 is the noise level for the one-dimensional problem. Note that the deterministic threshold $\Theta_d(t)$ from equation 2.15 is used together with equation 2.16. The probability density then depends on only one variable, S, and the Fokker-Planck equation is one-dimensional,

$$\frac{\partial P(S,t)}{\partial t} = -\frac{\partial [V_s(t)P(S,t)]}{\partial S} + \frac{\sigma_1^2}{2}\frac{\partial^2 P(S,t)}{\partial S^2} \tag{2.17}$$

where

$$V_s(t) = -(g+a)S(t) + (b-g-a)\Theta_d(t) + I(t).$$

There is only one boundary condition, and it is again absorbing: $P(0,t) = 0$. As shown in appendix B, the stochastic term of $S(t)$ in the 2D case is

$$\sigma \int_{t_{i-1}}^{t} \left(\frac{b-g-a}{b-g} e^{g(s-t)} + \frac{a}{b-g} e^{b(s-t))} \right) d\varepsilon(s), \tag{2.18}$$

and in the 1D case, it is

$$\sigma_1 \int_{t_{i-1}}^{t} e^{(g+a)(s-t)} d\varepsilon(s). \tag{2.19}$$

Note that the only difference between the 1D and 2D cases for the stochastic process $S(t)$ is the noise model. Both noise terms are gaussian processes, but the structure is more complex in the 2D case because of the additional exponential function. We show in appendix B that when t is long enough, its variance saturates at

$$\frac{\sigma^2}{2}\left(\frac{(-1+\alpha)^2}{g} - \frac{4(-1+\alpha)\alpha}{b+g} + \frac{\alpha^2}{b} \right), \tag{2.20}$$

where $\alpha = \frac{a}{b-g}$. Finally, we show in appendix B that the variance of the 1D noise term saturates in the same limit at

$$\frac{\sigma_1^2}{2(g+a)} \tag{2.21}$$

and that the noise levels in the 1D and 2D conditions become comparable if the following condition is met:

$$\sigma_1^2 = \sigma^2(g+a)\left(\frac{(-1+\alpha)^2}{g} - \frac{4(-1+\alpha)\alpha}{b+g} + \frac{\alpha^2}{b}\right). \tag{2.22}$$

Given $P(S,t)$ for the ith spike time interval, the likelihood is

$$f_i(t_i|t_k, k = 1, \ldots, i-1) = -\frac{\partial}{\partial t}\int P(S,t)dS|_{t=t_i}.$$

We show in section 3 that for the optimization of deterministic neuronal models, details of the noise model do not matter. Noise is treated as an auxiliary parameter in the optimization by starting with a large value and then decreasing it. The optimized parameters obtained by the 1D Fokker-Planck equation method and the 2D Fokker-Planck equation method converge in the case of infinitesimally small noise. For reasons of efficiency, only the 1D Fokker-Planck equation is solved in section 3.2.

2.3.3 Volterra Integral Equation. Since the time derivatives of the state variables S, Θ depend linearly on the variables in equations 2.11, the Mihalas and Niebur (2009) model is an Ornstein-Uhlenbeck process. Schrödinger (1915) showed that its likelihood for the first-passage time can be obtained by solving a Volterra integral equation of the second kind.

Using the same notation as in section 2.2, the likelihood for the ith spike time interval is the first-passage-time probability density (FPTPD) function evaluated at the neuron spike time t_i (Buonocore et al., 1987; Paninski et al., 2008). To improve computational efficiency (see section 2.3.2), we assume that noise is injected only in the voltage, not the threshold (this approach is complementary to the one we took in Russell et al., 2010, where we treated the threshold as stochastic variable), that is, in the following, we replace the stochastic threshold by the deterministic threshold $\Theta_d(t)$.

The equation governing $S(t)$ is the same as equation 2.16,

$$\begin{aligned} S'(t) = &-(g+a)S + (b-g-a)\Theta(t) + (g+a)V_{rest} - b\Theta_\infty \\ &+ I_{sti}(t) + I_{ind}(t) + \sigma\varepsilon(t). \end{aligned} \tag{2.23}$$

We choose the initial condition for the voltage relative to threshold as[2] $S(t_{i-1}) = x$. From equation 2.23, the average of the $S(t|x, t_{i-1})$ process in the ith interspike interval is then

$$\mu(t \mid x, t_{i-1}) = S_{rest} + (x - S_{rest})e^{-(g+a)(t-t_{i-1})} + \int_{t_{i-1}}^{t} I(s)e^{-(g+a)(t-s)}ds, \tag{2.24}$$

with the definition $S_{rest} := V_{rest} - \frac{b}{g+a}\Theta_\infty$. Taking the expectation value of the square of formula 2.19, we obtain the variance of the $S(t|x, t_{i-1})$ process as[3]

$$\Sigma^2(t \mid t_{i-1}) = \frac{\sigma^2}{2(g+a)}\left(1 - e^{-2(g+a)(t-t_{i-1})}\right). \tag{2.25}$$

Since $S(t)$ is an Ornstein-Uhlenbeck process, it is gaussian. Its transient probability $G(y, t \mid x, t_{i-1})$ is the probability that $S(t) = y$ given $S(t_{i-1}) = x$:

$$G(y, t \mid x, t_{i-1}) = \frac{1}{\sqrt{2\pi}\Sigma(t \mid t_{i-1})} \exp -\frac{(y - \mu(t \mid x, t_{i-1}))^2}{2\Sigma^2(t \mid t_{i-1})}. \tag{2.26}$$

Buonocore et al. (1987) showed that the FPTPD function $p(t)$ for the stochastic process $S(t)$ can be computed by a Volterra integral equation of the second kind,[4]

$$p(t) = -2\varphi(0, t \mid S_0, 0) + 2\int_{i-1}^{t} \varphi(0, t \mid 0, s)p(s)\,ds, \tag{2.27}$$

where $\varphi(0, t \mid x, t_{i-1})$ is the probability current with the singularity removed as proposed by Buonocore et al. (1987). As we show in appendix C, it is expressed as

$$\varphi(0, t \mid x, t_{i-1}) = \frac{1}{2}G(0, t \mid x, t_{i-1})\Big(-(g+a)S_{rest} - I(t) + \frac{\sigma^2}{\Sigma(t \mid t_{i-1})^2}\mu(t \mid x, t_{i-1})\Big). \tag{2.28}$$

[2]Under the dynamics defined in equations 2.4, x is always given by $x = V_{reset} - \max(\Theta(t_{i-1}), \Theta_\infty)\ \forall i \geq 1$ but equation 2.24 and the following expressions are valid for all values of x.

[3]The numerically costly evaluation of the exponential function in equation 2.25 can be avoided by transforming it into a differential equation, $\frac{d\Sigma}{dt} = -2(g+a)\Sigma + \sigma^2$.

[4]An equivalent equation for constant input was derived and solved by Schrödinger (1915). We also note the solution by Plesser and Tanaka (1997) in terms of a Volterra integral equation of the first kind.

In our previous study (Dong, Mihalas, & Niebur, 2011), we showed that calculating the FPTPD function by discretizing the probability current φ can result in large numerical errors if the noise is small. The problem is solved by analytically calculating the mean probability current $\bar{\varphi}$ over the bin size, that is, the time interval Δt, as follows. We assume that only $G(0, t|x, t_{i-1})$ varies within the time bin $[t', t' + \Delta t]$; thus,

$$C_0 = -(g + a)S_{rest} - I(t') + \frac{\sigma^2}{\Sigma(t' \mid t_{i-1})^2}\mu(t' \mid x, t_{i-1}) \tag{2.29}$$

is a constant. In the function $G(0, t|x, t_{i-1})$, we introduce the notational shortcuts $\mu_0 := \mu(t' \mid x, t_{i-1})$ and $\mu_1 := \mu(t' + \Delta t \mid x, t_{i-1})$ and consider $\mu(t \mid x, t_{i-1})$ to vary linearly within the interval $[\mu_0, \mu_1]$. These approximations allow us to integrate the probability current through the interval $[t', t' + \Delta t]$ analytically, making use of its gaussian shape. Defining

$$\xi = \frac{\mu_0}{\sqrt{2}\Sigma(t'|t_{i-1})}, \tag{2.30}$$

$$E = \frac{\mu_1 - \mu_0}{\sqrt{2}\Sigma(t'|t_{i-1})} \tag{2.31}$$

the mean probability current,

$$\bar{\varphi}(0, t \mid x, t_{i-1}) = \frac{1}{\Delta t}\int_{t'}^{t'+\Delta t} \varphi(0, \tau \mid x, t_{i-1})\, d\tau, \tag{2.32}$$

can be simplified as

$$\bar{\varphi}(0, t \mid x, t_{i-1}) = \frac{C_0}{4(\mu_1 - \mu_0)}(\mathrm{erf}(\xi + E) - \mathrm{erf}(\xi)), \tag{2.33}$$

where erf() is the error function.

