

# A Review of Biologically Plausible Neuron Models for Spiking Neural Networks

Lyle N. Long<sup>1</sup> and Guoliang Fang<sup>2</sup>  
*The Pennsylvania State University, University Park, Pennsylvania, 16802*

**In this paper, five mathematical models of single neurons are discussed and compared. The physical meanings, derivations, and differential equations of each model are provided. Since for many applications the spiking rates of neurons are of great importance, we compare the spiking rate patterns under different sustained current inputs. Numerical stability and accuracy are also considered. The computational cost and storage requirements needed to numerically solve each of the models are also discussed.**

## I. Introduction

Spiking neural networks (SNN) have become quite popular recently, due to their biological plausibility. Using spiking neuron models, SNN are able to encode spatial-temporal information into both spike timing and spiking rates, which has advantages over traditional spiking-rate based artificial networks, where only spiking rates are considered.

In applications, when choosing the neuron model for large SNN there is always a tradeoff between the biological plausibility and computational efficiency. For example, if the SNN is supposed to include the effect of complex biochemical reactions in the cortex, the Hodgkin-Huxley compartment model, with ionic channels should be used. On the other hand, if the computational efficiency is of more importance than biological plausibility, the Leaky Integrate and Fire (LIF) will most likely be adopted due to its low computational cost.

After the Hodgkin-Huxley (HH) neuron model was proposed [1], many simplified models were suggested, either based on simplifying the HH model according to its phase-plane behavior, or describing the ionic current behaviors using simpler perspectives. In this paper, we choose five common neuronal models, and study their biological plausibility and computational efficiency. Many interesting spiking behaviors exhibited by neuron models have been investigated using mathematical tools [2]. However, less attention has been paid to the numerical issues, i.e. accuracy and efficiency, when neuron models are numerically implemented, which, as demonstrated in this paper, might cause disastrous errors in SNN behaviors. Although we did discuss some SNN numerical issues in Ref. [11].

The outline of this paper is as follows: model descriptions and derivations are discussed in section II. The spiking frequency patterns of different models under sustained currents are provided in section III. In section IV, we discuss the numerical issues related to neuron models and algorithms. In section V, we discuss the model efficiencies. Conclusions are made in section VI.

## II. Introduction to Different Neuron Models

In this section we provide a basic introduction to the five neuron models under investigation. Their derivations, and mathematical descriptions (ordinary differential equations) are provided [3].

### A. Hodgkin-Huxley (HH) Model

In 1952 Hodgkin and Huxley [1] proposed a mathematical model to account for the electric current flow through the surface membrane of a squid giant nerve fiber. The Hodgkin-Huxley model is used to explain the different spiking phenomena of a neuron after it is exposed to various current stimulations. In their paper, the effects of different ionic channels to the capacity and resistance of the membrane were incorporated into the model; and empirical curve-fittings were used to generate the component functions for the equations. The Hodgkin-Huxley model is one of the most biological plausible models in computational neuroscience, and they won a Nobel prize for their research. Their model is a complicated nonlinear ODE system consisting of four equations describing the membrane potential, activation and inactivation of different ionic gating variables respectively. The HH equations are given as follows.

---

<sup>1</sup> Distinguished Professor of Aerospace Engineering, AIAA Fellow, LNL@psu.edu, <http://www.personal.psu.edu/lnl>

<sup>2</sup> Ph.D. Candidate, Applied and Computational Mathematics, 13 McAllister Bldg.

$$\frac{du}{dt} = -g_{Na}m^3h(u - E_{Na}) - g_Kn^4(u - E_K) - g_L(u - E_L) + I(t)$$

$$\frac{dm}{dt} = \alpha_m(u)(1 - m) - \beta_m(u)m$$

$$\frac{dn}{dt} = \alpha_n(u)(1 - n) - \beta_n(u)n$$

$$\frac{dh}{dt} = \alpha_h(u)(1 - h) - \beta_h(u)h$$

$$E_{Na} = 115 \text{ mV}, \quad E_K = -12 \text{ mV}, \quad E_L = 10.6 \text{ mV}$$

$$g_{Na} = 120 \text{ ms/cm}^2, \quad g_K = 36 \text{ ms/cm}^2, \quad g_L = 0.3 \text{ ms/cm}^2$$

$$\alpha_n(u) = \frac{0.1 - 0.01u}{\exp(1 - 0.1u) - 1}, \quad \alpha_m(u) = \frac{2.5 - 0.1u}{\exp(2.5 - 0.1u) - 1}, \quad \alpha_h(u) = 0.07 \exp\left(-\frac{u}{20}\right)$$

$$\beta_n(u) = 0.125 \exp\left(-\frac{u}{80}\right), \quad \beta_m(u) = 4 \exp\left(-\frac{u}{18}\right), \quad \beta_h(u) = \frac{1}{\exp(3 - 0.1u) + 1}$$

Correction from original paper

Here  $u(t)$  is the neuron membrane voltage, parameters  $g_{Na}$ ,  $g_K$ ,  $g_L$  are used to model channel conductances, e.g. the maximum conductance for  $Na^+$  channel is  $g_{Na}$ , and for  $K^+$  channel is  $g_K$ , when all channels are open. Additional variables  $m, n$  control the opening of  $Na^+$  channels, whereas  $h$  controls the opening of the  $K^+$  channel. The parameters  $E_{Na}, E_K, E_L$  are the reversal potentials.

## B. Izhikevich (IZ) Model

Using knowledge of bifurcation theory, Izhikevich (IZ) proposed his neuron model which is a two-dimensional system of ordinary differential equations [4]:

$$\frac{dv}{dt}(t) = 0.04v^2 + 5v + 140 - u + I(t)$$

$$\frac{du}{dt}(t) = a(bv - u)$$

with the auxiliary after-spike resetting of the form:

$$\text{if } v \geq 30 \text{ mV, then } \begin{cases} v = c \\ u = u + d \end{cases} \quad (1)$$

The variable  $v$  represents the voltage of a neuron, while the variable  $u$  functions as a recovery variable adjusting  $v$ .  $I(t)$  represents input currents, and  $a, b, c, d$  are adjustable parameters. When the membrane voltage  $v(t)$  reaches its apex (30 mV), a spike is emitted, and the membrane voltage and the recovery variable are reset to other values according to rule (1). The resting potential in the model is between -70 mV and -60 mV depending on the value of  $b$ . For example, to mimic regular spiking neurons, the time scale of the recovery variable  $a$  is set to be 0.02,  $b$  is 0.2,  $c$  is -65 mV, and  $d$  is 2.

