

# On the Capabilities and Computational Costs of Neuron Models

Michael J. Skocik and Lyle N. Long

**Abstract**—We review the Hodgkin–Huxley, Izhikevich, and leaky integrate-and-fire neuron models in regular spiking modes solved with the forward Euler, fourth-order Runge–Kutta, and exponential Euler methods and determine the necessary time steps and corresponding computational costs required to make the solutions accurate. We conclude that the leaky integrate-and-fire needs the least number of computations, and that the Hodgkin–Huxley and Izhikevich models are comparable in computational cost.

**Index Terms**—Accuracy, computational costs, Hodgkin–Huxley, Izhikevich, leaky integrate-and-fire.

## I. INTRODUCTION

**S**PIKING neural networks have become widely used in a variety of biological and engineering applications, and to date there have been a large number of neuron models that have been proposed in the scientific literature. These artificial networks are applied in such diverse fields as logic gating [1], pattern recognition [2], supervised and unsupervised learning [3], [4], and hybridized cognitive architecture creation [5]. This paper seeks to better understand the neuron models that are used in such endeavors, and to propose recommendations for which models should be used in certain circumstances.

Additionally, in the past decade there has been a lot of research conducted on understanding global neuron phenomena, such as how synchronous states arise from a set of interacting neurons [6]–[8], and how these states have emergent properties that can be used to perform computations [9]. Such studies are, however, often limited by the amount of computational power available; it is therefore of paramount importance that the computational costs of the different neuron models used to construct these networks be well understood.

Furthermore, many neural phenomena are dependent on the temporal order of neuron spiking [10], [11], including the widely used Hebbian learning algorithm of spike-time-dependent plasticity (STDP) [12]–[15] and the phenomenon of synchronous states. Even more engineering-oriented projects, for example, pattern recognition, are often dependent on spike timing [2]. For situations such as these, it is important to simulate the behavior of individual neurons exactly. To these ends, this paper not only examines the cost of implementing

different solution methods on different neuron models, but also studies how different time resolutions affect a neuron model's accuracy.

## II. OVERVIEW OF NEURON MODEL

This paper reviews three different neuron models: the biologically realistic Hodgkin–Huxley model, the widely used Izhikevich model, and the beautifully simple leaky integrate-and-fire model.

### A. Hodgkin–Huxley Model

The Hodgkin–Huxley model, which is one of the most biophysically meaningful models of neuron spiking, was created in [16]. Designed from the study of a squid giant axon, this model was proposed to describe the response of a neuron to external current stimulation, and included the effects of different ionic channels and leakage currents. The Hodgkin–Huxley model involves a system of nonlinear ordinary differential equations, and is given by

$$C_m \frac{du}{dt} = -g_{Na}(u - V_{Na})m^3h - g_K(u - V_K)n^4 - g_L(u - V_L) + I \quad (1)$$

where  $u$  is the membrane potential;  $C_m$  is the effective capacitance per area of the membrane;  $t$  is time;  $g_{Na}$ ,  $g_K$ , and  $g_L$  are the conductances of the sodium, potassium, and leak channels, respectively;  $V_{Na}$ ,  $V_K$ , and  $V_L$  are the reversal potentials of the sodium, potassium, and leak channels, respectively;  $I$  is the stimulation current per area;  $m$ ,  $n$ , and  $h$  are the coefficients determined by

$$\begin{aligned} \frac{dm}{dt} &= \alpha_m(1 - m) - \beta_m m \\ \frac{dn}{dt} &= \alpha_n(1 - n) - \beta_n n \\ \frac{dh}{dt} &= \alpha_h(1 - h) - \beta_h h \end{aligned} \quad (2)$$

where the  $\alpha$  and  $\beta$  coefficients are given by

$$\begin{aligned} \alpha_m &= \frac{2.5 - 0.1u}{\text{Exp}[2.5 - 0.1u] - 1} & \beta_m &= 4\text{Exp}\left[-\frac{u}{18}\right] \\ \alpha_n &= \frac{0.1 - 0.01u}{\text{Exp}[1 - 0.1u] - 1} & \beta_n &= 0.125\text{Exp}\left[-\frac{u}{80}\right] \\ \alpha_h &= 0.07\text{Exp}\left[-\frac{u}{20}\right] & \beta_h &= \frac{1}{\text{Exp}[3 - 0.1u] + 1} \end{aligned} \quad (3)$$

Manuscript received June 7, 2013; revised November 22, 2013; accepted December 1, 2013. Date of publication December 18, 2013; date of current version July 14, 2014.

M. J. Skocik is with the Department of Physics, The Pennsylvania State University, University Park, PA 16802 USA (e-mail: mike.skocik@live.com).

L. N. Long is with the Department of Aerospace Engineering, The Pennsylvania State University, University Park, PA 16802 USA (e-mail: lnl@psu.edu).

Digital Object Identifier 10.1109/TNNLS.2013.2294016

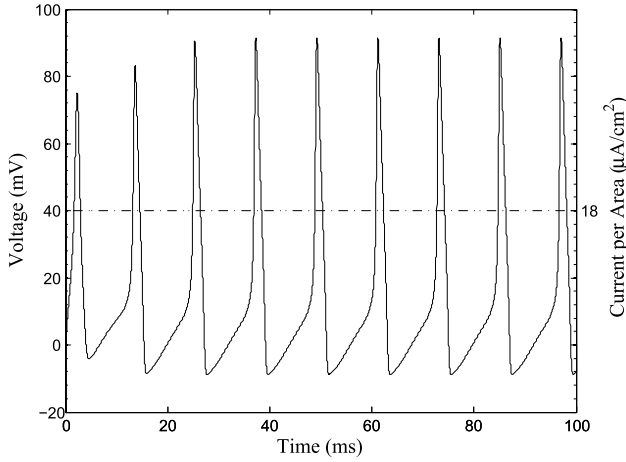


Fig. 1. Membrane potential (solid line) of a Hodgkin-Huxley neuron under a sustained current per area (dashed line) of  $18 \mu\text{A}/\text{cm}^2$ .

where  $\text{Exp}[\gamma]$  signifies  $e^\gamma$ , and the constants are often given by

$$\begin{aligned} V_{\text{Na}} &= 115 \text{ mV} & g_{\text{Na}} &= 120 \frac{\text{ms}}{\text{cm}^2} \\ V_K &= -12 \text{ mV} & g_K &= 36 \frac{\text{ms}}{\text{cm}^2} \\ V_L &= 10.6 \text{ mV} & g_L &= 0.3 \frac{\text{ms}}{\text{cm}^2} \\ C_m &= 1 \frac{\mu\text{F}}{\text{cm}^2}. \end{aligned} \quad (4)$$

From these equations, it is easy to obtain the impression that the Hodgkin-Huxley model is a very computationally expensive system. Verily, this model has been described as analytically impossible [17], inherently complex [18], and only viable for small neural networks [19]. However, those claims are disputed below; if coded correctly, we assert that this model, which is the most biologically plausible model considered [20], can be implemented in a manner that is more computationally efficient than most competing models. This topic will be returned to shortly.

