



# Exponential Time Differencing Algorithm for Pulse-Coupled Hodgkin-Huxley Neural Networks

Zhong-qi Kyle Tian and Douglas Zhou\*

School of Mathematical Sciences, MOE-LSC, Institute of Natural Sciences, Shanghai Jiao Tong University, Shanghai, China

The exponential time differencing (ETD) method allows using a large time step to efficiently evolve stiff systems such as Hodgkin-Huxley (HH) neural networks. For pulse-coupled HH networks, the synaptic spike times cannot be predetermined and are convoluted with neuron's trajectory itself. This presents a challenging issue for the design of an efficient numerical simulation algorithm. The stiffness in the HH equations are quite different, for example, between the spike and non-spike regions. Here, we design a second-order adaptive exponential time differencing algorithm (AETD2) for the numerical evolution of HH neural networks. Compared with the regular second-order Runge-Kutta method (RK2), our AETD2 method can use time steps one order of magnitude larger and improve computational efficiency more than ten times while excellently capturing accurate traces of membrane potentials of HH neurons. This high accuracy and efficiency can be robustly obtained and do not depend on the dynamical regimes, connectivity structure or the network size.

## OPEN ACCESS

### Edited by:

Thomas Nowotny,  
University of Sussex, United Kingdom

### Reviewed by:

Lianchun Yu,  
Lanzhou University, China  
Marcel Stimberg,  
INSERM U968 Institut de la Vision,  
Université de la Sorbonne, France

### \*Correspondence:

Douglas Zhou  
zdz@situ.edu.cn

**Received:** 20 November 2019

**Accepted:** 16 April 2020

**Published:** 08 May 2020

### Citation:

Tian ZK and Zhou D (2020)  
Exponential Time Differencing  
Algorithm for Pulse-Coupled  
Hodgkin-Huxley Neural Networks.  
*Front. Comput. Neurosci.* 14:40.  
doi: 10.3389/fncom.2020.00040

**Keywords:** Hodgkin-Huxley, exponential time differencing method, efficiency, pulse-coupled, second-order

## 1. INTRODUCTION

The Hodgkin-Huxley (HH) model (Hodgkin and Huxley, 1952; Hassard, 1978; Dayan and Abbott, 2003) is a classical neuron model, originally proposed to describe the behaviors of action potentials of the squid's giant axon. It provides a useful mechanism that accounts for the detailed generation of action potentials and the existence of the absolute refractory periods. It also serves as the foundation for other neuron models such as the one that can describe the behaviors of bursting and adaption (Pospischil et al., 2008). However, the HH equations are so complicated that it is difficult to study its properties analytically such as the Hopf bifurcation and chaotic dynamics (Aihara, 1986; Hansel and Sompolinsky, 1996; Guckenheimer and Oliva, 2002; Lin, 2006). Therefore, its investigation often relies on numerical simulations, for example, by the Runge-Kutta (RK) methods.

There are several difficulties to design an efficient and accurate numerical algorithm for the HH neural network, especially when the network size is large. First, when an HH neuron driven by external input generates an action potential (the interval of action potential is called spike period in this work), the HH neuron equations become stiff. Regular RK methods have to use very small time step to satisfy the requirement of numerical stability (Guckenheimer and Oliva, 2002; Börgers et al., 2005; Kassam and Trefethen, 2005; Börgers and Nectow, 2013). This small time step will significantly increase the computational cost when studying long time behavior of large-scale HH networks such as chaotic attractor dynamics or collecting reliable statistical information of HH neurons such as the distribution of inter-spike-intervals.

For more realistic situations, the neurons are generally driven by stochastic spike input and the interaction term is usually modeled by a Dirac delta function (pulse-coupled), while the spike-induced conductance dynamics are modeled by an *alpha* function (Somers et al., 1995; Hansel et al., 1998; Sun et al., 2009). These make the system become non-smooth and event-driven, while providing challenges for the design of efficient numerical simulation algorithms. For instance, it is impossible to predetermine the synaptic spike times since they are convoluted with neurons' trajectories themselves. As a result, one has to evolve the HH network by ignoring the spike interactions among neurons and then use spike-spike interaction to amend the neurons' trajectories at the end of the time step (Hansel et al., 1998; Brette et al., 2007). Without a careful recalibration for the neuronal spikes, the numerical algorithm often suffers from the issue of instability or relatively low numerical accuracy.