Using $\bar{\varphi}$ instead of φ in equation 2.27, we can compute the FPTPD accurately for a wide range of noise levels (see Dong et al., 2011, for details). Given the FPTPD function $p(t)$ for the ith spike time interval, the likelihood can then be evaluated as $f_i(t_i|t_k, k = 1, \ldots, i-1) = p(t_i)$.

2.4 Numerical Evaluation of Fokker-Planck and Integral Equations. The Fokker-Planck equation is a partial differential equation (PDE) that can be solved by finite volume methods. Briefly, state space is divided into discrete cells. Since total probability is conserved, the integrated probability flux through a cell's surfaces is equal to the probability change inside the cell. The probability density function within each cell, as well as its flux through the surfaces between cells, is computed. An exponential integration scheme (derived from the exact solution by exponentially interpolating each

subregion of the grid space) is used for the finite volume method to avoid negative probability densities (Figueiredo, 1997).

Paninski et al. (2008) described a numerical solution of the Volterra equation that yields the gradient of the likelihood function. It requires memory of size L^2, where L is the number of time bins. Since we have no need for the gradient, we instead use an algorithm that requires only memory of order L (to store the history of $\mu(t)$ and $\Sigma(t)$). We also compute μ (see equation 2.24) and Σ (see equation 2.25) by solving differential equations, rather than explicitly evaluating exponential functions (see footnote 3).

To increase computation speed, we use an optimization scheme developped in our earlier study (Dong et al., 2011). Briefly, time bins with negligible likelihoods can be easily identified if the membrane voltage is either far above or far below the threshold at both the beginning and the end of the bin. This is the case if

$$\xi = \frac{\mu_0}{\sqrt{2\Sigma^2(t'|t_{i-1})}} > M \quad \text{and} \quad \xi + E = \frac{\mu_1}{\sqrt{2\Sigma^2(t'|t_{i-1})}} > M \tag{2.34}$$

or

$$\xi < -M \quad \text{and} \quad \xi + E < -M, \tag{2.35}$$

where ξ and E were defined in equations 2.30 and 2.31 and M is the number of standard deviations beyond which the error function becomes indistinguishable from (positive or negative) unity in the floating number representation chosen. In our case (double precision accuracy), $M = 5.9$. No computation is required in any time bin in which the criteria of equations 2.34 or 2.35 are satisfied since within machine precision, this bin does not contribute to the likelihood. For low noise, this drastically decreases the effective number of time bins where the function is evaluated, which considerably decreases memory usage and computation time (see Dong et al., 2011, for details).

2.5 Optimization Methods. To determine the local minimum in the case of synthetic spike trains (see section 3.2), we use the r-algorithm with adaptive space dilation (Shor, Kiwiel, & Ruszcayǹski, 1985), implemented as the *ralg* algorithm at http://openopt.org. This heuristic algorithm is linearly convergent in the number K of arguments of the cost function and the number of model parameters subject to optimization, and it has been conjectured that this holds for general K (Burke, Lewis, & Overton, 2008). To avoid runaway simulations, we limit the maximal numbers of function evaluation and iterations. In our experience, the algorithm always converged (without exception in over 1000 runs), that is, the set limits were exceeded in none of them (results shown in Figure 3).

For the model optimization in section 3.2.2 and for fitting the experimental data in section 3.3, we use an evolutionary annealing-simplex algorithm that combines the Nelder-Mead simplex algorithm with simulated annealing and evolutionary techniques (Efstratiadis & Koutsoyiannis, 2002). It starts with a large population of vertices out of which a simplex is formed by random selection. The Nelder-Mead algorithm is then run on this simplex.

We optimized the neuronal model with 11 parameters, including some nonlinear terms like adaptation and absolute refractory period. Parameters are listed in Table 2.

3 Results

3.1 Accuracy of Likelihood Calculation. In this section, we compare the performance of the Fokker-Planck equation method with that of the Volterra integral equation method.

The first passage time probability density function of the model neuron is calculated by the Fokker-Planck method (FP in the legend of Figure 1C) and the Volterra integral method; both are described in section 2.4. The ground truth is obtained from a Monte Carlo scheme (MC) by simulating 1 million instances of the stochastic process $S(t)$, all starting with the same parameters but different random number seeds and computed from equation 2.23 with $\sigma = 10^{-11}$.

Two types of currents were provided to the model neuron. One is constant (discussed in the next paragraph), and the other is time varying. The latter, shown in Figure 1A, is the initial part of the mixed excitatory and inhibitory current defined in section 3.2.1 (the complete time course of this current is shown in Figure 2). For this input current, which vanishes at $t = 0$, the FPTPDs calculated by all three methods agree (see Figure 1C). It is known that a large ratio of drift to diffusion (the Péclet number) results in large numerical errors (Odman, 1997). Although diffusion is small at $t \approx 0$ (see Figure 1G), the drift is too, as is the Péclet number, and numerical errors are small. The variances of $S(t)$ calculated from Figures 1E and 1F agree with the result from equation 2.25.

If, on the other hand, a fairly strong constant current is injected from the beginning (see Figure 1B), the Fokker-Planck method is less accurate than the Volterra integral method (see Figure 1D). Figures 1E and 1F show the evolution of the probability density function evaluated at the grid points (S, t). As shown in Figure 1H, the variance of $S(t)$ calculated from the probability density function Figure 1F, is larger than the theoretical value (obtained from equation 2.25).

The Volterra integral method is not subject to this limitation since it does not require sampling of S. However, it is slower to calculate since its complexity increases quadratically in time, while the 1D Fokker-Planck

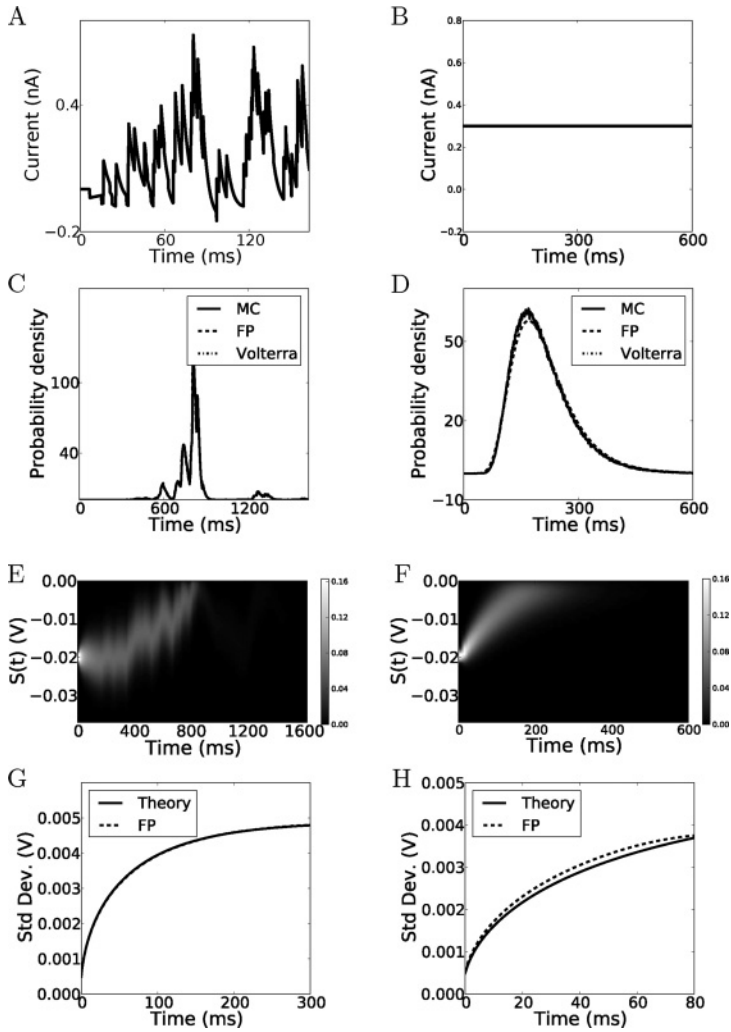

Figure 1: Numerical calculation of the likelihood function. Input current to the model neuron can be time variant (A) or constant (B). The FPTPD functions are calculated by three methods—Fokker-Planck (FP), Volterra, and Monte Carlo (MC) methods—and are shown for the time-variant (C) and constant (D) inputs. Note the agreement of all methods in *C* but not in *D* (see the text for details). The probability density functions (bin size 0.1 ms) for $S(t)$ calculated by the Fokker-Planck method are shown in *E* for the time variable and in *F* for constant input, and the variance of $S(t)$ for these two cases is shown in *G* and *H*, respectively. The curve labeled "Theory" is computed from equation 2.25. Note that the ranges of the abscissas are not identical; they were chosen to emphasize regions where the functions show the most significant changes.

method increases linearly.[5] Because of its superior accuracy, the Volterra integral method will be used in assessing the performance of the maximum likelihood model for the generalized LIF model in sections 3.2 and 3.3.