### C. Wilson (WIL) Model

Hugh Wilson proposed his alternative to the Hodgkin-Huxley equations based on polynomial curve fitting of the original HH model [5]. The equations for this model are:

$$c \frac{dv}{dt} = -(17.81 + 47.71v + 32.63v^2)(v - 0.55) - 26.0R(v + 0.92) + I(t)$$

$$\frac{dR(t)}{dt} = \frac{1}{\tau} (-R + 1.35v + 1.03)$$

In his book  $c = 0.8 \mu F/cm^2$  and  $\tau = 1.9 ms$ ,  $v$  is the membrane potential,  $R$  is the recovery variable. In his model, the voltage  $v$  is scaled to be confined in a small region say  $[-2, 2]$  mV, and the current is scaled to be around 1 percent of the input current to the HH model.

### D. Fitzhugh-Nagumo (FITZ) Model

The FitzHugh-Nagumo model is an approximation to the Hodgkin-Huxley model, which comes from phase plane analysis. By mimicking the nullclines of the Hodgkin-Huxley model with one cubic function and one straight line, a polynomial reduction model of the following of the form was obtained [6].

$$\frac{dv}{dt} = v - \frac{v^3}{3} - w + I(t)$$

$$\frac{dw}{dt} = \varepsilon(v + a - bw)$$

Here, the parameters have been scaled, with the fast variable  $v$  representing the potential, and the slow variable  $w$  functioning as a recovery variable. Parameters  $a, b, \varepsilon$  are used to account for the time scale and kinetics of the recovery variable.

### E. Leaky Integrate and Fire (LIF) Model

The Leaky Integrate and Fire (LIF) Model, considers the neuron membrane as a capacitor  $C$  in parallel with a resistor  $R$  subject to a synaptic charging current  $I(t)$  [8].

$$\tau_m \frac{du}{dt} = -u(t) + RI(t) \quad (2)$$

where  $u(t)$  is the membrane potential and  $\tau_m = RC$  is the time constant of the neuron membrane. During depolarization, if the voltage  $u(t)$  reaches the firing threshold  $u_{th}$ , at the moment  $t^{(f)} : u(t^{(f)}) = u_{th}$ , the neuron will immediately generate a spike, and right after the spike, the membrane voltage is reset to the rest value  $u_r < u_{th}$ .

$$\lim_{t \rightarrow t^{(f)}, t > t^{(f)}} u(t) = u_r$$

For  $t > t^{(f)}$ , the dynamics of the model are again given by (2), until the next threshold crossing occurs. More generally, the leaky integrate and fire neuron may add an absolute refractory period  $t_r$  to limit its firing frequency, by preventing it from firing during that period: if  $u(t)$  reaches the threshold at time  $t = t^{(f)}$ , we ‘freeze’ the dynamics during an absolute refractory time, and set the membrane voltage to be  $u_r$ . At the time  $t^{(f)} + t_r$ , we restart the dynamics with initial condition  $u_r$ .

## III. Spiking Rates of Neuron Models under Sustained Currents

In this section we discuss the spiking rates of the different models subject to constant current inputs. For most neurons, their spiking process takes place in the time scale of milliseconds. The spiking frequencies of many real neurons are roughly 10-120 Hz [7]. Thus, spiking frequencies that are higher than 120 Hz are considered to be biologically impossible and are not discussed here, even though they can be obtained from some of these neuron models.

### A. The Spiking Rate of HH Model

To begin, we will show the spiking rates of the HH model under different sustained current inputs. And the spiking rate versus currents curve is obtained as follows: Numerically solve the Hodgkin-Huxley equations with constant current  $I$  for 10 seconds. When the voltage exceeds 20 mV, we count that as a spike. The accumulated numbers of spikes divided by the time period yields the spike rate  $R(I)$  (in Hertz). Starting from  $0 \mu A/cm^2$  the current is gradually increased by an increment of  $0.3 \mu A/cm^2$ ; finally we obtain the spiking rates graph for the Hodgkin-Huxley model vs. current, as shown in Figure 1. From the graph we can see the Hodgkin-Huxley model is

a type I neuron, which means the model cannot generate a spiking frequency that is lower than a special threshold (around 50 Hz). The spiking rate is an increasing function with respect to the current, but it is nonlinear [8].

### B. Spiking rate of IZ model

For the Izhikevich model different values of the four parameters  $(a,b,c,d)$  correspond to different types of spiking neurons. For example a typical value  $(a,b,c,d) = (0.02, 0.2, -65, 8)$  corresponds to an excitatory regular spiking pattern. Whereas  $(a,b,c,d) = (0.1, 0.2, -65, 2)$  is used to describe inhibitory fast spiking neurons.

For spiking patterns of the Izhikevich model under sustained constant current, we only chose the two typical cases mentioned above, regular spiking and fast spiking. Using similar procedures as was used to obtain the HH frequency pattern, we obtained the frequency vs. current graph for these two types of neurons (Figures 2a and 2b). Comparing the spiking patterns of the Izhikevich model to the HH model, it can be concluded that the firing pattern for the Izhikevich model is also an increasing function of current, however, the firing pattern is almost linear, whereas the curve for the HH model looks more like a square root function. The regular spiking IZ model belongs to a type II neuron, which means it can fire at arbitrarily small frequency, provided the current falls into a particular region. Whereas the fast spiking IZ model belongs to a type I neuron as the HH model does.

### C. Spiking rate of WIL model

The original Wilson model had different parameters from those that are frequently used. Thus, to maintain the original approximation intention of Wilson's Model, we adjusted the two parameters  $C$  and  $\tau$ , using nonlinear least squares, making the firing pattern of Wilson model as close as possible to that of HH model. Their values were  $C=1.5$  and  $\tau=4.2$ .

The frequency graph of the WIL model is shown in Figure 3. It can be seen that the curve resembles that of HH model, even though the input current for Wilson's model has a smaller scale.

### D. Spiking rate of FITZ model

Although the FitzHugh-Nagumo model can be used to explain several HH type neural phenomena, the two dimensional model is unable to yield biologically reasonable frequencies. With the given parameters the highest possible frequency is less than 9 Hz. It may be possible, however, to tune the parameters in FITZ model to produce higher frequencies. The recovery variable equation has a time variable  $\epsilon$ , and there is not a significant adjustment on the voltage. Another thing worth mentioning is that, within reasonable current range, the frequency pattern is not monotonic, which can be explained by the bifurcation property of the ODE. The non-monotonic frequency chart makes the FITZ model less attractive because its inability to differentiate the magnitudes of input currents. The results for the FITZ model are shown in Figure 4.