The membrane potential of the Hodgkin-Huxley model under sustained current is shown in Fig. 1. The frequency at which this neuron fires is dependent on the stimulation current; this frequency response is shown in Fig. 2. This response curve was created by increasing the stimulation current of a Hodgkin-Huxley neuron in increments of  $0.3 \mu\text{A}/\text{cm}^2$  at 1000-ms intervals.

Other papers have shown that the Hodgkin-Huxley model can exhibit all neuron-spiking behaviors through parameter modification [20]. Also, adaptations to the model have been published that produce different neuron behaviors [21]. Furthermore, because this model is so biophysically meaningful, other models, such as the Morris-Lecar model, have been published as simplified versions of the Hodgkin-Huxley model [20].

One alternative to the biologically realistic Hodgkin-Huxley model is the Izhikevich model, introduced for its computational efficiency.

### B. Izhikevich Model

Through the study of dynamical systems and bifurcation theory, Izhikevich [20], [22] developed a model of neuron

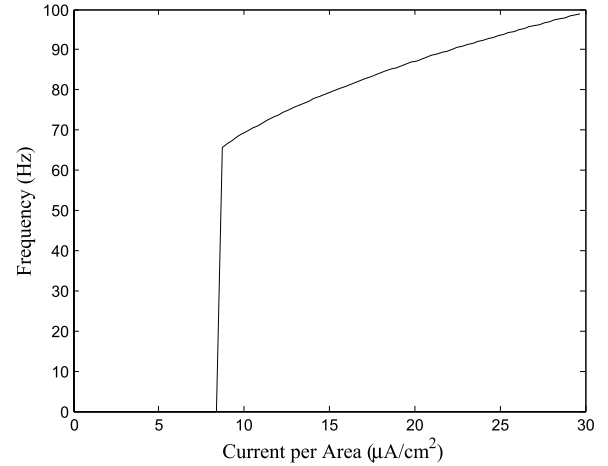


Fig. 2. Frequency response of a Hodgkin-Huxley neuron to sustained current. Each current was applied for an interval of 1000 ms, with current per area increasing in increments of  $0.3 \mu\text{A}/\text{cm}^2$ .

spiking that involved less than half the equations of the Hodgkin-Huxley model, yet seemed to replicate most of the neuron-spiking behaviors. The tradeoff, then, was that his model was not biologically realistic [20], but rather phenomenological. The equations of the Izhikevich model are

$$\begin{aligned} \frac{dv}{dt} &= 0.04v^2 + 5v + 140 - u + I \\ \frac{du}{dt} &= a(bv - u) \\ \text{if } v &\geq 30 \text{ mV, then } \begin{matrix} v \rightarrow c \\ u \rightarrow u+d \end{matrix} \end{aligned} \quad (5)$$

where  $v$  is the membrane potential,  $u$  is a recovery variable used in adjusting the membrane potential,  $t$  is time,  $I$  is the stimulation current, and  $a$ ,  $b$ ,  $c$ , and  $d$  are the adjustable parameters. Once the membrane potential reaches 30 mV, a spike is emitted and the membrane voltage is reset to a certain value.

The adjustable parameters in this model can take on a wide array of values, resulting in different spiking behaviors. This is not unique to the Izhikevich model; as was previously mentioned, the Hodgkin-Huxley model can also exhibit other spiking behaviors if its parameters are modified. The scope of this paper is limited to regular spiking modes; the forthcoming results are not necessarily applicable to all parameter subsets for the investigated models, including subsets that cause bursting behaviors. Therefore, to induce regular spiking, the parameters were set to

$$a = 0.02 \text{ Hz} \quad b = 0.2 \quad c = -65 \text{ mV} \quad d = 2. \quad (6)$$

The firing of this model and its frequency response to constant current are shown in Figs. 3 and 4, respectively.

This model has become one of the most widely used models of neuron spiking, and has been hailed by its author as the fastest model that exhibits all biological features [20]. However, these claims do not consider accuracy; as will be shown shortly, when achieving a level of accuracy comparable with other models, this model requires a comparable number of computations. Furthermore, it has been demonstrated that

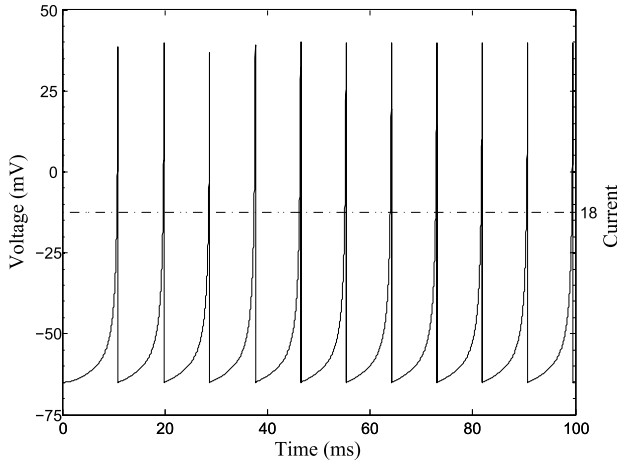


Fig. 3. Membrane potential (solid line) of an Izhikevich neuron under a sustained current per area (dashed line) of 18.

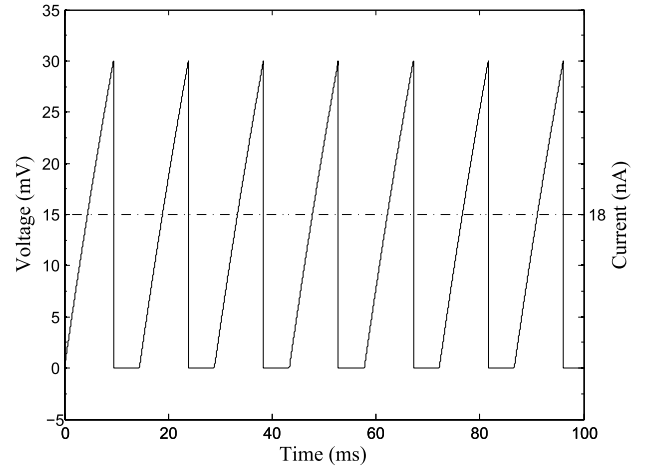


Fig. 5. Membrane potential (solid line) of a leaky integrate-and-fire neuron under a sustained current per area (dashed line) of 18 nA.