3.2 Validation of the Maximum Likelihood Estimator. To validate the method, in section 3.2.1 we will generate spike trains from the generalized integrate-and-fire neuron model (Mihalas & Niebur, 2009) with known parameters. In section 3.2.2, we then select random parameter values and attempt to find the correct ones, using the negative log likelihood for generating the original spike trains as a cost function. We will use both a local and a global optimization algorithm.

The local optimization serves to find, for each initial condition (randomly selected parameters), the local minimum closest to the respective initial values. This is a (somewhat rough) indicator of the usability of the cost function: in general, it is desirable that more rather than fewer initial conditions are in the basin of attraction of the global minimum rather than in that of a local, nonglobal one. Indeed, we will find that the two variations of the method (which will be introduced in section 3.2.2 and differ in the treatment of noise) that we analyze differ in this respect, with a substantially larger fraction of initial conditions ending up in the global minimum for one versus the other (about 3/4 versus 1/2).

In the second set of numerical experiments, we will use a global minimization algorithm. We show that it does indeed find the exact parameter values in all cases in which we used it. Finally, we show that our results are not sensitive to the details of the fitted parameters. This validates that the maximum likelihood method is suitable for parameter estimation for the class of problems studied here.

3.2.1 Setting Up the Validation Experiments. To generate the input current for the numerical experiment, two spike trains of 10 s length with homogeneous Poisson statistics were generated: one excitatory with mean firing rate 150 Hz and one inhibitory with rate 300 Hz. The input current to the neuron is then calculated by convolving the Poisson trains (each a sum of delta functions) with exponentially decaying functions of the form

$$Ae^{-t/\tau}H(t), \tag{3.1}$$

where A is the amplitude of the currents, τ is the decay time constant, and $H(t)$ is the Heaviside step function. For the excitatory input, we choose $A = 0.2$ nA and $\tau = 5$ ms, and for the inhibitory input, $A = -0.04$ nA and $\tau = 25$ ms. The injected input current is the sum of excitatory and

[5]The integration in equation 2.27 takes i steps in the ith iteration; therefore, computation to a time of order n takes $\approx (1+n)n/2$ steps, which is $O(n^2)$.

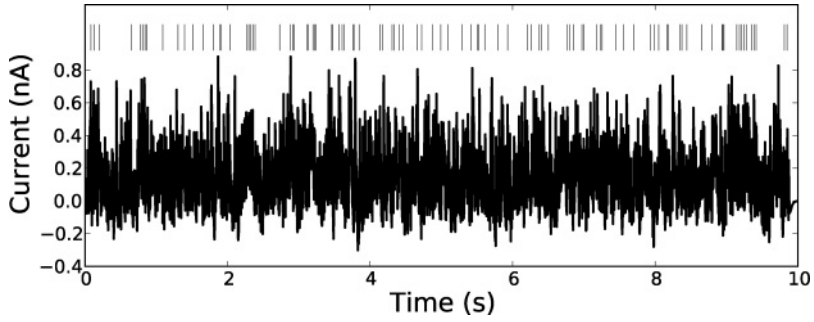

Figure 2: Input current for the model neuron. Neuron parameters are chosen to give bursting and adaptive behavior. Action potentials of the resulting spike train are shown as vertical lines above the input current.

Table 1: Parameters Used for the ABIF Model Neuron and Optimization.

	C[F]	g[S]	a[s^{-1}]	E[A]	I[A]	σ[As]
ABIF	2e-10	1e-8	2	4e-9	−1e-9	0
Upper bound	8e-10	4e-8	8	1.6e-8	4e-9	1.2e-11
Lower bound	0	0	0	−1.6e-8	−4e-9	1e-15
Fixed noise parameters	2.2e-10	9.9e-9	2.4	4.4e-9	−1.1e-9	-
Variable noise parameters	2.000e-10	1.000e-8	1.992	3.995e-9	−9.984e-10	9.952e-16

Notes: The first row shows the parameters used for generating spike trains (see Figure 2) without noise. The second and third rows show the lower and upper bounds of the parameter space, which is linearly mapped onto the scaled space (see text). The fourth row shows the parameters corresponding to the global minimum of the negative log likelihood in the fixed noise case ($\sigma = 5.5e - 12$). Note that values differ from the targets (in the ABIF row) due to the injected white noise. The last row shows the parameters found with the variable noise method. Values are very close to the targets, including the noise level.

inhibitory currents. The resulting current (shown in Figure 2) is injected in the generalized integrate-and-fire neuron (term I_{sti} on the right-hand side of equation 2.1), and the model generates the spike train shown in the figure. Note that the neuron model used here is deterministic, without any injected noise ($\sigma = 0$).

The parameters of the IF model (row ABIF in Table 1) were chosen to make it both adapting and bursting (thus, "ABIF"). The membrane resting potential and membrane resetting potential were fixed at −70 mV and the equilibrium threshold at −50 mV. The threshold rebound time constant was set to 100 ms and the absolute refractory period of the neuron to 2 ms. Parameters to be determined by the optimization process are capacitance C, membrane conductance g, and adaptation constant a. Furthermore, two spike-induced currents are included in the optimization: E is the amplitude

of an excitatory spike-induced current, with $\tau_1 = 5$ ms, and I that of an inhibitory spike-induced current, with $\tau_2 = 25$ ms. These are the time constants of the two components of the input current. While our method allows optimization of any parameter, including the time constants, the larger the number of free parameters is, the more difficult the optizimation. In many cases, current time constants are determined by the biophysics of their respective ion channels and thus known and invariable quantities, and we assume that this is the case here (see also section 4.2). All parameters subject to optimization are shown in Table 1, row ABIF.

3.2.2 Local and Global Minima. An important insight of Paninski (2004) was that the negative log-likelihood function of the stochastic LIF neuron is concave; it has only one minimum. This is not necessarily the case for more realistic models, like that described by equation 2.2. It is thus important to analyze the cost function (the negative log likelihood), and we find in the remainder of this section that in general, this model has multiple local minima. We also study two different cost functions, which differ in how stochasticity is treated, and we compare an indicator of how many local minima each of them has.

Likelihood estimation requires stochasticity of the neuron model, that is, noise needs to be added to the deterministic model. We consider two ways to include noise, which give rise to two different likelihood functions (and thus cost functions). The first is to fix the noise amplitude σ (at a low level) and optimize the model parameters (fixed noise case). The second is to formally treat the noise level as a parameter and optimize the solution for all parameters, including the noise level (variable noise case). Obviously, the true noise level is 0, and solutions that yield nonzero results are incorrect (not at the global minimum).

We used the r-algorithm with space dilation on 300 randomly selected initial locations in the parameter space defined below.[6] For each of these 300 points, we find the closest local minimum of the negative log likelihood as described in section 2.5. In each dimension, upper and lower boundaries of the search space are defined as ± 4 times of the absolute value of the true parameter; the nonlinear optimization procedure we use (r-algorithm) will not search outside these boundaries. In the variable noise case, the true noise level is 0, and we use the boundaries shown in rows 2 and 3 of Table 1. To ensure comparable scaling for the different parameters, we map the parameter space linearly into a multidimensional rectangle with edges $[-1, 1]$ for parameters with both positive and negative values (E, I), and with edges $[0, 1]$ for positive parameters (C, g, a, σ). The initial parameters for the optimization are drawn from a uniform distribution in

[6] A simple gradient descent method will run into problems because the likelihood function is not everywhere differentiable, for example, when the threshold is crossed.

this multidimensional rectangle (with the exception of the variable noise case, discussed in the next paragraph), and the distance in parameter space is defined as the sum of the squared distances along each axis in this space.

In the fixed noise case, we set the noise level to a small number,[7] $\sigma = 5.5e^{-12}$. Because of the injected white noise, the parameters corresponding to the global minimum differ from the correct values—those used to generate the original spike train (Table 1, row 4). Since the noise level is subject to optimization in the variable noise case, it is advantageous to use a larger initial noise level. Rather than drawing it uniformly from the full range given in Table 1, we choose it (uniformly) in the range $\sigma \in [9e^{-12}, 1.2e^{-12}]$, which is mapped in the transformed space to the interval $[0.75, 1]$.