### E. Spiking rate of LIF model

For the leaky integrate and fire model with constant current we can explicitly deduce the time needed for the neuron to depolarize from rest state to voltage threshold. In this paper we set  $u_r = 0$ . At  $t_0 = 0$  the voltage is at rest state  $u_r$ , with the onset of current  $I_0$ , membrane voltage begins to increase. By integrating the equation to time  $t$  we get:

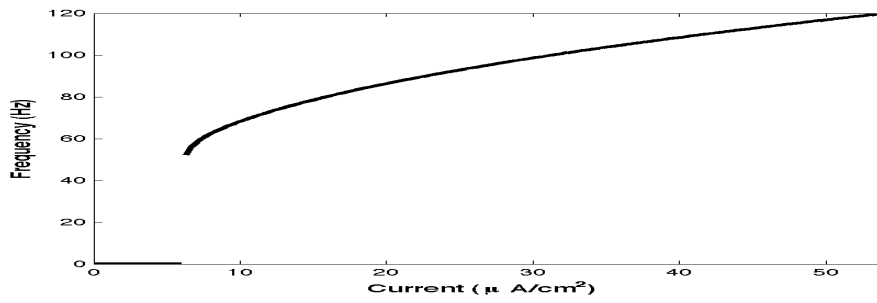
$$u(t) = RI_0[1 - e^{-\frac{t}{\tau_m}}]$$

when the voltage reaches its threshold  $u_{th}$ , the time lapsed is  $T = \tau_m \ln \frac{RI_0}{RI_0 - u_{th}}$ . The time from rest state to voltage threshold plus the refractory time is the length of one period. With this we can write the firing frequency as  $F(I_0) = [t_r + \tau_m \ln \frac{RI_0}{RI_0 - u_{th}}]^{-1}$ .

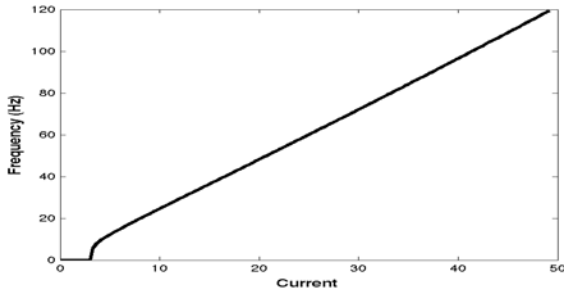
There are four parameters in this expression and we can tune them to make the firing frequency of the LIF model to be close to the Hodgkin-Huxley model, using nonlinear least squares. The 4 parameters we chose are:

$$R = 8.22 \text{ M}\Omega, C = 5.0675 \text{ nF}, u_{th} = 29.85 \text{ mV}, t_r = 5.17 \text{ ms},$$

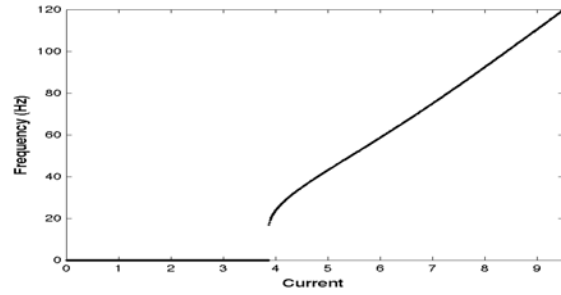
It can be seen from the LIF frequency graph (Figure 5) that the LIF model belongs to a type I neuron model, which can fire at arbitrarily small currents.



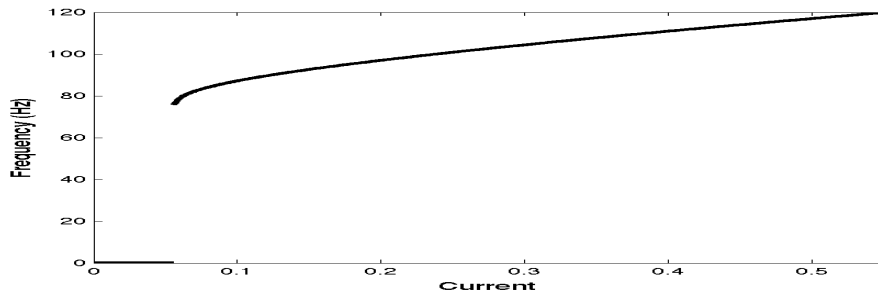
**Figure 1, Frequency curve for HH model under sustained currents**



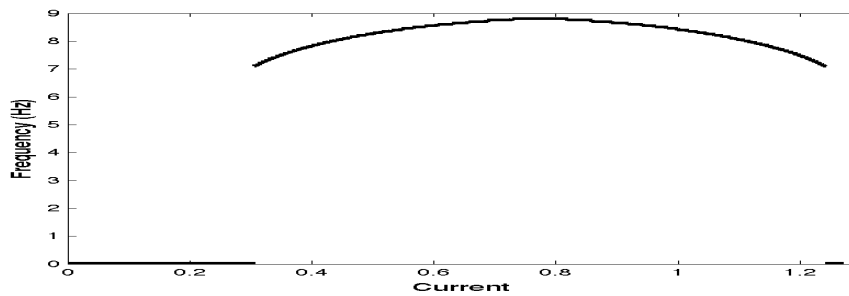
**Figure 2(a), Frequency curve for regular spiking IZ model under sustained currents**



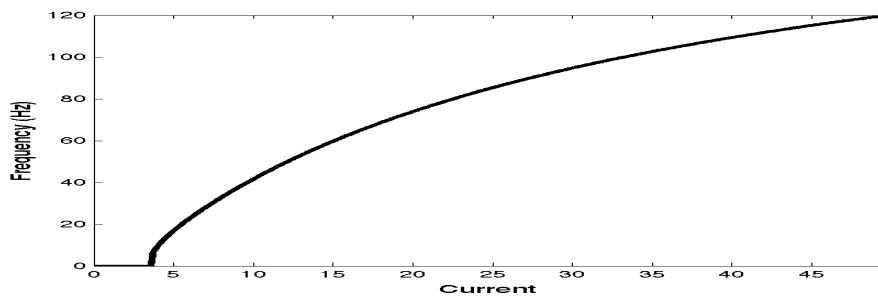
**Figure 2(b), frequency curve for fast spiking IZ model under sustained currents**



**Figure 3, Frequency curve for WIL model under sustained currents**



**Figure 4, Frequency curve for FITZ model under sustained currents**



**Figure 5, Frequency curve for LIF model under sustained currents**

## **F. The Voltage Patterns from Different Neuron Models**

In this section we presented the voltage patterns of different neuron models under select sustained constant current. The frequency patterns of the IZH, WIL, LIF models intercept the frequency pattern of HH model, in other words, there exists current values which will yield the same spiking frequency as the HH model. We will compare the voltage time histories at these particular current values. Also there are current values where the spiking rates of the IZH, WIL, LIF models differ significantly from the HH model. We compare the models at these values also. We also include the voltage pattern of the FITZ model for completeness, which is quite different from that of the HH model. For the voltage pattern of the WIL model, we scaled the voltage by a factor of 20, for better visualizations. And for the LIF model, we set a higher value at the spiking time to denote the formation of a spike also for better visualizations.