The exponential time differencing (ETD) method (Hochbruck et al., 1998; Cox and Matthews, 2002; Kassam and Trefethen, 2005; de la Hoz and Vellido, 2008; Nie et al., 2008; Hochbruck and Ostermann, 2010) is proposed for efficient simulation of stiff ordinary differential equations (ODEs). The basic idea is to decompose the ODEs into a linear stiff part and a nonlinear non-stiff part. Then, the linear stiff part can be solved by using the integrating factor method, while the nonlinear non-stiff part can be approximated by numerical quadrature (Cox and Matthews, 2002). A second-order ETD (ETD2) method for HH neural networks has been proposed in a recent work (Börgers and Nectow, 2013), which allows using a large time step to raise computational efficiency. In Börgers and Nectow (2013), the HH equations are linearly approximated in each time step, and then solved analytically over the time step. The ETD2 method proposed in Börgers and Nectow (2013) is a reduced situation of that in Cox and Matthews (2002), but it is difficult to generalize to higher-order cases, e.g., the fourth-order ETD method. Besides, although the ETD2 method proposed in Börgers and Nectow (2013) is proven to be unconditionally stable for HH system, it will be inaccurate using a large time step (Börgers and Nectow, 2013).

In this work, we first provide an ETD2 method following the idea proposed in Cox and Matthews (2002) to evolve a pulse-coupled HH neural network driven by stochastic spike input. Note that the stiffness of HH equations are quite different, especially between the spike and non-spike periods, and we find that the ETD2 method may introduce a relatively large error in the membrane potentials in the non-stiff period if using the same time step as that in the stiff period. We then design an adaptive ETD2 method (AETD2) that using different decompositions of the linear and non-linear parts in stiff and non-stiff periods. In addition, for the situation where neurons generate spikes in the time step, the effects of the spikes are carefully recalibrated in our AETD2 method to achieve a second-order numerical accuracy. Our AETD2 method is capable of using a large time step, while achieving the same high accurate traces of membrane potential of each neuron as the second-order RK (RK2) method using a very small time step. It can improve computational efficiency more than one order of magnitude compared with the RK2 method. This high numerical accuracy and computational efficiency can

be achieved over a wide range of dynamical regimes and does not depend on the network connectivity or size.

## 2. MATERIALS AND METHODS

### 2.1. The Model

The dynamics of the  $i$ th neuron of an HH neural network is governed by

$$C \frac{dV_i}{dt} = -(V_i - V_{Na})G_{Na}m_i^3h_i - (V_i - V_K)G_Kn_i^4 - (V_i - V_L)G_L + I_i^{\text{input}}, \quad (1)$$

$$\frac{dz_i}{dt} = (1 - z_i)\alpha_z(V_i) - z_i\beta_z(V_i), \quad \text{for } z = m, h, n, \quad (2)$$

where  $C$  is the cell membrane capacitance,  $V_i$  is the membrane potential,  $m_i$ ,  $h_i$ , and  $n_i$  are gating variables for sodium and potassium currents, respectively (Dayan and Abbott, 2001). The parameters  $V_{Na}$ ,  $V_K$ , and  $V_L$  are the reversal potentials for the sodium, potassium, and leak currents, respectively,  $G_{Na}$ ,  $G_K$ , and  $G_L$  are the corresponding maximum conductances. The form of  $\alpha_z$  and  $\beta_z$  are set as (Dayan and Abbott, 2001):  $\alpha_m(V) = (0.1V + 4)/(1 - \exp(-0.1V - 4))$ ,  $\beta_m(V) = 4 \exp(-(V + 65)/18)$ ,  $\alpha_h(V) = 0.07 \exp(-(V + 65)/20)$ ,  $\beta_h(V) = 1/(1 + \exp(-3.5 - 0.1V))$ ,  $\alpha_n(V) = (0.01V + 0.55)/(1 - \exp(-0.1V - 5.5))$ , and  $\beta_n(V) = 0.125 \exp(-(V + 65)/80)$ .

The input current  $I_i^{\text{input}}$  is given by

$$I_i^{\text{input}} = -G_i^E(t)(V_i - V_G^E) - G_i^I(t)(V_i - V_G^I), \quad (3)$$

where  $G_i^E$  and  $G_i^I$  are excitatory and inhibitory conductances, respectively,  $V_G^E$  and  $V_G^I$  are the corresponding reversal potentials. The dynamics of conductance  $G_i^Q$ ,  $Q = E, I$ , is governed by

$$\frac{dG_i^Q}{dt} = -\frac{G_i^Q}{\sigma_r^Q} + H_i^Q, \quad (4)$$

$$\frac{dH_i^Q}{dt} = -\frac{H_i^Q}{\sigma_d^Q} + F^Q \sum_l \delta(t - s_{il}) + \sum_j S_{ij}^Q \sum_l \delta(t - \tau_{jl}), \quad (5)$$