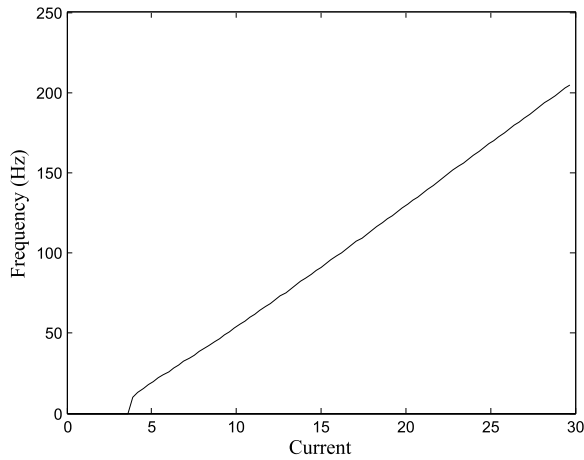


Fig. 4. Frequency response of an Izhikevich neuron to sustained current. Each current was applied for an interval of 1000 ms, with current per area increasing in increments of 0.3.

the behavior of the Izhikevich model changes as the time step is varied [23], [24]. Accuracy is important because, if the equations are not solved accurately, then one is not really solving the original equations at all but rather some different equations. The solutions then depend on the size of the time step (i.e., the solution is nonconvergent), and may be simply fortuitous. For partial differential equations, this is a well known and much studied issue [25]; membrane potential solutions will not simply be off from the exact solution by a constant amount, but rather the complexity of the network could lead to entirely different dynamics. This is caused by the finite difference numerical methods, which are based upon Taylor series expansions, and the higher order terms are neglected. Therefore, they are reasonable approximations to the original equations only if the time step size is very small. Therefore, the appearance of certain phenomenon, such as synchronization, may be called into question entirely, and the network may not display the desired behaviors.

### C. Leaky Integrate-and-Fire Model

The leaky integrate-and-fire model is one of the simplest models of neuron spiking [26]. It involves only one differential equation, and it models a neuron membrane as a capacitor of capacitance  $C$  in parallel with a resistor of resistance  $R$ . The dynamics are then given by

$$\begin{aligned} \tau_m \frac{du}{dt} &= -u + RI \\ \text{if } u &\geq u_{th}, \text{ then } u \rightarrow u_{rst} \end{aligned} \quad (7)$$

where  $\tau_m = RC$  is the circuit's time constant,  $u$  is the membrane potential,  $t$  is time, and  $I$  is the stimulation current. Once the firing threshold  $u_{th}$  is crossed, a spike is emitted and the membrane potential is reset to its rest value  $u_{rst}$ . Additionally, it is customary to program in an absolute refractory period of length  $t_r$  after each firing.

In this paper, the parameters were set to

$$\begin{aligned} R &= 8.22 \text{ M}\Omega & C &= 5.0675 \text{ nF} \\ u_{th} &= 30 \text{ mV} & u_{rst} &= 0 \text{ mV} \\ t_r &= 5 \text{ ms.} \end{aligned} \quad (8)$$

While this model is too simple to exhibit any interesting spiking behavior, it will be taken as an example of the simplest a neuron model can be. The firing of this model and its frequency response to constant current are shown in Figs. 5 and 6, respectively.

Now that the three neuron models have been introduced, the following section discusses the solution methods and their accuracies.

## III. SOLUTION METHODS AND THEIR ACCURACIES

It is not possible to analytically solve the differential equations of any of these models except for in the most trivial of cases. This section discusses applicable approximate solution methods and the resulting accuracies.

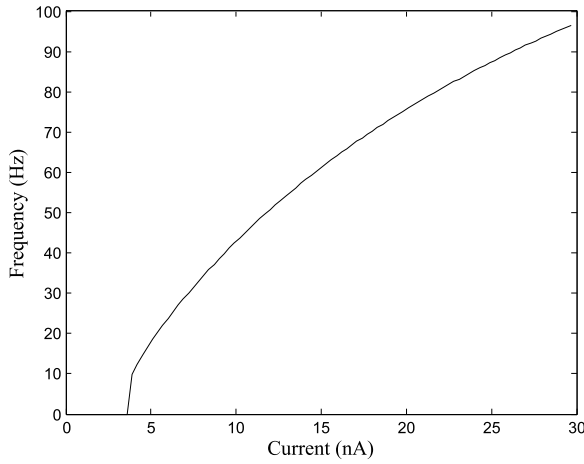


Fig. 6. Frequency response of a leaky integrate-and-fire neuron to sustained current. Each current was applied for an interval of 1000 ms, with current per area increasing in increments of 0.3 nA.

### A. Solution Methods

There are many different methods that can be used to approximate solutions to these models. Three of them will be discussed here: the forward Euler method, the fourth-order Runge–Kutta method, and the exponential Euler method. In this section, the following notation will be used: the value of any quantity  $\gamma$  at time step  $n$  will be denoted as  $\gamma^n$ .

1) *Forward Euler Method*: One of the simplest methods for approximating solutions to the above equations is the forward Euler method [27]. If one has an equation of the form

$$\frac{d\zeta}{dt} = \Psi(\zeta, t) \quad (9)$$

where  $\Psi(\zeta, t)$  is some arbitrary function of  $\zeta$  and  $t$ , and if one knows the value of  $\zeta$  at some time  $t$  corresponding to time step  $n$  given in increments of  $\Delta t$ , then one can approximate  $\zeta$  at time  $t + \Delta t$  by

$$\zeta^{n+1} = \zeta^n + \Psi(\zeta^n, t)\Delta t. \quad (10)$$

This method is very simple, but it is only first-order accurate, meaning its error is proportional to the time step size  $\Delta t$ . The backward Euler method (an implicit method) can be more stable, but is also only first-order accurate. A better solution can be given by a fourth-order Runge–Kutta method, which is fourth-order accurate [error  $\propto (\Delta t)^4$ ].

2) *Fourth-Order Runge–Kutta Method*: One fourth-order Runge–Kutta method [28] determines the value of  $\zeta^{n+1}$  via

$$\zeta^{n+1} = \zeta^n + \frac{\Delta t}{6}(\zeta_1 + 2\zeta_2 + 2\zeta_3 + \zeta_4) \quad (11)$$

where

$$\begin{aligned} \zeta_1 &= \Psi(\zeta^n, t) \\ \zeta_2 &= \Psi\left(\zeta^n + \frac{1}{2}\zeta_1\Delta t, t + \frac{1}{2}\Delta t\right) \\ \zeta_3 &= \Psi\left(\zeta^n + \frac{1}{2}\zeta_2\Delta t, t + \frac{1}{2}\Delta t\right) \\ \zeta_4 &= \Psi(\zeta^n + \zeta_3\Delta t, t + \Delta t). \end{aligned} \quad (12)$$

This method is easily generalizable to coupled, multivariable systems. From these equations, it is immediately obvious that this method is much more computationally intensive than the forward Euler method, but it is also much more accurate and stable. A third method, which is more middle-of-the-road in terms of computational complexity, is the exponential Euler method.