Figure 6 shows the voltage vs. time results for the HH model. These curves are very smooth, since the spikes are not caused by a thresholding operation. In addition, you can see the initial transient period where the spike amplitude varies from the subsequent spikes. The HH model allows varying spike amplitudes. Figure 9 shows the results from the IZ model simulations. The shift in levels is not too important. The shape of the waveforms agree fairly well with the HH model. Figure 10 shows the WIL model results, and you can see the spikes have finite width as does the HH model. The FITZ model results are shown in Figure 11. This model produces wave forms that are quite different from the HH model. Figure 12 shows the results for the LIF model. The spikes in the LIF model have very narrow widths since the spike is just a value inserted when a threshold has been crossed.

## **IV. Accuracy and Stability of Numerical Model Solver**

The accuracy of a numerical solver can obviously affect the behavior of the neural network. In SNN networks, the spike timing contains the tempo-spatial information from outside stimulus, inaccuracy in detecting spike timing can result in phenomena that is either not biological meaningful or spurious. In this section, we consider numerical issues such as algorithms, discrete time steps, which can affect the accuracy of the simulation.

### **A. Two Model Subgroups**

First we divide the five models we studied into two subgroups, according to whether they have an intrinsic ‘numerical safeguard’ or not. For the LIF and IZ models, once the model fires a spike, it will be immediately set to a rest value, and the integration evolution will start all over again. In this sense, if the previous spike time is accurately captured, the accumulated error from previous calculations will not affect next iteration cycle. Hence, it is less likely that the numerical result will become unstable even with a large time step due to accumulated round-off errors. This does not mean they are accurate at these time step sizes, and they might give different answers depending on the time step size. On the other hand, the HH, WIL, and FITZ models keep evolving, even after a spike has been detected. Thus the error keeps accumulating throughout the implementation, which can cause both stability and accuracy issues. Therefore we will categorize the LIF and IZ models into subgroup I, and the HH, WIL, and FITZ models into Subgroup II. We will mainly discuss the HH model as a representative of Subgroup II.

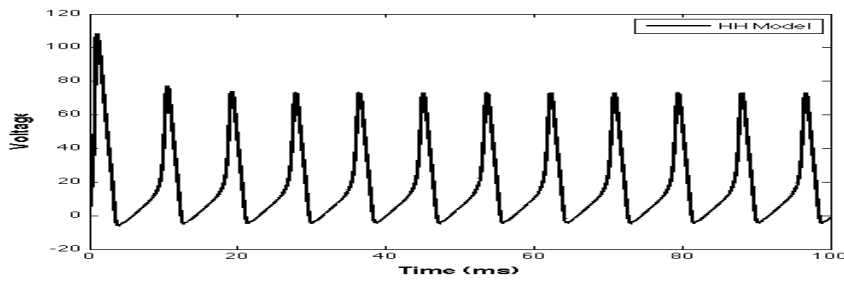


Figure 6, Voltage curve for HH model under sustained current (55)

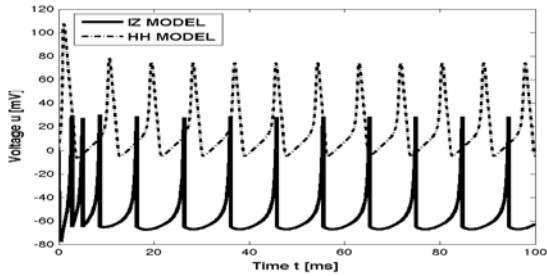


Figure 7a, Voltage curve for IZ regular model under one sustained current (47.2) and fire frequency is close to HH model

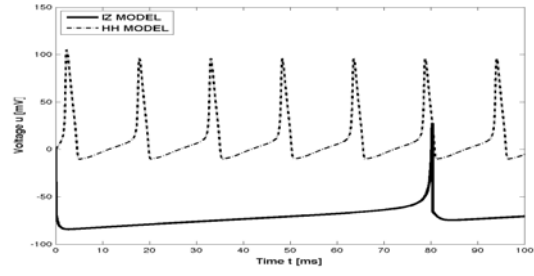


Figure 7b, Voltage curve for IZ regular model under one sustained current (8) and firing frequency is far from HH model

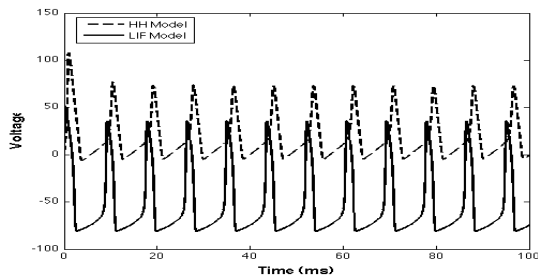


Figure 8a, Voltage curve for WIL model under one sustained current (55) and firing frequency is close to HH model

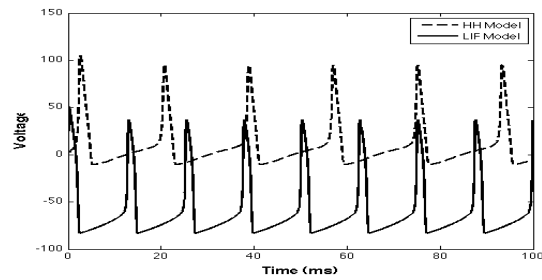


Figure 8b, Voltage curve for WIL model under one sustained current (8) and fire frequency is far from HH model

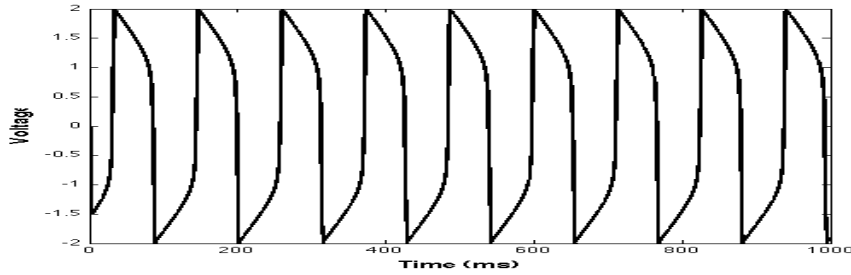


Figure 9, Voltage curve for FITZ model under one sustained current (0.7)