We ran 302 independent optimizations for the fixed noise case and 303 independent optimizations for the variable noise case with the local optimization method. We also ran 434 independent optimizations for the variable noise case with the global optimization method.[8] As shown in Figure 3A, 49% of the optimizations successfully converged to the global minimum in the fixed noise case. This number was substantially higher, 73.9%, in the variable noise case (see Figure 3B). Furthermore, the computation time of the variable noise method was lower than for the fixed noise method. In the fixed noise case, the noise level is low but not negligible ($\sigma = 5.5e^{-12}$), and the number L of time bins for the function evaluation is essentially constant. In the variable noise case, the optimization starts with a large noise level and consequently runs slowly in the beginning. Over time, the true noise level (0) is approached, and the contribution to the total likelihood of more and more bins becomes negligible. These bins are then eliminated from the evaluation, as discussed in section 2.4, which increases the computation speed dramatically since the complexity of the algorithm is quadratic in the number of bins (see footnote 5). As a consequence, we find that in practice, variable noise optimization is an order of magnitude faster than fixed noise optimization.

We also find that fixing the noise level leads to systematic errors in the parameter estimation, as shown in Table 1, row 4. Furthermore, we find that in the example chosen, more points are trapped in a prominent local minimum in the fixed noise case than in the variable noise case, presumably

[7] We cannot choose this too small; otherwise, most of the small likelihood range for random initial parameters is beyond the range of double precision calculations. During the numerical computation, very small probabilities are bounded by the smallest possible number that can be represented in double precision. Thus, if the initial conditions are too far away from the correct values, the gradient of the negative log likelihood vanishes, and the optimization cannot find the minimum. This problem is alleviated in the variable noise case, where we can choose a larger initial noise level since the optimization converges on the correct (0) level. Furthermore, computation of the likelihood involves the subtraction of two error functions, which can lead to truncation errors (see Dong et al., 2011).

[8] The exact number of simulations is of no importance; we stopped when we were convinced that we had characterized the methods in a satisfactory manner.

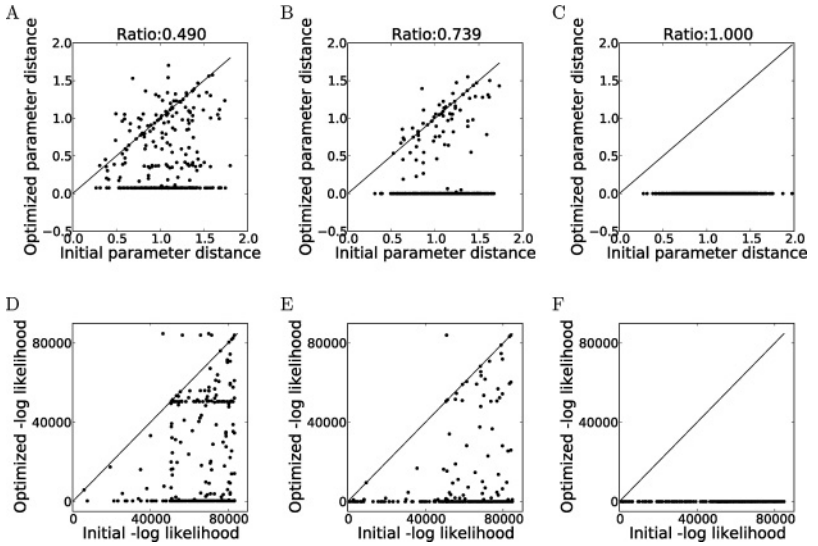

Figure 3: Numerical optimization results. (A, D) Fixed noise case. (B, E) Variable noise case with local optimization. (C, F) Variable noise case with global optimization. Panels A–C show the distance of initial and optimized parameters from the true values in the transformed parameter space. Panels D–F show the negative log likelihood before and after the optimization. The fraction of runs in which the global minimum was found is shown in the top line. In the fixed noise case, optimized parameters have generally larger errors than in the variable noise case due to the injected white noise. Parameters are also more likely to fall into a deep local minimum, visible as a horizontal line of points at negative log likelihood$\approx 50,000$ in panel D. Occasional data points (five in panel D, one in panel E) where optimized values of the negative log likelihood are greater than the original ones are due to limitations of the optimization algorithm; see the text.

due to the fact that the initial values of the likelihood for some of the parameters are too small to allow computation of their correct values (see note 7). This problem is alleviated in the variable noise case by starting with a larger noise level. The initial likelihood values are therefore larger, and the finite gradient of the likelihood function then allows accurate computation of all parameters.

In Figure 3, we note that in rare cases (five for fixed noise, one for variable noise), the negative likelihood actually increases during the optimization. These failures of the r-algorithm are due to the fact that it is a subgradient optimization method (Shor et al., 1985) which allows minimization of nondifferential functions. Given that the gradient of such a function does not

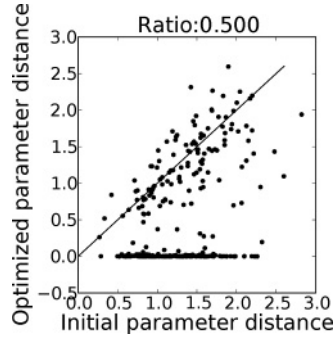

Figure 4: Same as Figure 3B, but with varying true parameters. See the text for details.

always exist, there is no guarantee that the solution always descends, and in the rare cases shown, it actually climbs.

To make sure that our results do not depend on the details of the chosen parameters, we repeated our analysis but with true parameters that were chosen randomly. They were drawn from the same distribution as the initial values in the previous optimization, see Table 1 (range given by rows 2 and 3). Initial values were chosen exactly as in Figure 3, and we ran the local optimization method with variable noise for 300 parameter sets. Since the parameters now varied over a large range, the number of spikes generated in a trial could vary widely. If the number of spikes generated is too small, there is not enough information to compute a "good" likelihood function. On the other hand, if the randomly chosen parameters generate too many spikes, the activity level is unrealistic (note that restricting ourselves to physiologically realistic firing rates is a conservative assumption; the more spikes are available, the easier the optimization becomes). We therefore limited our analysis to those sets of "true" parameter values that generated between 3 and 30 spikes per second (thus, between 30 and 300 spikes for the 10 second simulated interval). Results are shown in Figure 4, which is analogous to Figure 3B. Note that it would not make sense to also plot the distances between likelihoods, as in Figure 3E, since the true parameters are different for each of the points and likelihood distances cannot be compared for different true parameter values.

The local optimization found the global minimum somewhat less frequently than for fixed "true" parameters (0.5 versus 0.739). A possible reason is that in the fixed parameter case, the true parameter values were in the center of the interval from which the starting points were selected. When we varied the true parameters, this was no longer the case; therefore, the squared distance between starting points and true parameter was now larger, which may have led to the somewhat lower fraction of cases

in which the global minimum was reached. Nevertheless, our results show that likelihood maximization is suitable for optimal parameter search.

3.3 Fitting of Experimental Data. Having validated the model with synthetic test data generated from a model where we know the correct parameter values, we now apply the method to the estimation of model parameters for fitting experimental (biological) data.

The experimental data were taken from the Spike Prediction Competition 2009 Challenge B (Gerstner & Naud, 2009). A layer-5 fast-spiking cell was recorded with in vivo–like current injection. We used data from eight trials of 14 s length each for the training of the model and another 8 s for the prediction. We used the same generalized integrate-and-fire model as before and optimized 11 parameters, including the amplitudes of four spike-induced currents (their decay constants are kept fixed at 5 ms, 10 ms, 20 ms, 50 ms), absolute refractory period, noise level, threshold adaption, threshold rebound, membrane time constant, voltage resetting, and equilibrium membrane voltage. The boundaries for these 11 parameters are shown in Table 2, rows 1 and 2. The final optimized parameters are shown in row 3.

Training and prediction results are shown in Figures 5A and 5B. The benchmark proposed for the quality of the training and prediction by Jolivet et al. (2008) is the average spike time coincidence ($\Gamma-$ factor) between predicted spikes and measured spikes across eight repetitions, scaled by the intrinsic reliability of spiking across several trials. On the training data, the model explained 68.4% of the spikes, corresponding to a normalized performance (Jolivet et al., 2008) of 98.3%, with a confidence level (computed by bootstrap) of $\pm$4.79%. For prediction, our model explained 66.3% of the spikes, corresponding to a performance of 93.6%, confidence level $\pm$1.58%. We emphasize that our choice of which parameters to vary (top row of Table 2) and which not to vary (e.g., the number of spike-induced currents and their time constants) is based on insight into the biophysics of neuronal dynamics but can, in the end, be justified only by the success of the model in explaining the experimental data. Including spike-induced currents with the generic time constants of 5, 10, 20, and 50 ms explains the data for this specific neuron quite well, but, of course, other neuron types have other transmembrane currents and require other assumptions.

Note that we used the data from the spike prediction competition only as a conveniently available spike train data set. A direct comparison of the performance of our method with others used in the competition is not possible since we use only a small part of the training data, and we use it for both training and prediction (since the "prediction data" have been published, a genuine prediction is no longer possible). Nevertheless, we note that although our prediction results are lower than those of the winners of the competition, they are better than the majority of submitted results (7/11). Importantly, we optimize the neuronal model based on the likelihood function for the spike trains, without making any use of the

Table 2: Table of Parameters in the Optimization.