3) *Exponential Euler Method*: The exponential Euler method is a powerful method that only works if  $\Psi(\zeta, t)$  from (9) takes the form

$$\Psi(\zeta, t) = \frac{d\zeta}{dt} = \Theta(t) - \Phi(t)\zeta \quad (13)$$

where  $\Theta(t)$  and  $\Phi(t)$  are the arbitrary functions of  $t$ . The approximate solution can then be given by

$$\zeta^{n+1} = \left(\zeta^n - \frac{\Theta^n}{\Phi^n}\right)e^{-\Phi^n\Delta t} + \frac{\Theta^n}{\Phi^n} \quad (14)$$

which is the exact solution if  $\Theta(t)$  and  $\Phi(t)$  are constant over the interval  $\Delta t$  [29]. Conversely, if  $\Theta(t)$  and  $\Phi(t)$  vary rapidly in time, or if they are dependent on  $\zeta$ , then this method might require iterations within each time step.

It is easy to observe how the leaky integrate-and-fire model can take advantage of this method. The other two models can, too, with a small amount of rewriting. For the Hodgkin–Huxley model, the voltage equation takes the form

$$\begin{aligned} \frac{du}{dt} &= (g_L V_L + g_{Na} V_{Na} m^3 h + g_K V_K n^4 + I) \\ &\quad - (g_L + g_{Na} m^3 h + g_K n^4)u \end{aligned} \quad (15)$$

where  $C_m$  has been assumed to have a value of unity, and the coefficient equations become

$$\begin{aligned} \frac{dm}{dt} &= \alpha_m - (\alpha_m + \beta_m)m \\ \frac{dn}{dt} &= \alpha_n - (\alpha_n + \beta_n)n \\ \frac{dh}{dt} &= \alpha_h - (\alpha_h + \beta_h)h. \end{aligned} \quad (16)$$

Similarly, the Izhikevich model is rewritten as

$$\begin{aligned} \frac{dv}{dt} &= (140 - u + I) - (-0.04v - 5)v \\ \frac{du}{dt} &= abv - au. \end{aligned} \quad (17)$$

Because the coefficients of (15) and (17) depend on  $v$  and  $u$ , iteration must be considered when implementing this method on these models. In our tests, iteration was required for the Izhikevich but not for the Hodgkin–Huxley model, making the exponential Euler method competitive only for the latter. We believe this to be due to the strong nonlinearity and sharp discontinuities of the Izhikevich model. The exponential Euler method actually proved itself to be the most stable method that was tested, even in the presence of noise.

In using all three of the aforementioned solution methods, it is important to realize that there exists a key difference between the Hodgkin–Huxley and the other two models: the former is a continuous function while the latter have jump discontinuities. Jump discontinuities are quite rare in nature;

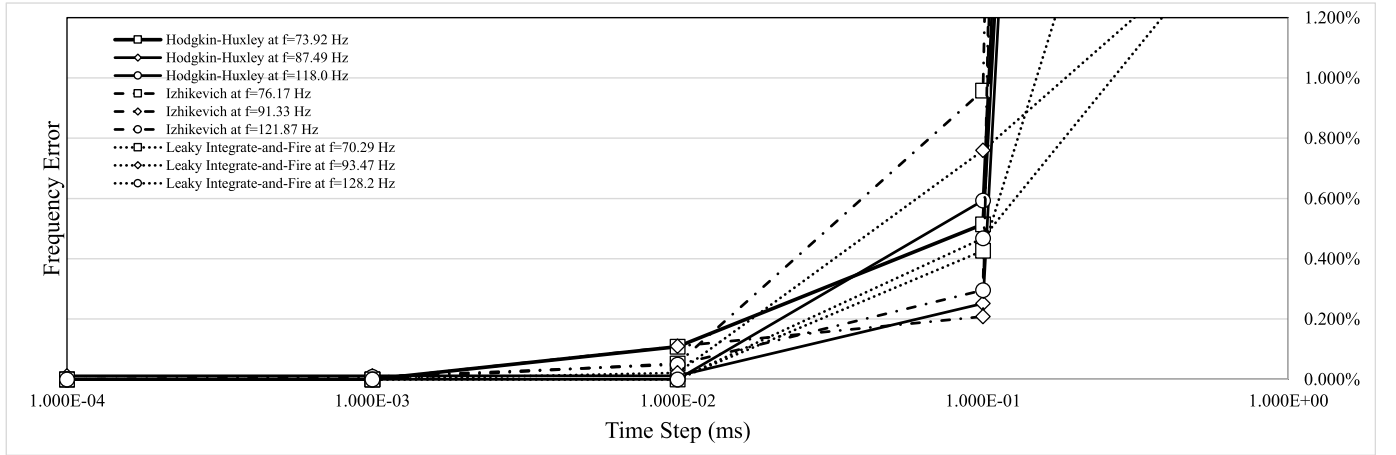


Fig. 7. Frequency errors for varying time steps for the forward Euler method on the leaky integrate-and-fire model (dotted lines), the fourth-order Runge–Kutta method on the Izhikevich model (dash dotted lines) and the exponential Euler method on the Hodgkin–Huxley model (solid lines). These tests were conducted at roughly 70, 90, and 120 Hz.

TABLE I  
FREQUENCY ERRORS FOR LOW-FREQUENCY RUNS  
(ROUGHLY 70 Hz), IN PERCENTAGES

Time Step Size:		$10^{-4}$	$10^{-3}$	$10^{-2}$	$10^{-1}$	$10^0$
Hodg.-Hux.	FE	0.11	0.11	0.51		
	RK4	BM	0.00	0.20	2.65	
	EE	0.00	0.00	0.11	0.51	19.01
Izhikevich	FE	0.00	0.01	0.22	1.64	13.12
	RK4	BM	0.00	0.05	0.96	46.92
	EE	0.00	0.03	0.25	2.30	339.0
Leaky I&F	FE	0.00	0.00	0.00	0.43	3.70
	RK4	BM	0.00	0.00	0.43	3.70
	EE	0.00	0.00	0.00	0.43	3.70

TABLE II  
FREQUENCY ERRORS FOR MEDIUM-FREQUENCY RUNS  
(ROUGHLY 90 Hz), IN PERCENTAGES

Time Step Size:		$10^{-4}$	$10^{-3}$	$10^{-2}$	$10^{-1}$	$10^0$
Hodg.-Hux.	FE	0.10	0.10	0.50		
	RK4	BM	0.00	0.15	2.52	
	EE	0.01	0.01	0.01	0.25	19.03
Izhikevich	FE	0.00	0.01	0.20	1.93	8.09
	RK4	BM	0.00	0.11	0.21	50.62
	EE	0.01	0.03	0.16	2.54	10.61
Leaky I&F	FE	0.36	0.02	0.02	0.76	1.65
	RK4	BM	0.00	0.02	0.76	1.65
	EE	0.00	0.00	0.02	0.76	1.65

even shockwaves are continuous when viewed on a fine enough scale. It is quite difficult to model jump discontinuities using finite difference methods, which have their origin in Taylor series expansions. The Hodgkin–Huxley equations are quite smooth, and this is very beneficial for solving them numerically.