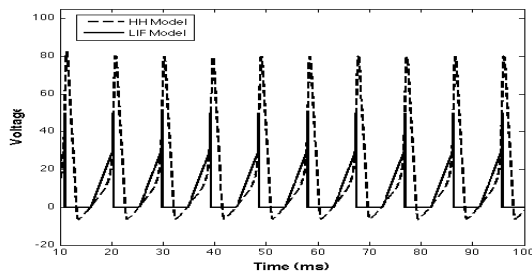


Figure 10a, Voltage curve for LIF model under one sustained current (34) and fire frequency is close to HH model

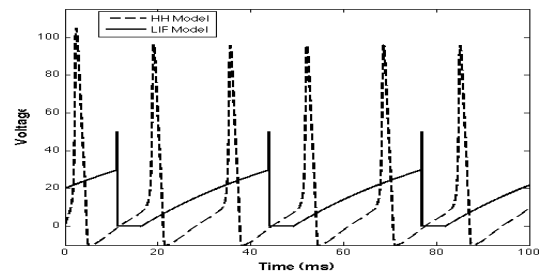


Figure 10b, Voltage curve for LIF model under one sustained current (8) and fire frequency is far from HH model

## B. The Numerical Stability and Accuracy for Model Subgroup I.

As mentioned above the models that belong to subgroup I have better stability properties due to the ‘safeguard’ settings. Let’s use a simple example to illustrate this. Consider a one-step forward Euler Scheme for the Izhikevich model.

$$\frac{v^{n+1} - v^n}{\Delta t} = 0.04(v^n)^2 + 5v^n + 140 - u^n + I(t_n)$$

$$\frac{u^{n+1} - u^n}{\Delta t} = a(bv^n - u^n)$$

$$\text{if } v^{n+1} \geq 30 \text{ mV, then } \begin{cases} v^{n+1} = c \\ u^{n+1} = u^{n+1} + d \end{cases}$$

According to computational experience for most of the time  $u^{n+1} < 140$ , hence once  $v^n \geq 0$ , we always have  $v^{n+1} > v^n$ , which implies the voltage keeps increasing no matter how large the time step. And once  $v^{n+1} \geq 30$  is detected, the system starts another cycle all over again. As a result, subgroup I models are more robust to time step size.

The loose constraints from stability concerns on the discrete time step seem enticing, however, a large discrete time step is still very likely to result in inaccuracies in spike timing and it could produce different answers depending on the time step size.

Figure 11 illustrates this idea. Suppose we are using high order accurate method, e.g. 4th order Runge-Kutta to obtain the numerical solution. Thus, even with large time step, the truncation error is negligible for one period of simulation, and we assume the numerical solution agrees fairly well with the exact solution at our discrete time points.

In figure 11, the black curve is the true trajectory of voltage under current input, and suppose if the voltage goes above 9, a spike is emitted, and the voltage goes to the rest state and continues to evolve from there. The X-axis is the time scale. If we use one unit as our numerical time step size, since  $v(3\Delta t) < 9$ ,  $v(4\Delta t) > 9$ , numerically we assume a spike will be emitted at time point  $4\Delta t$ , and the difference between the real spiking point and the numerical spiking point will be  $< \Delta t$ . Next if we tripled the time step size  $\Delta t^{new} = 3\Delta t$ , then  $v(\Delta t^{new}) < 9$  and  $v(2\Delta t^{new}) > 9$ , numerically a spike is emitted at  $2\Delta t^{new} = 6\Delta t$ . And the difference between the real spiking point and the numerical spiking point is  $> 2\Delta t$ . Neurons in the network will now experience this spike later. The post-synaptic behaviors are dependant on the timing of pre-synaptic spikes. And an erroneous spike timing report in the spike timing might cause a large effect to the whole network evolution. In addition, the network behavior will be dependant upon the time step size.

## B. The Numerical Stability and Accuracy for Model Subgroup II

For neuron models that belong to subgroup II, stability issues and accuracy issues should also be taken into consideration. In this section, we take the HH model as representative. For the HH model, due to its complex nonlinearity, it is difficult to perform an analytical stability analysis. According to our simulation work, as well as work in [9], the stability requirement for the forward Euler scheme is roughly  $\Delta t \leq 0.07$ , that of a standard Runge-Kutta 4<sup>th</sup> order is  $\Delta t \leq 0.08$ . And these will vary if the HH coefficients are changed. For models that belong to Subgroup II, both the time step and numerical scheme that we chose will affect the neuron spike timing in the following way:

Suppose  $v(t)$  is the real underlying unknown solution,  $v^n(l\Delta t)$  is the numerical solution we obtained using certain scheme, at a multiple of time step  $\Delta t$ ,  $err(l\Delta t, \Delta t)$  is the cumulative error between the real voltage and the numerical one, then the spike detecting

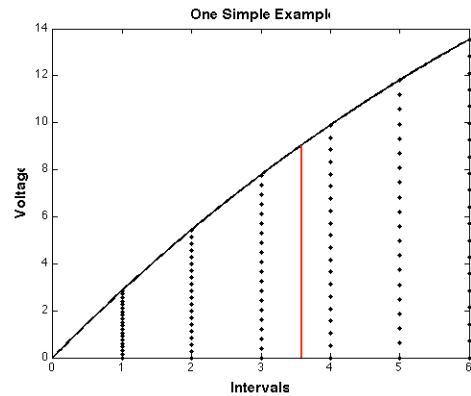


Figure 11. One example to illustrate the accuracy issue accompanied with subgroup I.

criteria:  $v^n(I\Delta t) \geq v_{th} \Leftrightarrow v(I\Delta t) + err(I\Delta t, \Delta t) \geq v_{th}$ , since the time step  $\Delta t$  is always small due to the stability constraints, an important factor that determines the accuracy is cumulative error  $err(I\Delta t, \Delta t)$ . In addition, since there is no intrinsic safeguard mechanism in the model, the cumulative error will dominate the accuracy of spike timing in the evolution process.

### C. Tests for Numerical Stability and Accuracy Issues

In this section we discuss several numerical tests related to accuracy issues. First we examine how different time steps and numerical schemes can effect the firing frequency of a model. For the HH, IZ, WIL models, we choose particular currents, such that the resulting frequency is around 80 Hz. We applied two numerical methods: a 1<sup>st</sup> order Forward Euler method and a 4<sup>th</sup> order Runge-Kutta method. We also used different time step sizes. The numerical frequencies obtained are shown in the following graphs. Note that the time step sizes and vertical scale are different for the HH model than the others.