	C/g[s]	σ[As]	V_{rest}[V]	V_{reset}[V]	A_1[A]	A_2[A]	A_3[A]	A_4[A]	a[s^{-1}]	b(s^{-1})	REF(s)
Lower	0.001	1e-13	−0.14	−0.14	−1e-7	−1e-7	−1e-7	−1e-7	0.0	0.1	0.0
Upper	0.06	5e-12	−0.03	−0.03	1e-7	1e-7	1e-7	1e-7	20.0	100.0	0.011
Optimized	0.025	4.13e-12	−0.073	−0.067	1.92e-09	−5.30e-10	5.29e-11	1.78e-12	19.62	47.27	4.34e-3

Notes: Rows 1 and 2 show the box-bound of the optimized parameters. Row 3 shows the optimized parameter values. C/g is the membrane time constant of the neuron. REF is the absolute refractory period of the neuronal model. All other parameters are defined in section 2.1.

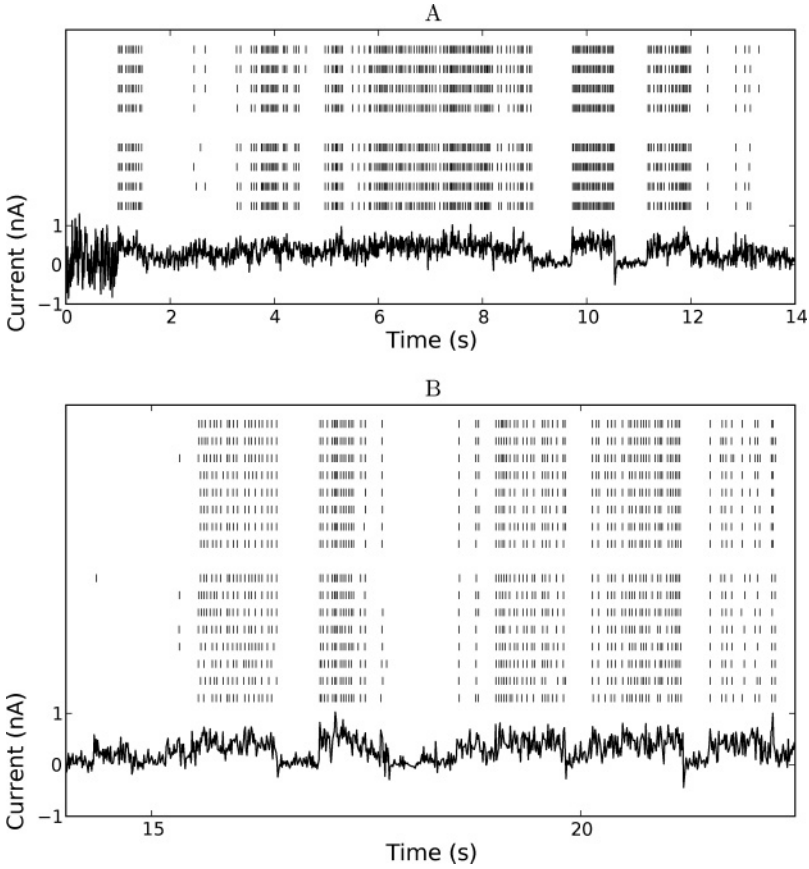

Figure 5: Fitting the neural model to experimental data. (A) Training data set. (B) Prediction data set. The bottom trace in each is the input current to the neuron. In each panel, the upper group of ticks represents experimentally recorded action potentials, with each row corresponding to one trial. The lower group of ticks represents action potentials generated by the model. Current injections are identical for all rows, but random seeds differ between rows.

definition of the $\Gamma-$ factor while the entries to the competition were tailored to achieve maximum performance under this measure.

4 Discussion

4.1 Maximum Likelihood for Parameter Estimation. For neuronal model optimization, the key question is the choice of the cost function. One possibility is to use the model for generating a set of spike trains and

then compute the "distance" between these and the observed spike trains using one of a variety of proposed spike train distance measures as a cost function. This approach has been used with great success by the group of S. Shinomoto (Gerstner & Naud, 2009), which won the International Competition on Quantitative Neuron Modeling in three straight years (2007–2009). Shinomoto shows that maximizing the coincidence between the model and data spike times, measured by the $\Gamma-$ factor (Kistler et al., 1997), the criterion adopted by the International Competition on Quantitative Neuron Modeling, can result in good agreement of the model and biological spike trains.

However, there are two drawbacks of using spike train distance measures. The first is that their cost function in general has many local minima, and therefore a stochastic global optimization method has to be used, which is time-consuming. The second is that spike train distance measures are not parameter free. Popular spike distance measures (Victor & Purpura, 1996; van Rossum, 2001; Jolivet et al., 2008) have at least one freely chosen parameter. The choice of the free parameter could be made part of the optimization problem, but this increases the dimension of the problem even further and is not an approach commonly taken.

Maximum likelihood estimation solves both of these problems. The likelihood function is a natural measure of distance which is also parameter free. Furthermore, Paninski et al. (2004) showed that the negative log-likelihood function of the leaky integrate-and-fire neuron is convex. This guarantees the existence of a unique global minimum and also allows finding it using simple gradient descent methods. Unfortunately, the dynamics of the straightforward LIF model are not rich enough to explain the behavior of many biological neurons. The generalized LIF model (Mihalas & Niebur, 2009) generates a much richer dynamics, but the proof of convexity is not applicable to it. Indeed, our numerical experiments show that the likelihood function is encumbered with local minima; in our somewhat crude estimation, about 25% to 50% (depending on how stochasticity was treated; see section 4.3) of initial conditions were closer to the local minima rather than the global minimum. The global minimum is easily attained with a global optimization algorithm in all cases (see Figures 3C and 3F).

4.2 Known Knowns and Known Unknowns. Optimization of model parameters does not occur in an intellectual vacuum. As in all other model building, both explicit and implicit knowledge, skills, and assumptions are necessary to build successful computational models. The situation that we consider here and, we believe, is the practically relevant one, is that optimization is performed on a neuronal model based on some preexisting insight rather than in a tabula rasa situation in which nothing at all is known about the underlying system. One always has to start with some assumptions, for example, that the model to be used is of the integrate-and-fire type or that it has only one compartment, since the space of all

parameters is usually too large to be explored. Which parameters are subject to optimization and which are selected depends on knowledge about the neural system studied in particular and insight into properties of nervous systems in general.

In the two examples studied in this letter, the validation of synthetic data (see section 3.2) and the fitting of experimental data (see section 3.3), we decided to allow the variation of current amplitudes but not current time constants. We believe that this is typical for many real-world situations. It is frequently known which currents (e.g., resulting from synaptic inputs) are likely present and, furthermore, at least rudiments of the biophysical properties of these currents. In particular, their time dependence is frequently assumed as exponential, with time constants from a relatively limited range–for instance, glutamatergic AMPA conductance can always be approximated with time constants on the order of a few milliseconds, and glutamatergic NMDA channels have time constants on the order of 150 ms (for NR2A) or 500 ms (for NR2B). In contrast, current amplitudes are proportional to the number of ion channels of a given type and can change over orders of magnitude.

Another tool that severely constrains the parameter space to be explored is the separation of timescales. For example, in the synthetic data example (see section 3.2), spike frequency adaptation can be obtained by modifying either of the two spiking-induced currents we implement, or leak currents, or threshold adaptation. Similar considerations can be made about bursting, preferred frequencies, and refractoriness. By fixing the timescale of each channel (or searching for their variations in bounded nonoverlapping regions), observing the behavior in a particular timescale basically guarantees that the parameters at that timescale will be the most important in fitting this behavior. Since the subthreshold dynamics is linear in most parameters, it allows the behaviors at one timescale to be minimally influenced by parameters in channels at other timescales.

4.3 Fixed Versus Adaptive Noise Level. Finding the minimum of the negative log-likelihood function requires that the neuron model is stochastic, (i.e., the presence of nonzero noise). It is a matter of practicality how this noise is treated. One possibility is to fix the noise level to a small, finite number and find the minimum negative log likelihood for the model parameters. Alternatively, the noise can be introduced as an additional parameter that is itself subject to optimization. Although one might expect that increasing the dimensionality of the optimization problem (in the case of Table 1, from 5 to 6) would substantially decrease numerical efficiency, we found the opposite: optimizing the noise level simultaneously with the model parameters not only eliminates the need for choosing a somewhat arbitrary parameter but actually increases performance.

The reason for this seemingly counterintuitive result is that the choice of the noise level is nontrivial, and, indeed, the optimal level usually varies

during the course of the optimization process. If the noise level is chosen too small, the optimization is likely to stay in local minima. If it is set too large, estimated parameters will have a bias (see Table 1). The better solution is thus to start with a larger noise level and let the optimization process adaptively find the optimal level. Our results show that this procedure finds the global minimum substantially more frequently than if noise is chosen fixed. Furthermore, due to the speed optimization introduced earlier (Dong et al., 2011), we find that the adaptive noise level algorithm also runs considerably faster than if a fixed level of noise is chosen.