The accuracy of these methods, which is dependent on the time step  $\Delta t$  used, is discussed in the following section.

### B. Accuracies of Different Solution Methods

To examine the accuracy of the above methods with varying input currents and time steps, the three aforementioned neuron models were constructed, tested, and compared. There were two different kinds of tests run: one that compared the firing frequency over an extended time period, and another that compared the voltage profile for a single action potential. For the former, the neuron models were run with three different input currents and with time steps ranging from  $10^{-4}$  to 1 ms. More details on this process can be found in the appendix. For this paper, it was our intention to show the relative computational abilities of the three neuron models; for this

reason, the time steps were varied by orders of magnitude. We are confident that the models can be further optimized for specific tasks; it was not our intention to carry out this optimization, but rather to point to the key concepts involved.

After running the firing frequency test, from the resulting firing pattern the frequency error was determined by

$$\text{Frequency Error (\%)} = \frac{|f - f_0|}{f_0} \times 100 \quad (18)$$

where  $f$  is the firing frequency of the case being tested and  $f_0$  is the firing frequency of the benchmark solution, which was taken to be the solution produced by the fourth-order Runge–Kutta method with a time step of  $\Delta t = 10^{-4}$  ms. The benchmark solutions were different for each model, i.e., the frequency produced by the fourth-order Runge–Kutta method with the Hodgkin–Huxley model is different from the frequency produced by the fourth-order Runge–Kutta method for the Izhikevich model.

A select number of these frequency errors are shown in Fig. 7; as will be discussed in Section IV, the results shown in this figure are the errors for the methods that allow for the most efficient implementation of each model. The complete results are shown in Tables I–III. In this section, a blank cell means

that the solution method was unstable for the given time step and BM means that the given time step and method was used as the benchmark solution. Additionally, the following abbreviations will be used for the rest of the paper: Hodg.-Hux. is the Hodgkin–Huxley model, leaky IF is the leaky integrate-and-fire model, FE is the forward Euler method, RK4 is the fourth-order Runge–Kutta method, and EE is the exponential Euler method.

The second test, which compared the voltage profile for a single spike, took neuron models at rest and triggered a single action potential by inputting a short-current pulse. The error was determined via the root mean square (rms) deviation

$$\text{RMS Deviation} = \sqrt{\frac{1}{n} \sum_{i=1}^n (V^i - V_0^i)^2} \quad (19)$$

where  $n$  is the number of time steps,  $V$  is the voltage of the case being tested, and  $V_0$  is the voltage of the benchmark solution, which was calculated as it was for the frequency error. A few examples of the voltage profiles being compared with the benchmark solutions are shown in Fig. 8. Complete results are shown in Table IV, and more details on how these tests were run are discussed in the appendix.

Therefore, if one is to keep the frequency error under one percent and the rms Deviation under 15 mV, one is required to use the time steps listed in Table V.

It is important to realize the limitations of these tests: they are very focused on spike timing. Spike timing is very important for such applications as STDP [11] and synchronous states [6], and may be less relevant in such cases as population encoding [30] where spike timing is not as important. An interesting future area of research would be the study of the oscillation frequency error of these models [31]. It is also worth noting that these tests, even though they were performed under deterministic conditions, are entirely applicable to (and are often used in) stochastic situations; the addition of a noise term can be added regardless of the numerical scheme used.

In discussing the accuracies of the solution methods, it is also important to consider the Nyquist frequency condition, which states that one normally needs more than two points per wavelength, including the refractory period, to avoid aliasing, e.g., for a 50-Hz signal (which has a period of 20 ms) the time step size must be smaller than 10 ms. This is an absolute upper limit; to accurately capture a complex waveform one might need 50–75 points per wavelength (with a low-order scheme). Because one does not know ahead of time what the firing rate will be, the rate must be smaller than that which is required for the highest possible frequency. Furthermore, for a complex signal such as bursting, the time step size will be determined by the firing rate in the burst, not just the time period between bursts. This requirement could be relaxed if one could use an adaptive time step scheme.

Another factor, which affects the choice of time step size, is the actual firing of the neuron. In the Izhikevich and leaky integrate-and-fire models, the firing is determined when the voltage exceeds a certain threshold. As is discussed in [24], because the voltage is typically only compared with the

TABLE III  
FREQUENCY ERRORS FOR HIGH-FREQUENCY RUNS  
(ROUGHLY 120 Hz), IN PERCENTAGES

Time Step Size:		$10^{-4}$	$10^{-3}$	$10^{-2}$	$10^{-1}$	$10^0$
Hodg.-Hux.	FE	0.00	0.10	0.66	6.86	
	RK4	BM	0.00	0.40	2.80	
	EE	0.00	0.00	0.00	0.59	22.14
Izhikevich	FE	0.00	0.02	0.16	1.48	231.5
	RK4	BM	0.01	0.05	0.30	52.78
	EE	0.00	0.02	0.19	2.15	18.17
Leaky I&F	FE	0.23	0.08	0.08	0.14	1.72
	RK4	BM	0.00	0.00	0.14	1.72
	EE	0.00	0.00	0.08	0.14	1.72

threshold at the end of each time step, if the time step size is too large, the firing will be delayed and in error.

If a spiking neuron model is being used, then it should be solved properly. If one permits inaccuracies, aliasing, lack of convergence, time-step-dependent solutions, and so on, then one is not actually solving the model equation but rather a modified equation that may not lead to the same system dynamics. If this is the case, then one should instead use some sort of stochastic model, where the randomness can be documented and the results are repeatable.

#### IV. COMPUTATIONAL COST

This section discusses the computational costs of implementing the above methods for the neuron models. In calculating the computational cost of the above methods, addition, multiplication, division, and subtraction have been assumed to cost one operation, and exponentials have been assumed to cost 10 operations (which is the same assumption Izhikevich made when comparing the neuron models [20]; however, modern compilers can often perform these operations much faster).

Here, the Hodgkin–Huxley model has been computationally simplified through the use of table lookups for coefficients containing the  $\alpha$  and  $\beta$  coefficients, e.g., for the exponential Euler method, plugging the  $m$  equation of (16) into (14), one obtains

$$m^{n+1} = (m^n - \Gamma_m(u))e^{\Upsilon_m(u)} + \Gamma_m(u) \quad (20)$$

where

$$\begin{aligned} \Gamma_m(u) &= \frac{\alpha_m(u)}{\alpha_m(u) + \beta_m(u)} \\ \Upsilon_m(u) &= -(\alpha_m(u) + \beta_m(u))\Delta t. \end{aligned} \quad (21)$$

By storing periodic values of  $\Gamma(u)$  and  $\Upsilon(u)$ , one can drastically reduce the number of computations required. This method was used when determining the accuracy of the solution methods in the previous section, where values were stored in 1-mV increments. There are no analogous coefficients in the Izhikevich or the leaky integrate-and-fire models that could take advantage of this method.