From Fig. 12 we can see that, for the HH model, we get a consistent firing rate as long as the time step size is smaller than 0.05 ms. For the IZ and LIF models, the time step size needs to be below 0.01 or 0.001 before the solution converges to a steady value. For a large time step size ( $\Delta t = 1$  ms), the error in the frequency is around 10% per time step, which is enormous. Even for a moderate time step ( $\Delta t = 0.1$  ms), the error in frequency is still apparent.

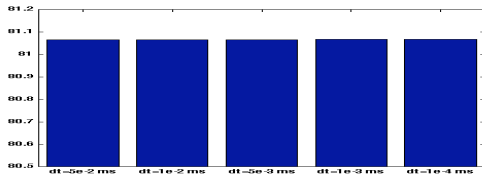


Figure 12a. HH frequency using forward FE scheme

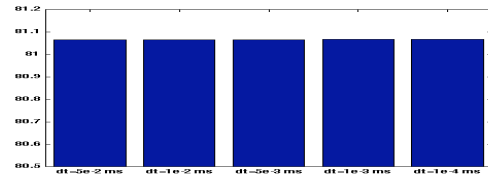


Figure 12b. HH frequency using RK4 scheme

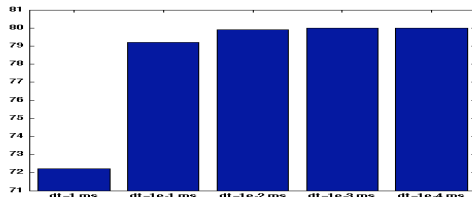


Figure 13a. IZ frequency using forward FE scheme

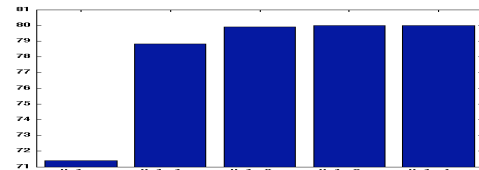


Figure 13b. IZ frequency using RK4 scheme

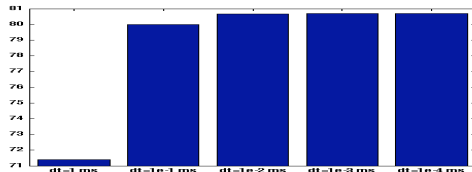


Figure 14a. LIF frequency using FE Scheme

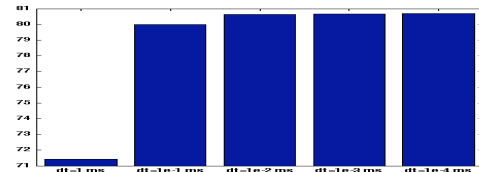


Figure 14b. LIF frequency using RK4 scheme

To explore the accuracy issues in more detail we did an additional comparison. We numerically solved the frequency vs. current functions  $F^S(I, \Delta t)$  for the three models using different schemes and time steps. Then we used the result obtained from the 4<sup>th</sup> order Runge-Kutta method with a small time step size ( $10^{-4}$ ) to be representative of the real solution  $F(I)$ . And we computed the  $l^2$  norm of the error  $\|F^S(I, \Delta t) - F(I)\|_2 = \left( \int (F^S(I, \Delta t) - F(I))^2 dI \right)^{\frac{1}{2}}$ . These results are shown in Tables 1, 2, and 3.

Some interesting results can be obtained from these tables. First for HH model, using a RK 4<sup>th</sup> order scheme with a bigger time step  $\Delta t = 0.05$  the scheme is more accurate than using a FE scheme with a very fine time step  $\Delta t = 0.001$ , as we'd expect. We would also like to mention that, with the FE scheme the error is tolerable, as long as the time step size is small enough. The main error for the HH case comes from the threshold current that is used to define a spike from the HH model. The FE scheme with larger time step size cannot capture this critical value as accurately as the RK 4<sup>th</sup> scheme. Also, when the current is larger, the frequencies obtained from the various methods are closer to each other. As discussed in Ref. [11], the time step required will depend on the frequency.

For the IZ model, when  $\Delta t = 1$ , the RK 4<sup>th</sup> scheme fails to find a solution since it goes unstable. When the time steps gets smaller, the error from the RK 4<sup>th</sup> scheme is roughly 50% of the FE scheme, but the advantage is not apparent. To get an error less than about 0.1 % you would need to use a time step size of roughly 0.01 ms. This is still a large error since this is per time step, and we often run for hundreds of thousands of time steps. While the FE is stable for time step sizes of 1 ms, the solution is not accurate. Or put another way, the equation you are solving is not the ODE you started with. The discrete form of the equation differs from the original ODE by about 10%. So the solutions obtained using large time steps cannot be replicated when using smaller time steps.

Explicit Euler Scheme		4 <sup>th</sup> order Runge-Kutta Scheme	
Time step	L <sup>2</sup> norm of error	Time step	L <sup>2</sup> norm of error
dt= 5*10 <sup>-2</sup> ms	9.99	dt= 5*10 <sup>-2</sup> ms	0.001
dt=10 <sup>-2</sup> ms	0.79	dt=10 <sup>-2</sup> ms	0
dt= 5*10 <sup>-3</sup> ms	0.39	dt= 5*10 <sup>-3</sup> ms	0
dt=10 <sup>-3</sup> ms	0.12	dt=10 <sup>-3</sup> ms	0
dt=10 <sup>-4</sup> ms	0.002	dt=10 <sup>-4</sup> ms	Benchmark Solution

**Table 1. L<sup>2</sup> error norm of HH model frequency graph, obtained using different schemes and time steps**

Explicit Euler Scheme		4 <sup>th</sup> order Runge-Kutta Scheme	
Time step	L <sup>2</sup> norm of error	Time step	L <sup>2</sup> norm of error
dt= 10 <sup>0</sup> ms	9.67	dt= 10 <sup>0</sup> ms	NA*
dt=10 <sup>-1</sup> ms	1.26	dt=10 <sup>-1</sup> ms	0.5
dt= 10 <sup>-2</sup> ms	0.13	dt= 10 <sup>-2</sup> ms	0.056
dt=10 <sup>-3</sup> ms	0.032	dt=10 <sup>-3</sup> ms	0.0047
dt=10 <sup>-4</sup> ms	0.0037	dt=10 <sup>-4</sup> ms	Benchmark Solution

\* (for time step =1 ms, RK 4<sup>th</sup> order method failed at some point to obtain the solution)