4.4 Efficiency and Accuracy of Numerical Likelihood Functions. Two evaluation methods for the likelihood function are compared in this study: Fokker-Planck equations and Volterra integral equations. The former uses a PDE solver to numerically solve the Fokker-Planck equation for the probability density of the time-dependent state variables (voltage and threshold). Its one-dimensional version (where the threshold is considered a deterministic, nonstochastic variable) has the advantage of fast calculation, and the complexity is $O(n)$. A weakness is that it suffers from numerical diffusion problems, and, as we show, it is less accurate than the solution of the Volterra integral equation. These problems are, in our experience, alleviated if interspike intervals are long compared to the membrane time constant, in which case the numerical diffusion is frequently small compared to actual diffusion caused by noisy current input.

The Volterra integral method consists of computing numerically the integral in a Volterra integral equation (of the second kind). It requires only discretization of the time variable. We find that this method is more accurate in the calculation of the likelihood function than the solutions of the Fokker-Planck equation; however, its complexity grows quadratically with the number of time bins, (i.e., the length of the interspike intervals).

These complementary properties of the two methods point to an obvious solution by combining them. For long interspike intervals, the Fokker-Planck method is fast, and its weakness (numerical diffusion) is masked by actual diffusion (at least for sufficiently complex input, which is usually found in biologically relevant circumstances). For short interspike intervals, the Volterra integral method is accurate, and its weakness (quadratic complexity in number of time bins) is of little importance. Thus, by using the Fokker-Planck equation method for long interspike intervals and the Volterra integral equation method for short intervals, we obtain a fast and accurate solution.

4.5 Limitations. While we believe that the algorithms proposed in this study are powerful tools for optimal parameter estimation, they are subject to limitations, like all other tools. Such limitations pertain

to both the underlying neuronal model and the maximum likelihood method.

Regarding the single neuron model, the generalized leaky integrate-and-fire model (Mihalas & Niebur, 2009) needs to be compared with both simpler and more complex models; application of Occam's razor suggests that the simplest model that explains a given phenomenon is preferable. If a simple integrate-and-fire neuron, with constant threshold, is sufficient to explain the behavior of a given neuron, then it should be preferred over the model used here, in particular since Paninski et al. (2004) showed that the negative log likelihood is a concave function and can be solved easily and efficiently. Note that the proof by Paninski et al. (2004) is applicable even in the presence of spike-induced currents. As far as more complex models are concerned, there are at least two different classes of limitations of the Mihalas & Niebur (2009) model. One is that the model cannot explain phenomena resulting from any type of current that cannot be described by a combination of spike-induced currents and an adaptive threshold. We emphasize that the phenomenology of the model is very rich (Figure 1 in their article shows a large number of explained neural behaviors), but that of nature is certainly richer. The second limitation is that the model is formulated as a point neuron. Thus, any phenomena that depend on the spatial extent of the neuron, for example, intradendritic computations (Polsky, Mel, & Schiller, 2004), backpropagating action potentials (Stuart, Schiller, & Sakmann, 1997), and many more—cannot be explained by this model.

A severe limitation of likelihood maximization is the computational effort needed to compute and optimize the likelihood, which can substantially exceed that of some alternative methods. One class of parameter estimation uses a computational model to generate (parameter-dependent) spike trains and then determines the parameters by minimizing the distance between observed and model-generated spike trains. We have already discussed in section 4.1 three examples of spike distance measures—the $\Gamma-$performance (Kistler et al., 1997; Jolivet et al., 2008), the Victor & Purpura (1996) distance—and the metric developed by van Rossum (2001) designed as a simple measure of the postsynaptic effects of the spiketrain. Typically all three are less costly to compute than the likelihood and therefore preferable when they are sufficient to solve a given problem. Shinomoto's group developed a series of models that won the International Competition on Quantitative Neuron Modeling three times in a row using the $\Gamma-$factor method, demonstrating that it is useful and sufficient in this case (Gerstner & Naud, 2009). In a separate study (under review), we find, however, that all three spike distance measures discussed are plagued by a large number of local minima. While their performance for simple leaky integrate-and-fire models may be acceptable, for more complex model neurons, the ease of computing a single distance is more than offset by the need to use costly global optimization algorithms.

5 Conclusion

Quantitative studies of biological nervous systems characterize their behavior in terms of analytical models whose parameters need to be fit to the available data. This requires choosing a model that is, on the one hand, complex enough to capture the essential properties of the biological neurons and, on the other hand, accessible for efficient and accurate parameter fitting. One example of a successful compromise is the leaky integrate-and-fire model, in particular, since Paninski et al. (2008) showed that its maximum likelihood for generating a specific spike train can be found by gradient ascent. The dynamics of this model are, however, too simple for many biological neurons. A generalized leaky integrate-and-fire model (Mihalas & Niebur, 2009) provides much richer dynamics, and we show in this letter that although simple gradient ascent techniques will not work, the maximum likelihood function of this model can be found readily and efficiently. We expect that this will allow the wide spread use of the powerful maximum likelihood methods to this important interface of experimental and theoretical neuroscience.

Appendix A: Boundary Condition of the 2D Fokker-Planck Equation

The Fokker-Planck equation 2.12 can be understood as a continuity equation,

$$\begin{aligned}\frac{\partial P(S,\Theta,t)}{\partial t} &= -\bigtriangledown \bullet J \\ &= -\frac{\partial J_S}{\partial S} - \frac{\partial J_\Theta}{\partial \Theta},\end{aligned} \tag{A.1}$$

for the two-dimensional probability current J. It has two components (J_S, J_Θ):

$$J_S = (-(g+a)S + (b-g-a)\Theta(t) + I(t))P - \frac{\sigma^2}{2}\frac{\partial P}{\partial S}, \tag{A.2}$$

$$J_\Theta = (a(S - V_{rest}) + (a-b)\Theta(t) + b\Theta_\infty)P. \tag{A.3}$$

Before calculating the boundary condition, we first try to find the conditional probability distribution of Θ, conditioned on $S(t) < 0, t < t_i$ and $S(t_i) = 0$. Intuitively, one might think it is $J_S(S = 0, \Theta, t_i)$, which is the probability of P going beyond $S = 0$ per unit time and per unit Θ at time t_i and threshold Θ. However, due to the absorbing boundary condition, the probability density vanishes at the threshold, $P(0, \Theta, t) = 0$. From equation A.3, we find $J_S(S = 0, \Theta, t_i) = 0$. Next, we consider the probability current at $S = -\Delta S$, which is the probability of P going beyond $S = -\Delta S$ per unit time and per unit Θ at time t_i. This current will push up the probability

density P in the range $S = [-\Delta S, 0]$. Due to the absorbing boundary condition, we consider the absorbed P only as "firing" probability. Absorption happens at the point $S = 0$ (not in an interval). To calculate the absorbed part of the probability density P at the single point $S = 0$, the absorbed probability can be approximated by

$$\frac{J_S(S = -\Delta S, \Theta, t_i)}{\Delta S} \tag{A.4}$$

if ΔS is very small. Then equation A.4 is the conditional probability of Θ. Considering $J_S(S = 0, \Theta, t_i) = 0$, equation A.4 can be expressed in another form:

$$-\frac{J_S(S = 0, \Theta, t_i) - J_S(S = -\Delta S, \Theta, t_i)}{\Delta S}. \tag{A.5}$$

In the limit $\Delta S \to 0$, the difference in equation A.5 becomes the derivative,

$$\begin{aligned}
&-\frac{\partial J_S(S, \Theta, t_i)}{\partial S}\bigg|_{S=0} \\
&= \left(-(-(g + a)S + (b - g - a)\Theta(t) + I(t))\frac{\partial P}{\partial S}\right)\bigg|_{S=0} \\
&\quad + ((g + a)P)\bigg|_{S=0} + \frac{\sigma^2}{2}\frac{\partial^2 P}{\partial S^2}\bigg|_{S=0} \\
&= \left(-((b - g - a)\Theta(t) + I(t))\frac{\partial P}{\partial S}\right)\bigg|_{S=0} + \frac{\sigma^2}{2}\frac{\partial^2 P}{\partial S^2}\bigg|_{S=0}.
\end{aligned} \tag{A.6}$$