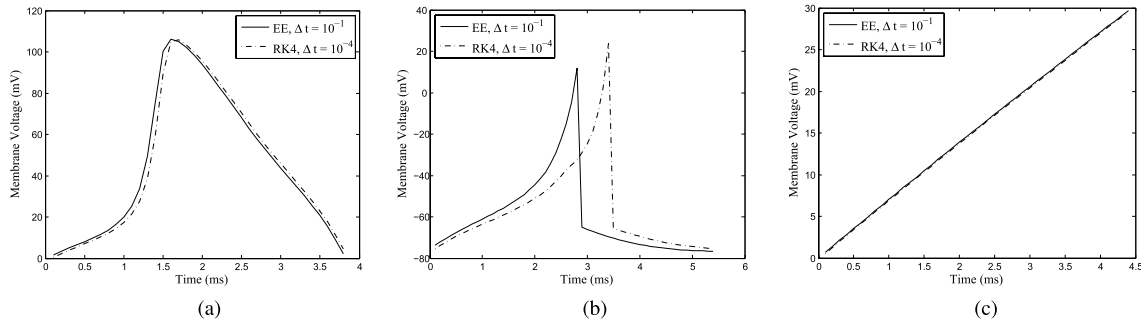


Fig. 8. Examples of the comparisons performed when determining the rms deviation. All of these examples are comparisons of the exponential Euler method ( $\Delta t = 10^{-1}$  ms) with the fourth-order Runge–Kutta method ( $\Delta t = 10^{-4}$  ms). (a) Hodgkin–Huxley neuron. (b) Izhikevich neuron. (c) Leaky integrate-and-fire neuron. The poor convergence of the Izhikevich model suggests that an iterative method (e.g., a fourth-order Runge–Kutta method) should be used instead for this time step size.

TABLE IV  
RMS DEVIATIONS FOR ALL METHODS AND MODELS

Time Step Size:		$10^{-4}$	$10^{-3}$	$10^{-2}$	$10^{-1}$	$10^0$
Hodg.-Hux.	FE	0.03	0.15	1.33		
	RK4	BM	0.11	0.94	19.71	
	EE	0.01	0.01	0.19	4.20	12.11
Izhikevich	FE	0.01	0.10	6.19	12.08	34.92
	RK4	BM	0.00	0.02	12.39	22.04
	EE	0.03	0.32	10.33	22.35	88.08
Leaky I&F	FE	0.00	0.45	1.43	0.22	5.82
	RK4	BM	0.45	1.43	0.17	5.33
	EE	0.18	0.78	1.43	1.99	5.65

TABLE V  
TIME STEPS, IN MILLISECONDS, REQUIRED FOR  
ACCURATE SIMULATION

	Hodg.-Hux.	Izhikevich	Leaky I&F
FE	0.01	0.01	0.1
RK4	0.01	0.1	0.1
EE	0.1	0.01	0.1

TABLE VI  
NUMBER OF OPERATIONS REQUIRED FOR EACH TIME STEP  
UPDATE. (ADDITION, MULTIPLICATION = 1 OPERATION.  
EXPONENTIATION = 10 OPERATIONS)

	Hodg.-Hux.	Izhikevich	Leaky I&F
FE	31	14	4
RK4	137	96	28
EE	69	33	14

Then, counting the number of operations in each time step update, one obtains the computational costs listed in the Table VI.

Using the results listed in Table V, the number of operations required to accurately simulate 1 ms of voltage updates is listed in Table VII.

From this table it is clear that, if the goal is to simulate accurately and efficiently, the Hodgkin–Huxley model can be best implemented with the exponential Euler method,

TABLE VII  
NUMBER OF OPERATIONS REQUIRED FOR 1 ms OF ACCURATE  
SIMULATION (ADDITION, MULTIPLICATION = 1 OPERATION.  
EXPONENTIATION = 10 OPERATIONS)

	Hodg.-Hux.	Izhikevich	Leaky I&F
FE	3100	1400	40
RK4	13 700	960	280
EE	690	3300	140

TABLE VIII  
CPU TIME ELAPSED PER SECOND OF SIMULATION TIME, USING  
THE TIME STEPS THAT MAKE THE MODELS ACCURATE,  
IN MICROSECONDS

	Hodg.-Hux.	Izhikevich	Leaky I&F
FE	8.64	3.01	0.12
RK4	12.86	1.05	0.84
EE	2.11	7.10	0.14

costing 690 operations per millisecond; the Izhikevich with a fourth-order Runge–Kutta method, costing 960 operations per millisecond; and the leaky integrate-and-fire model with the forward Euler method, costing 40 operations per millisecond. Therefore, the Hodgkin–Huxley model requires 28% less operations per millisecond than the Izhikevich model, while the leaky integrate-and-fire model requires 94% and 96% less computations than the Hodgkin–Huxley and Izhikevich models, respectively.

To demonstrate the validity of this analysis, the neuron models were run with the time steps recommended in Table V on the computer and with the compiler and options described in the appendix for a simulation time of  $10^6$  ms. The CPU time elapsed per millisecond of simulation time for each of these runs is listed in Table VIII.

From this Table, it is clear that while the number of operations per millisecond does not definitively give the runtime of a model, it is nonetheless generally a good indicator of how fast the model will run relative to other models. The fastest algorithm for each neuron model is the same in both Tables VII and VIII. Additionally, memory costs were calculated; these requirements are listed in Table IX.

TABLE IX

MEMORY REQUIREMENTS FOR THE NEURON MODELS. BOTH THE VARIABLES (VAR.S) NEEDED AND THE TOTAL NUMBERS (#) NEEDED ARE LISTED

	Var.s	#
Hodg.-Hux.	V, m, n, h	4
Izhikevich	v, u	2
Leaky I&F	u	1

TABLE X

INPUT CURRENTS FOR THE FREQUENCY ERROR TESTS

	Low-Freq.	Med-Freq.	High-Freq.
Hodg.-Hux.	13	20	50
Izhikevich	13	15	19
Leaky I&F	18	28	55

The results presented here show that the oft-cited trends presented in Fig. 2 of Izhikevich [20] are not accurate. This figure shows the Hodgkin–Huxley model to be prohibitive, and it clearly is not. Additionally, it shows the Izhikevich model to be nearly as computationally expedient as the leaky integrate-and-fire model; this is also incorrect. Furthermore, that figure shows the Izhikevich model to be of roughly the same level of biological plausibility as the Hodgkin–Huxley model, which is not true. The Izhikevich model is a phenomenological model of one variable (the potential), while the Hodgkin–Huxley model simulates not only the potential, but also the ionic channel activation and inactivation (m and n, and h) as well. If the methods were compared on an information content, the Hodgkin–Huxley model would be much higher. Not only is it simulating more physics, but it also has a very rich data content due to the lack of discontinuities.