**Table 2. L<sup>2</sup> error norm of IZH model frequency graph, obtained using different schemes and time steps**

Explicit Euler Scheme		4 <sup>th</sup> order Runge-Kutta Scheme	
Time step	L <sup>2</sup> norm of error	Time step	L <sup>2</sup> norm of error
dt= 10 <sup>0</sup> ms	7.19	dt= 10 <sup>0</sup> ms	7.68
dt=10 <sup>-1</sup> ms	0.85	dt=10 <sup>-1</sup> ms	0.77
dt= 10 <sup>-2</sup> ms	0.094	dt= 10 <sup>-2</sup> ms	0.088
dt=10 <sup>-3</sup> ms	0.014	dt=10 <sup>-3</sup> ms	0.0098
dt=10 <sup>-4</sup> ms	0.0027	dt=10 <sup>-4</sup> ms	Benchmark Solution

\* (for time step =1 ms, RK 4<sup>th</sup> order method failed at some point to obtain the solution)

**Table 3. L<sup>2</sup> error norm of IZH model frequency graph, obtained using different schemes and time steps**

For the LIF model, the error is almost completely determined by the time step size. The RK 4<sup>th</sup> order scheme has little advantage over the FE scheme. And we mention again the reason is because the cumulative error within one voltage period is not effective, due to the ‘safeguard’ resetting, the accuracy is maintained. This is discussed in more detail in Ref. [11], where we showed that it was not even worth using the exact solution to the LIF model, since the time evolution occurs over such short periods and the exact solution requires computing exponential function. Therefore many of the interesting effects presented in Ref. [3] are fortuitous.

Shown in Fig. 15 are firing frequency graphs obtained using the FE scheme with larger time steps, compared with an accurate solution (time step size of  $10^{-4}$ ) for all three models. For the HH model, discrepancies only occur at lower current input (spiking threshold) [10], so we have only shown that region for better visualization.

To show the importance of spike timing, we used a deterministic neural network, first proposed by Izhikevich [3]. The network consisted of 80 excitatory neurons (regular spiking type), and 20 inhibitory neurons (fast spiking type). The synaptic connections between neurons are randomly set at the beginning, and fixed for all simulations. All neurons receive external currents, the input currents for excitatory neurons are 7, and those for inhibitory neurons are 3, and a spike from one neuron will cause a change of input current to neurons that are connected to it. We plotted the firing patterns of the network obtained using the forward Euler scheme under different time steps in Figure 16. A black dot represents a neuron fired at a certain time. We simulated the network for  $\Delta t = 1$ ,  $\Delta t = 10^{-1}$ , and  $\Delta t = 10^{-4}$ . We use the pattern obtained when  $\Delta t = 10^{-4}$  as a representative of the exact solution. From the figure we can see some interesting phenomena: first of all, according to the accurate solution, inhibitory neurons never fire,

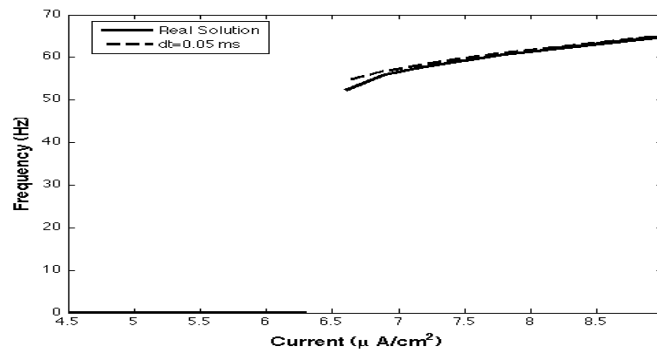


Figure 15a. HH model using different time step sizes

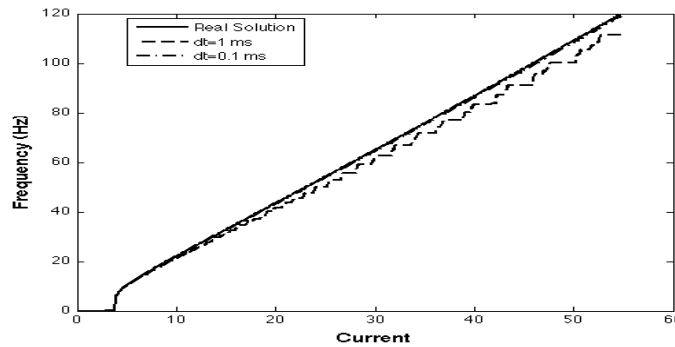


Figure 15b. IZ model using different time step sizes

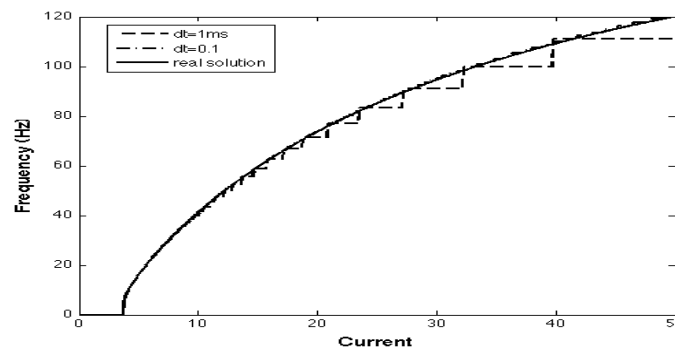


Figure 15c. LIF model using different time step sizes

however, if the time step is taken to be large enough, we got spurious spikes from inhibitory neurons. Secondly, the spikes obtained using bigger time steps are more sporadic than the spikes obtained using small time step. In other words small time step achieved a better synchronization in this case. Last but not least, the spike train timing is also affected by the time step we use. We can see for the middle case when  $\Delta t = 10^{-1}$ , although the firing pattern resembles that of real solution, there is a visible difference at the arriving time of the spike train.