Equation A.6 is not really a density distribution function; it may contain negative terms, and it is not normalized. Defining R as the usual linear rectifier,

$$R(u) = \begin{cases} 0 & \text{if u} \le 0 \\ u & \text{if u} > 0 \end{cases}, \tag{A.7}$$

we obtain the conditional density distribution function for Θ,

$$\frac{1}{Z} R\left(-\frac{\partial J_S(S, \Theta, t_i)}{\partial S}\bigg|_{S=0}\right), \tag{A.8}$$

where Z is the normalization factor,

$$Z = \int R\left(-\frac{\partial J_S(S, \Theta, t_i)}{\partial S}\bigg|_{S=0}\right) d\Theta. \tag{A.9}$$

Having obtained the conditional probability distribution function for Θ, computing the resetting boundary condition now simply involves mapping the density function onto the line $S = V_{reset} - \Theta$ in 2D space (S, Θ),

$$\begin{aligned} P(S,\Theta,t_i^+) &= \frac{1}{Z}\delta(V_{reset}-\Theta-S)R\left(-\frac{\partial J_S(S,\Theta,t_i^-)}{\partial S}\bigg|_{S=0}\right) \\ &= \frac{1}{Z}\delta(V_{reset}-\Theta-S)R\left(-\frac{\partial P(S,\Theta,t_i^-)}{\partial S}\bigg|_{S=0}\right. \\ &\quad \left.\big((b-g-a)\Theta(t_i^-)+I(t_i^-)\big)+\frac{1}{2}\frac{\partial^2 P(S,\Theta,t_i^-)}{\partial S^2}\bigg|_{S=0}\right), \end{aligned} \tag{A.10}$$

where $\delta()$ is the Dirac Delta function and t_i^+, t_i^- approach the spike times t_i from above and below, respectively.

Appendix B: 1D and 2D Noise Models for the Stochastic Process S

We find from equation 2.1 that the noise of the process $V(t)$ is

$$N_V(t) = \sigma e^{-tg}\int_{t_{i-1}}^{t} e^{gs}d\varepsilon(s). \tag{B.1}$$

Combining equations 2.1 and 2.2, we obtain an equation of S in terms of V:

$$\begin{aligned} S'(t) &= -bS + (b-g-a)V(t) + (g+a)V_{rest} - b\Theta_\infty + I_{sti}(t) \\ &\quad + I_{hist}(t) + \sigma\varepsilon(t). \end{aligned} \tag{B.2}$$

From equation B.2, we see that the noise of $S(t)$ is

$$\begin{aligned} N_S(t) &= \sigma(b-g-a)\int_{t_{i-1}}^{t} e^{-\tau g}\int_{t_{i-1}}^{\tau} e^{gs}d\varepsilon(s)e^{-b(t-\tau)}d\tau \\ &\quad + \sigma e^{-tb}\int_{t_{i-1}}^{t} e^{bs}d\varepsilon(s). \end{aligned} \tag{B.3}$$

Changing the order of integrations in the first term in equation B.3,

$$\begin{aligned} &\sigma(b-g-a)\int_{t_{i-1}}^{t} e^{-\tau g}\int_{t_{i-1}}^{\tau} e^{gs}d\varepsilon(s)e^{-b(t-\tau)}d\tau \\ &\quad = \sigma(b-g-a)e^{-tb}\int_{t_{i-1}}^{t}\int_{s}^{t} e^{b\tau-g\tau+gs}d\tau d\varepsilon(s). \end{aligned} \tag{B.4}$$

Equation B.3 can be simplified as

$$\begin{aligned} N_S(t) &= \sigma e^{-bt} \int_{t_{i-1}}^{t} \left(\int_s^t (b-g-a) e^{b\tau - g\tau + gs} d\tau + e^{bs} \right) d\varepsilon(s) \\ &= \sigma e^{-bt} \int_{t_{i-1}}^{t} \left(\frac{(b-g-a)e^{gs}}{b-g} (e^{t(b-g)} - e^{s(b-g)}) + e^{bs} \right) d\varepsilon(s) \\ &= \sigma \int_{t_{i-1}}^{t} \left(\frac{b-g-a}{b-g} e^{g(s-t)} + \frac{a}{b-g} e^{b(s-t))} \right) d\varepsilon(s). \end{aligned} \tag{B.5}$$

For noise appearing only in the $V(t)$ term, we find from equation 2.23 that the noise of the process $S(t)$ is

$$\bar{N}_S(t) = \sigma \int_{t_{i-1}}^{t} e^{(g+a)(s-t)} d\varepsilon(s). \tag{B.6}$$

Defining $\alpha = \frac{a}{b-g}$ and assuming $\tau_1 < \tau_2$, we can calculate the covariance:

$$\begin{aligned} &\mathrm{Cov}(N_S(\tau_1), N_S(\tau_2)) \\ &= \sigma^2 \int_{t_{i-1}}^{\tau_1} ((1-\alpha)e^{g(s-\tau_1)} + \alpha e^{b(s-\tau_1)})((1-\alpha)e^{g(s-\tau_2)} + \alpha e^{b(s-\tau_2)}) ds \\ &= \frac{\sigma^2}{2} \Bigg(- \frac{e^{g(2t_{i-1}-\tau_1-\tau_2)}(-1+\alpha)^2}{g} + \frac{e^{g(\tau_1-\tau_2)}(-1+\alpha)^2}{g} \\ &\quad + \frac{2e^{(b+g)t_{i-1}}(e^{-g\tau_1 - b\tau_2} + e^{-b\tau_1 - g\tau_2})(-1+\alpha)\alpha}{b+g} \\ &\quad - \frac{2e^{-(b+g)\tau_2}(e^{g\tau_1 + b\tau_2} + e^{b\tau_1 + g\tau_2})(-1+\alpha)\alpha}{b+g} \\ &\quad - \frac{e^{b(2t_{i-1}-\tau_1-\tau_2)}\alpha^2}{b} + \frac{e^{b(\tau_1-\tau_2)}\alpha^2}{b} \Bigg). \end{aligned} \tag{B.7}$$

Then the variance of $N_S(t)$ is computed as

$$\begin{aligned} &\frac{\sigma^2}{2} \left(\frac{(-1+\alpha)^2}{g} - \frac{e^{2g(-t+t_{-1+i})}(-1+\alpha)^2}{g} - \frac{4(-1+\alpha)\alpha}{b+g} \right) \\ &\quad + \frac{\sigma^2}{2} \left(\frac{4e^{(b+g)(-t+t_{-1+i})}(-1+\alpha)\alpha}{b+g} + \frac{\alpha^2}{b} - \frac{e^{2b(-t+t_{-1+i})}\alpha^2}{b} \right). \end{aligned} \tag{B.8}$$

If t is large compared to $1/g, 1/(b+g), 1/b$, the variance of $N_S(t)$ saturates at

$$\frac{\sigma^2}{2}\left(\frac{(-1+\alpha)^2}{g} - \frac{4(-1+\alpha)\alpha}{b+g} + \frac{\alpha^2}{b}\right). \tag{B.9}$$

Appendix C: Computation of the Probability Current with Singularity Removed

Taylor expansion of equation 2.24 around t_{i-1} up to second order yields

$$\mu(t \mid x, t_{i-1}) = x + (-(g+a)(x - S_{rest}) + I(t_{i-1}))\,\Delta t + O(\Delta t^2), \tag{C.1}$$

where $\Delta t = t - t_{i-1}$ and, from the Taylor expansion of equation 2.25 up to second order:

$$\Sigma^2(t \mid t_{i-1}) = \sigma^2 \Delta t + O(\Delta t^2). \tag{C.2}$$

The transient probability G in equation 2.26 has a singularity in the limit $t \to t_{i-1}$; it is 0 if $y \neq x$ and ∞ if $y = x$, which is a Dirac function:

$$\lim_{t \to t_{i-1}} G(y, t \mid x, t_{i-1}) = \delta(x - y). \tag{C.3}$$

The FPTPD $p(t)$ can be computed from a Volterra integral equation of the second kind,

$$p(t) = -2\psi(0, t \mid S_0, 0) + 2\int_{i-1}^{t} \psi(0, t \mid 0, s)\, p(s) ds, \tag{C.4}$$

where $\psi(y, t \mid x, t_{i-1}) = \frac{\partial}{\partial t}\int_{-\infty}^{y} G(y', t \mid x, t_{i-1}) dy'$, is the time derivative of the cumulative distribution of the random variable $S(t)$, and $S_0 = S(0)$ is the initial condition of $S(t)$. The time derivative of $\mu(t \mid x, t_{i-1})$ is

$$\frac{\partial \mu(t \mid x, t_{i-1})}{\partial t} = -(g+a)\mu(t \mid x, t_{i-1}) + (g+a)S_{rest} + I(t), \tag{C.5}$$

and the time derivative of $\Sigma^2(t \mid t_{i-1})$ is

$$\frac{\partial}{\partial t}\Sigma^2(t \mid t_{i-1}) = \sigma^2 - 2(g+a)\Sigma^2(t \mid t_{i-1}). \tag{C.6}$$