## V. CONCLUSION

When coding neuron models, to have meaningful results, it is important to not only consider the models' computational costs per time step but also the time steps and the solution methods that make them accurate and convergent. While the Hodgkin–Huxley model seems more complex than the Izhikevich model, it is actually comparable in cost when implemented correctly. The leaky integrate-and-fire model requires much less operations than either of the two, but is a crude approximation of a neuron.

If one is to use the most efficient methods, one obtains the following conclusions.

- 1) The Hodgkin–Huxley model can be best implemented with the exponential Euler method, requiring roughly 690 operations per millisecond, or  $2.11 \mu\text{s}$  per millisecond of simulation time.
- 2) The Izhikevich model can be best implemented with the fourth-order Runge–Kutta method, requiring roughly 960 operations per millisecond, or  $1.05 \mu\text{s}$  per millisecond of simulation time.
- 3) The leaky integrate-and-fire model can be best implemented with the forward Euler method, requiring

roughly 40 operations per millisecond, or  $0.12 \mu\text{s}$  per millisecond of simulation time.

- 4) All of these methods can be run accurately with a time step of 0.1 ms.

It is important to remember the scope of these results. These experiments were carried out under regular spiking mode and were evaluated via tests that focused on spike frequency and spike timing. This is critically important for many applications, such as STDP, processes that encode information in neuron spikes and spike trains [32], and winner-take-all approaches [33], and is less important in areas such as population encoding [30]. Further research should be done on different spiking modes and with different error tests.

Even with these qualifiers in mind, from these results it is clear that, contrary to many claims in the literature, the biophysically meaningful Hodgkin–Huxley model is not computationally prohibitive, and should therefore be used more often in neural network simulations, including in biological simulations and engineering applications.

## APPENDIX

### TECHNICAL DETAILS

All these tests were carried out on an Asus Zenbook Laptop running the Windows 8 operating system on an Intel i5-3317U (1.70 GHz) processor. The compiler cl.exe (version 11.00.50727.1) was used with the optimization flag/Ox. In analyzing the data, MATLAB 2013a Student Edition was used, and charts were created either with MATLAB or with Microsoft Excel 2013. Timing was done with the function QueryPerformanceCounter(). The code used to run these tests is available from the authors via email request.

In this section, the units of current have been omitted because all of the models use different units. The implicit units are: microamperes per square centimeter for the Hodgkin–Huxley model, dimensionless for the Izhikevich model, and nanoamperes for the leaky integrate-and-fire model.

### A. Frequency Errors

In calculating the frequency errors, as was stated in the accuracies of different solution methods section, the neuron models were run with three different input currents, and with time steps ranging from  $10^{-4}$  to 1 ms. The length of each run was 1000 ms. When calculating the frequency, the first spike was always omitted to correct for any discrepancy in variable initialization. The frequency was then calculated as

$$\text{Frequency}^{-1} = \frac{t_{\text{Last}} - t_{2\text{nd}}}{N_{\text{spikes}} - 1} \quad (22)$$

where  $t_{\text{Last}}$  was the time of the last action potential,  $t_{2\text{nd}}$  was the time of the second action potential, and  $N_{\text{spikes}}$  was the number of action potentials that occurred (with one subtracted to remove the first action potential).

When calculating the frequency errors, the frequencies were recorded to four significant figures, and the errors were reported to two decimal places. The benchmark frequency was always taken to be the frequency produced by the fourth-order Runge–Kutta method with a time step of  $\Delta t = 10^{-4}$  ms,



which was calculated separately for each model. The input currents used are listed in Table X.

### B. RMS Deviations

In calculating the rms deviation, the model was run for 50 ms with no input current. Then a current pulse of 18 was applied for 2.5 ms for the Hodgkin–Huxley and Izhikevich models, while a current pulse of 36 was applied for 5 ms for the leaky integrate-and-fire models. The discrepancy here arises from the fact that the leaky integrate-and-fire model would not spike for the same current pulse that spiked the Hodgkin–Huxley and Izhikevich models.

The rms deviations were calculated by comparing the voltage profile with the benchmark case. In all these runs, the benchmark case was chosen to be the voltage profile generated by the fourth-order Runge–Kutta method. For the Hodgkin–Huxley and leaky integrate-and-fire models, this benchmark case was taken to be from the time the membrane voltage rose above 0.5 V to the time it fell below 0.5 V. Cases to be compared with the benchmark were then made to be equally long to the (appropriately down sampled) benchmark case starting at the time where they first rose above 0.5 V. The only difference with the Izhikevich model is that the voltage of interest was  $-75.5$  V instead of 0.5 V (because that more accurately reflected its resting voltage).