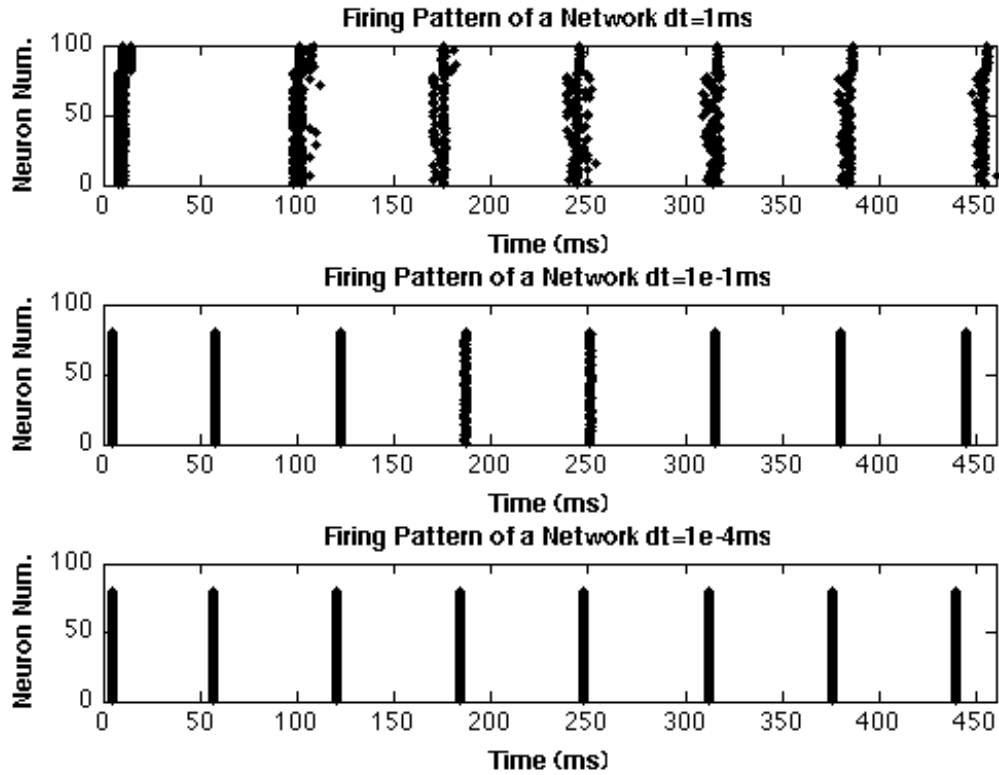


Figure 16. The spiking pattern of a network under different time steps

## V. Computational Cost

In this section, we will discuss the issues related to the cost of neuron model simulations, such as: the numbers of variables needed to be stored, floating point operations (flops) needed to implement one voltage update, and the CPU time needed to simulate different neuron models. For all numerical discretizations involved in this section, we use the explicit Forward Euler scheme. The 4<sup>th</sup> order RK scheme could be implemented to use the same memory, but it would cost about 4 times more in CPU time per step. The computational cost of a single neuron model will determine the capacity as well as the speed of a neural network consisting of these models. In the table below we show the name of models, the numbers of variables needed to be stored and the floating-point operations needed for one-step voltage update. Based on modern compilers, our estimation is: addition and multiplication needs 1 operation, division needs 7 operations, and exponential needs 10 operations. Table 4 shows the number of variables required and the number of flops needed for each update.

From Table 4 it can be seen that the storage requirement for HH model is 2 times bigger than the storage of IZ, WIL, and FZ model, and 4 times greater than that of LIF model. The HH model, with most significant biological plausibility, however, requires 12 times more operations than LIF, and is the most expensive model to solve.

To test the CPU time needed to carry out the simulation, we used the simulation task to get the frequency vs. current graphs used in the discussions above. The procedures are as follows: let the current increase by a unit of 0.35, from 0 to 70. For each fix current value, solve the ODE models for 10 seconds, the time step we take for this task is  $\Delta t = 10^{-2}$  ms. First we compare the CPU time for solving different models using different time steps. The codes were implemented on two identical 2.8 GHz 8-core Mac OS X servers with 24 GB RAM using GNU C++ version 4.0.1. And the comparison results are shown below in Table 5. The models don't differ significant from each other in terms of CPU time, except that the HH model requires the most amount of CPU time which is more than 30 times that of LIF model.

Model	Number of Variables	FLOPS per Step
HH	4	126
IZ	2	17
WIL	2	22
FITZ	2	13
LIF	1	9

Table 4. CPU time comparison between five models using forward Euler scheme for the task dt=0.01 millisecond,

Model	Computer I		Computer II	
	Real CPU Time	Relative CPU Time	Real CPU Time	Relative CPU Time
HH	64.57 seconds	30.26	85.92 seconds	32.30
IZ	4.34 seconds	2.03	7.88 seconds	2.96
WIL	4.47 seconds	2.12	8.11 seconds	3.05
FITZ	4.26 seconds	2.01	7.18 seconds	2.7
LIF	2.13 seconds	1	2.66 seconds	1

Table 5. CPU time comparison between five models using forward Euler scheme for the task dt=0.01 millisecond,

## VI. Conclusions

In this paper we evaluated five different neuron models for possible use in spiking neural networks. The models have a wide range of possible biological plausibility and computational difficulty. While previous studies have suggested that very large time step sizes can be used, this results in solutions which do not converge to the same answer when the time step size is reduced. A time step size of roughly 0.01 to 0.001 ms is required to obtain an reasonably accurate solution for all the methods, although for the HH model there is an advantage to using the 4<sup>th</sup> order RK method.

## References

1. A. L. Hodgkin., and A. F. Huxley., "A quantitative description of membrane current and application to conduction and excitation in nerve," *J. Physiol.*, Vol. 117, pp. 500–544, 1954.
2. Izhikevich, E.M., "Neural excitability, spiking, and bursting" *International Journal of Bifurcation and Chaos* Vol. 10, 2000, pp. 1171–1266.
3. E. M. Izhikevich, "Which Model to Use for Cortical Spiking Neurons?," *IEEE Transactions on Neural Networks*, Vol. 15, No. 5, Sep. 2004, pp.1063-1070.
4. E. M. Izhikevich, "Simple Model of Spiking Neurons" *IEEE Transactions on Neural Networks*, Vol. 14, No. 6, Nov. 2003, pp.1569-1572.
5. Wilson, H. R., *Spikes, Decisions, and Actions: The Dynamical Foundations of Neuroscience*, 1st ed., Oxford Univ. Press, Oxford,1999, Chaps. 5,6,7.

6. R. Fitzhugh, "Mathematical models of threshold phenomena in the nerve membrane," *Bull. Math. Biophysics*, Vol. 17, 1955, pp.257-278.
7. Koch, C., *Biophysics of Computation: Information Processing in Single Neurons*. 1999: Oxford Press.
8. Gerstner, W., Kistler, W. I., *Spiking Neuron Models. Single Neurons, Populations, Plasticity*, Cambridge University Press, 2002.
9. Y. Sun., D. Zhou., A. V. Rangan., D. Cai., "Library-based numerical reduction of the Hodgkin-Huxley neuron for network simulation." *Journal of Computational Neuroscience*, Vol. 27, 2009, pp.369-390.
10. Gupta, A. and Long, Lyle N., "Hebbian Based Learning with Winner-Take-All for Spiking Neural Networks," APS March Meeting, 2009, *Bull. of American Physical Society (APS)*, Vol. 53, No. 1, 2009.
11. Long, Lyle N. and Gupta, Ankur, "Biologically-Inspired Spiking Neural Networks with Hebbian Learning for Vision Processing," AIAA Paper No. 2008-0885, presented at the 46th AIAA Aerospace Sciences Meeting, Reno, NV, Jan. 7-10, 2008.