Defining $z = -\frac{y-\mu(t|x,t_{i-1})}{\sqrt{2\Sigma^2(t|t_{i-1})}}$, we then obtain ψ as

$$\psi(y, t \mid x, t_{i-1}) = G(y, t \mid x, t_{i-1})\Big((g+a)(y - S_{rest}) - I(t) - \frac{\Sigma^2}{2\Sigma(t \mid t_{i-1})^2}(y - \mu(t \mid x, t_{i-1}))\Big). \tag{C.7}$$

In equation 2.27, only ψ at $y = 0$ is used, so equation C.7 can be simplified as

$$\psi(0, t \mid x, t_{i-1}) = G(0, t \mid x, t_{i-1})\Big(-(g+a)S_{rest} - I(t) + \frac{\sigma^2}{2\Sigma(t \mid t_{i-1})^2}\mu(t \mid x, t_{i-1})\Big). \tag{C.8}$$

Since $G(0, t \mid x, t_{i-1}) = \delta(x - 0)$ when $t \to t_{i-1}$, G diverges at $x = 0$. To remove the singularity, we use the following transformation, as proposed by Buonocore et al. (1987):

$$\varphi(0, t \mid x, t_{i-1}) = \psi(0, t \mid x, t_{i-1}) + \lambda(t)G(0, t \mid x, t_{i-1}). \tag{C.9}$$

When $x = 0$ and $t \to t_{i-1}$, $\varphi(0, t \mid x, t_{i-1})$ is set to be 0. $\lambda(t)$ is obtained as

$$\lambda(t) = \frac{1}{2}\left((g+a)S_{rest} + I(t)\right). \tag{C.10}$$

We thus find the probability current with singularity removed as

$$\varphi(0, t \mid x, t_{i-1}) = \frac{1}{2}G(0, t \mid x, t_{i-1})\Big(-(g+a)S_{rest} - I(t) + \frac{\sigma^2}{\Sigma(t \mid t_{i-1})^2}\mu(t \mid x, t_{i-1})\Big). \tag{C.11}$$

Acknowledgments

This work was supported by NIH grants NEI R01EY016281 and NINDS 5R01NS40596 and the Office of Naval Research, grant N000141010278.

References

Brette, R., & Gerstner, W. (2005). Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. *J. Neurophysiol.*, *94*, 3637–3642.

Brown, E., Barbieri, R., Eden, U., & Frank, L. (2003). Likelihood methods for neural spike train data analysis. In J. Feng (Ed.), *Computational neuroscience: A comprehensive approach* (pp. 252–286). London: Chapman & Hall/CRC.

Buonocore, A., Nobile, A. G., & Ricciardi, L. M. (1987). A new integral equation for the evaluation of first-passage-time probability densities. *Advances in Applied Probability*, *19*(4), 784–800.

Burke, J., Lewis, A. S., & Overton, M. L. (2008). The speed of Shor's R-algorithm. *IMA Journal of Numerical Analysis*, *28*, 711–720.

Burkitt, A. N. (2006). A review of the integrate-and-fire neuron model: I. Homogeneous synaptic input. *Biol. Cybern.*, *95*(1), 1–19.

Dong, Y., Mihalas, S., & Niebur, E. (2011). Improved integral equation solution for the first passage time of leaky integrate-and-fire neurons. *Neural Computation*, *23*(2), 421–434.

Efstratiadis, A., & Koutsoyiannis, D. (2002). An evolutionary annealing-simplex algorithm for global optimisation of water resource systems. In *Proceedings of the Fifth International Conference on Hydroinformatis* (pp. 1423–1428). Cardiff, UK: International Water Association.

Figueiredo, J. (1997). Unified finite-volume finite-differencing exponential-type scheme for convective-diffusive fluid transport equations. *Rev. Brasil Cienc. Mec./J. Braz. Soc. Mesh. Science*, *19*, 371–391.

Gerstner, W., & Naud, R. (2009). Neuroscience: How good are neuron models? *Science*, *326*(5951), 379–380.

Hille, B. (1992). *Ionic channels of excitable membranes*. Sunderland, MA: Sinauer.

Izhikevich, E. (2003). Simple model of spiking neurons. *IEEE Transactions on Neural Networks*, *14*(6), 1569–1572.

Jolivet, R., Kobayashi, R., Rauch, A., Naud, R., Shinomoto, S., & Gerstner, W. (2008). A benchmark test for a quantitative assessment of simple neuron models. *J. Neurosci. Methods*, *169*(2), 417–424.

Kim, S. S., Sripati, A., Vogelstein, R., Armiger, R., Russell, A., & Bensmaia, S. (2009). Conveying tactile feedback in sensorized hand neuroprostheses using a biofidelic model of mechanotransduction. *IEEE Transactions on Biomedical Circuits and Systems*, *3*(6), 398 –404.

Kistler, W. M., Gerstner, W., & van Hemmen, J. L. (1997). Reduction of the Hodgkin-Huxley equations to a single-variable threshold model. *Neural Computation*, *9*(5), 1015–1045.

Lapicque, L. (1907). Recherches quantitatives sur l'excitation électrique des nerfs traitée comme une polarization. *J. Physiol. Pathol. Gen.*, *9*, 620–635.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, *5*, 115–133.

Mihalas, S., Dong, Y., von der Heydt, R., & Niebur, E. (2011). Event-related simulation of neural processing in complex visual scenes. In *45th Annual Conference on Information Sciences and Systems*. Piscataway, NJ: IEEE.

Mihalas, S., & Niebur, E. (2009). A generalized linear integrate-and-fire neural model produces diverse spiking behaviors. *Neural Comput.*, *21*(3), 704–718.

Mikula, S., & Niebur, E. (2003). The effects of input rate and synchrony on a coincidence detector: Analytical solution. *Neural Computation*, *15*, 539–547.

Mikula, S., & Niebur, E. (2004). Correlated inhibitory and excitatory inputs to the coincidence detector: Analytical solution. *IEEE Transactions on Neural Networks*, *15*(5), 957–962.

Mikula, S., & Niebur, E. (2008). Exact solutions for rate and synchrony in recurrent networks of coincidence detectors. *Neural Computation*, *20*(11), 2637–2661.

Odman, M. T. (1997). A quantitative analysis of numerical diffusion introduced by advection algorithms in air quality models. *Atmospheric Environment*, *31*(13), 1933–1940.

Paninski, L. (2004). Maximum likelihood estimation of cascade point-process neural encoding models. *Network*, *15*(4), 243–262.

Paninski, L., Haith, A., & Szirtes, G. (2008). Integral equation methods for computing likelihoods and their derivatives in the stochastic integrate-and-fire model. *J. Comput. Neurosci.*, *24*(1), 69–79.

Paninski, L., Pillow, J. W., & Simoncelli, E. P. (2004). Maximum likelihood estimation of a stochastic integrate-and-fire neural encoding model. *Neural Comput.*, *16*(12), 2533–2561.

Plesser, H., & Tanaka, S. (1997). Stochastic resonance in a model neuron with reset. *Physics Letters A*, *225*, 228–234.

Polsky, A., Mel, B. W., & Schiller, J. (2004). Computational subunits in thin dendrites of pyramidal cells. *Nat. Neurosci.*, *7*, 621–627.

Prinz, A. (2007). Neuronal parameter optimization. *Scholarpedia*, *2*(1), 1903.

Rossant, C., Goodman, D.F.M., Platkiewicz, J., & Brette, R. (2010). Automatic fitting of spiking neuron models to electrophysiological recordings. *Front. Neuroinformatics*, *4*, 2.

Russell, A., Orchard, G., Dong, Y., Mihalas, S., Niebur, E., Tapson, J., et al. (2010). Optimization methods for spiking neurons and networks. *IEEE Transactions on Neural Networks*, *21*(12), 1950–1962.

Schrödinger, E. (1915). Zur Theorie der Fall- und Steigversuche an Teilchen mit Brownscher Bewegung. *Physikalische Zeitschrift*, *16*, 289–295.

Shor, N. Z., Kiwiel, K. C., & Ruszcayǹski, A. (1985). *Minimization methods for non-differentiable functions*. New York: Springer-Verlag.

Stuart, G., Schiller, J., & Sakmann, B. (1997). Action potential initiation and propagation in rat neocortical pyramidal neurons. *J. Physiol. (Lond.)*, *505 (Pt 3)*, 617–632.

Tonnelier, A., Belmabrouk, H., & Martinez, D. (2007). Event-driven simulations of nonlinear integrate-and-fire neurons. *Neural Comput.*, *19*(12), 3226–3238.

van Rossum, M. C. (2001). A novel spike distance. *Neural Comput.*, *13*(4), 751–763.

Victor, J. D., & Purpura, K. P. (1996). Nature and precision of temporal coding in visual cortex: A metric-space analysis. *J. Neurophysiol.*, *76*(2), 1310–1326.

Received September 7, 2010; accepted April 26, 2011.