where  $H_i^Q$  is an auxiliary dynamical variable to make the conductance  $G_i^Q$  as a continuous function,  $\delta(\cdot)$  is the Dirac delta function,  $s_{il}$  is the spike time of the feedforward Poisson input with strength  $F^Q$  and rate  $\nu$ ,  $\tau_{jl}$  is the  $l$ th spike time of the  $j$ th neuron, and  $\sigma_d^Q$  and  $\sigma_r^Q$  are slow decay and fast rise time scale, respectively. Each neuron is either excitatory or inhibitory and its coupling strength is labeled by its type  $E$  or  $I$ , respectively. For example,  $S_{ij}^E$  ( $S_{ij}^I$ ) is the coupling strength from the  $j$ th excitatory (inhibitory) neuron to its postsynaptic  $i$ th neuron. By analytically solving Equations (4) and (5), the spike-induced conductance change can be explicitly expressed as

$$\mathbb{G}(\sigma_d^Q, \sigma_r^Q, t) = \frac{\sigma_d^Q \sigma_r^Q}{\sigma_d^Q - \sigma_r^Q} (e^{-t/\sigma_d^Q} - e^{-t/\sigma_r^Q}) \Theta(t), \quad (6)$$

where  $\Theta(\cdot)$  is the Heaviside function. For all neurons, we take  $F^E = f$  and  $F^I = 0$ . The model parameters are  $C = 1\mu\text{F}\cdot\text{cm}^{-2}$ ,  $V_{Na} = 50\text{ mV}$ ,  $V_K = -77\text{ mV}$ ,  $V_L = -54.387\text{ mV}$ ,  $G_{Na} = 120\text{ mS}\cdot\text{cm}^{-2}$ ,  $G_K = 36\text{ mS}\cdot\text{cm}^{-2}$ ,  $G_L = 0.3\text{ mS}\cdot\text{cm}^{-2}$ ,  $V_G^E = 0\text{ mV}$ ,  $V_G^I = -80\text{ mV}$ ,  $\sigma_r^E = 0.5\text{ ms}$ ,  $\sigma_d^E = 3.0\text{ ms}$ ,  $\sigma_r^I = 0.5\text{ ms}$ , and  $\sigma_d^I = 7.0\text{ ms}$  (Dayan and Abbott, 2001).

The voltage  $V_i$  evolves continuously according to Equations (1) and (2). When it reaches the firing threshold  $V^{\text{th}} = -50\text{ mV}$  (Sun et al., 2009), we say the  $i$ th neuron generates a spike at this time, say  $\tau_{il}$ . Then it will trigger its postsynaptic  $j$ th neuron's conductance change in the form of  $S_{ji}^Q \mathbb{G}(\sigma_d^Q, \sigma_r^Q, t - \tau_{il})$ ,  $Q = E, I$ . For the ease of discussion about our algorithm design, we consider an all-to-all connected network with  $S_{ij}^Q = S/N$ , where  $Q = E, I$ ,  $S$  is the coupling strength, and  $N$  is the total number of neurons in the network. Note that our algorithm can be easily extended to networks with more complicated connectivity structure.

## 2.2. Runge-Kutta Method

Without loss of generality, we consider the RK2 method as the benchmark and compare it with the ETD methods. We first introduce the RK2 method to evolve the HH neural network with a fixed time step  $\Delta t$ , for example, to evolve the system from time  $t = t_k = k\Delta t$  to  $t = t_{k+1} = (k+1)\Delta t$ . Since the synaptic spike times in  $[t_k, t_{k+1}]$  can not be predetermined, one has to evolve the network without considering synaptic spike interactions and reconsider their effects by using spike-spike interactions at the end of time step (Hansel et al., 1998; Brette et al., 2007).

Due to the pulse-coupled dynamics in Equation (5), the numerical accuracy may be very low if the spike timing is not well estimated. For example, suppose that a presynaptic spike fired at  $\tilde{t}$  between  $t_k$  and  $t_{k+1}$ . If one simply assigns it to be the end of time step  $t_{k+1}$ , then the error of the spike-induced conductance change is

$$\frac{S}{N} [\mathbb{G}(\sigma_d^Q, \sigma_r^Q, t - \tilde{t}) - \mathbb{G}(\sigma_d^Q, \sigma_r^Q, t - t_{k+1})] = O(t_{k+1} - \tilde{t}) \quad (7)$$

$$= O(\Delta t), Q = E, I.$$

Therefore, the error with the magnitude of  $\Delta t$  will be introduced when the system evolves to  $t = t_{k+1}$ .