## REFERENCES

- [1] S. Kampakis, "Investigating the computational power of spiking neurons with non-standard behaviors," *Neural Netw.*, vol. 43, pp. 41–54, Jul. 2013.
- [2] J. Hu, H. Tang, K. C. Tan, H. Li, and L. Shi, "A spike-timing-based integrated model for pattern recognition," *Neural Comput.*, vol. 25, no. 2, pp. 450–472, 2013.
- [3] Y. Xu, X. Zeng, and S. Zhong, "A new supervised learning algorithm for spiking neurons," *Neural Comput.*, vol. 25, no. 6, pp. 1472–1511, 2013.
- [4] S. Bohte, H. La Poutre, and J. N. Kok, "Unsupervised clustering with spiking neurons by sparse temporal coding and multi-layer RBF networks," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 426–435, Mar. 2001.
- [5] Y. Bodyanskiy, "Image processing using self-learning fuzzy spiking neural networks in the presence of overlapping classes," in *Proc. 12th Int. Biennial Baltic Electron. Conf.*, 2008, pp. 213–216.
- [6] Y. Wang, D. Chik, and Z. Wang, "Coherence resonance and noise-induced synchronization in globally-coupled Hodgkin–Huxley neurons," *Phys. Rev. E Statist. Phys. Plasmas Fluids Rel. Int. Topic.*, vol. 61, no. 1, pp. 740–746, 2000.
- [7] C. Batista, A. M. Batista, J. A. C. de Pontes, R. L. Viana, and S. R. Lopes, "Chaotic phase synchronization in scale-free networks of bursting neurons," *Phys. Rev. E*, vol. 76, no. 1, pp. 016218–1–016218–10, 2007.
- [8] D. Zanette and A. Mikhailov, "Mutual synchronization in ensembles of globally coupled neural networks," *Phys. Rev. E*, vol. 58, no. 1, pp. 872–875, 1998.
- [9] M. Zanin, D. Papo, I. Sendiña-Nadal, and S. Boccaletti, "Computation as an emergent feature of adaptive synchronization," *Phys. Rev. E*, vol. 84, no. 6, pp. 060102(R)–1–060102(R)–4, 2011.
- [10] W. Levy and O. Steward, "Temporal contiguity requirements for long-term associative potential/depression in the hippocampus," *Neuroscience*, vol. 8, no. 4, pp. 791–797, 1983.
- [11] Y. Dan and M. Poo, "Spike timing-dependent plasticity of neural circuits," *Neuron*, vol. 44, no. 1, pp. 22–30, 2004.
- [12] L. N. Long and A. Gupta, "Biologically-inspired spiking neural networks with Hebbian learning for vision processing," in *Proc. 46th AIAA Aerosp. Sci. Meeting*, Jan. 2008, no. 2008-0885.
- [13] S. Song, K. D. Miller, and L. F. Abbott, "Competitive Hebbian learning through spike-timing-dependent synaptic plasticity," *Nat. Neurosci.*, vol. 3, no. 9, pp. 919–926, Sep. 2000.
- [14] A. Gupta and L. N. Long, "Hebbian learning with winner take all for spiking neural networks," in *Proc. IEEE IJCNN*, Jun. 2009, pp. 1054–1060.
- [15] G. Q. Bi and M. M. Poo, "Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type," *J. Neurosci.*, vol. 18, no. 24, pp. 10464–10472, Dec. 1998.
- [16] A. L. Hodgkin and A. F. Huxley, "A quantitative description of ion currents and its applications to conduction and excitation in nerve membranes," *J. Physiol.*, vol. 117, no. 4, pp. 500–544, 1952.
- [17] N. Fourcaud-Trocmé and N. Brunel, "Dynamics of the instantaneous firing rate in response to changes in input statistics," *J. Comput. Neurosci.*, vol. 18, no. 3, pp. 311–321, 2005.
- [18] R. Jolivet, T. J. Lewis, and W. Gerstner, "Generalized integrate-and-fire models of neuronal activity approximate spike trains of a detailed model to a high degree of accuracy," *J. Neurophysiol.*, vol. 92, no. 2, pp. 959–976, Mar. 2004.
- [19] R. Muresan and I. Ignat, "Principles of design for large-scale neural simulators," *IEEE Trans. Neural Netw.*, vol. 15, no. 5, pp. 11–14, Feb. 2004.
- [20] E. M. Izhikevich, "Which model to use for cortical spiking neurons?" *IEEE Trans. Neural Netw.*, vol. 15, no. 5, pp. 1063–1070, Sep. 2004.
- [21] R. E. Plant, "Bifurcation and resonance in a model for bursting nerve cells," *J. Math. Biol.*, vol. 11, no. 1, pp. 15–32, 1981.
- [22] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE Trans. Neural Netw.*, vol. 14, no. 6, pp. 1569–1572, Nov. 2003.
- [23] L. N. Long, "Efficient neural network simulations using the Hodgkin–Huxley equations," in *Proc. Conf. Hodgkin Huxley, Trinity College*, Jul. 2012, pp. 1–10.
- [24] L. N. Long and G. Fang, "A review of biologically plausible neuron models for spiking neural networks," in *Proc. AIAA InfoTech. Aerosp. Conf.*, 2010, no. 2010-3540.
- [25] R. Warming and B. Hyett, "The modified equation approach to the stability and accuracy analysis of finite-difference methods," *J. Comput. Phys.*, vol. 14, no. 2, pp. 159–179, 1974.
- [26] C. Koch and I. Segev, *Methods in Neuronal Modeling: From Ions to Networks*, 2nd ed. Cambridge, MA, USA: MIT Press, 1999.
- [27] L. Euler, *Institutionum Calculi Integralis*, vol. 2. San Diego, CA, USA: Academic Press, 1769.
- [28] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, 9th ed. New York, NY, USA: Dover, 1972, pp. 896–897.
- [29] General Neural Simulation System, Zanesville, OH, USA. (2007). *Exponential Euler* [Online]. Available: <http://www.genesis-sim.org/BABEL/gum-tutorials/cornelis/doc/html/node9.html>
- [30] S. Wu, S.-I. Amari, and H. Nakahara, "Population coding and decoding in a neural field: A computational study," *Neural Comput.*, vol. 14, no. 5, pp. 999–1026, 2013.
- [31] E. Izhikevich, "Resonance-and-fire neurons," *Neural Netw.*, vol. 14, nos. 6–7, pp. 883–894, 2001.
- [32] W. Bialek, F. Rieke, R. R. de Ruyter van Steveninck, and D. Warland, "Reading a neural code," *Science*, vol. 252, no. 5014, pp. 1854–1857, 1991.
- [33] D. Jin and H. Seung, "Fast computation with spikes in a recurrent neural network," *Phys. Rev. E, Statist. Nonlinear Soft Matter Phys.*, vol. 65, pp. 051922–1–051922–3, May 2002.



**Michael J. Skocik** received the B.Sc. degree from The Pennsylvania State University, State College, PA, USA, in 2012, where he is currently pursuing the D.Phil. degree in physics.

He was with the Experimental High-Energy Physics Laboratory, Department of Energy's STAR Detector of the Relativistic Heavy Ion Collider, Brookhaven National Laboratory, and the Experimental Biophysics Laboratory, The Pennsylvania State University, where he designed systems to study the neural pathways of songbirds. He was inducted into Sigma Pi Sigma and the National Physics Honor Society in 2011, and has served as a Teaching Assistant for more than 175 undergraduate students.

Mr. Skocik is a recipient of the Department of Defense Science, Mathematics and Research for Transformation fellowship.



**Lyle N. Long** received the B.M.E. degree from the University of Minnesota, Minneapolis, MN, USA, the M.S. degree from Stanford University, Stanford, CA, USA, and the D.Sc. degree from George Washington University, Washington, DC, USA.

He is a Distinguished Professor of aerospace engineering, computational science, neuroscience, and mathematics with The Pennsylvania State University, State College, PA, USA. He is the Founder and Director of the Graduate Minor Program in Computational Science. He was the Founding Editor-in-

Chief of the *AIAA Journal of Aerospace Information Systems*. From 2007

to 2008, he was a Moore Distinguished Scholar with the California Institute of Technology, Pasadena, CA, USA. He was with Lockheed Aircraft and Thinking Machines Corporation, Cambridge, MA, USA. He has authored more than 250 journal and conference papers. His current research interests include neural networks, software engineering, high performance computing, robotics, and unmanned vehicles.

Dr. Long received the Penn State Engineering Society Outstanding Research Award in 1996, the 1993 IEEE Computer Society Gordon Bell Prize for achieving highest performance on a parallel computer, and the Lockheed Aeronautical Systems Company Award for excellence in research and development. He is a fellow of the American Physical Society and the American Institute of Aeronautics and Astronautics.