We now solve the above issue arising from the pulse-coupled dynamics to achieve a second-order numerical accuracy. First, we evolve the HH neural network without considering the feedforward and synaptic spikes during the time interval  $[t_k, t_{k+1}]$ . Then, at time  $t = t_{k+1}$ , some neuron's voltage may be above the threshold, i.e., generating a spike, say neuron  $i$ , if  $V_{i,k} < V^{\text{th}}$  and  $V_{i,k+1} \geq V^{\text{th}}$  where  $V_{i,k}$  and  $V_{i,k+1}$  represent  $V_i(t_k)$  and  $V_i(t_{k+1})$ , respectively. The spike time, say  $\tau_{il}$ , can be estimated following the idea proposed in Hansel et al. (1998) and Shelley and Tao (2001). The neuron's membrane potential during the time interval can be approximated by a linear interpolation:

$$V_i(t) \approx V_{i,k} + \frac{V_{i,k+1} - V_{i,k}}{\Delta t} (t - t_k), \quad (8)$$

and the spike time  $\tau_{il}$  can be estimated by solving the equation:

$$V^{\text{th}} = V_{i,k} + \frac{V_{i,k+1} - V_{i,k}}{\Delta t} (\tau_{il} - t_k). \quad (9)$$

Since there may be some neurons firing and some feedforward spikes emitting during the time interval and they will induce the conductance change, the conductance should be then recalibrated. When the neuron firing and the feedforward spikes are not considered, the conductance variables in such cases, denoted by  $\tilde{G}^Q$  and  $\tilde{H}^Q$ ,  $Q = E, I$ , are

$$\tilde{H}_{j,k+1}^Q = H_{j,k}^Q e^{-\Delta t / \sigma_d^Q}, \quad (10)$$

$$\tilde{G}_{j,k+1}^Q = G_{j,k}^Q e^{-\Delta t / \sigma_r^Q} + H_{i,k}^Q \mathbb{G}(\sigma_d^Q, \sigma_r^Q, \Delta t), \quad (11)$$

and the conductance variables are then recalibrated by taking into account the neuron firing and the feedforward spikes as

$$H_{j,k+1}^Q = \tilde{H}_{j,k+1}^Q + F^Q \sum_{t_k < s_{jl} \leq t_{k+1}} e^{-(t_{k+1} - s_{jl}) / \sigma_d^Q}$$

$$+ \frac{S}{N} \sum_i \sum_{t_k < \tau_{il} \leq t_{k+1}} e^{-(t_{k+1} - \tau_{il}) / \sigma_d^Q}, \quad (12)$$

$$G_{j,k+1}^Q = \tilde{G}_{j,k+1}^Q + F^Q \sum_{t_k < s_{jl} \leq t_{k+1}} \mathbb{G}(\sigma_d^Q, \sigma_r^Q, t_{k+1} - s_{jl})$$

$$+ \frac{S}{N} \sum_i \sum_{t_k < \tau_{il} \leq t_{k+1}} \mathbb{G}(\sigma_d^Q, \sigma_r^Q, t_{k+1} - \tau_{il}), \quad (13)$$

for  $j = 1, 2, \dots, N$ . A detailed algorithm of the RK2 method is given in Algorithm 1.

---

### Algorithm 1: RK2 algorithm

---

**Input:** an initial time  $t_k$  and feedforward input times  $\{s_{jl}\}$

**Output:** Solutions at time  $t_{k+1}$

- 1 **for**  $i = 1$  to  $N$  **do**
  - 2     Solve the HH equations for the  $i$ th neuron without considering spike input using RK2 scheme.
  - 3     **if**  $V_i(t_k) < V^{\text{th}}$  and  $V_i(t_{k+1}) \geq V^{\text{th}}$  **then**
  - 4         The  $i$ th neuron spiked in  $[t_k, t_{k+1}]$ .
  - 5         Estimate the spike time  $\tau_{il}$  by Equation (9).
  - 6     **end**
  - 7 **end**
  - 8 Recalibrate the conductance by Equations (12) and (13).
- 

We show that the above algorithm can indeed achieve a second-order numerical accuracy as follows. If there are no feedforward or synaptic spikes, then all the dependent variables are infinitely differentiable and the RK2 method can achieve an error of order  $O(\Delta t^2)$ . For the time step that contains feedforward or synaptic spikes, an error of order  $O(\Delta t)$  is introduced in the conductance with the form of  $G^Q - \tilde{G}^Q$ ,  $Q = E, I$ . Nevertheless, the dependent variables of  $V, m, h$ , and  $n$  can have an error of order  $O(\Delta t^2)$ . The synaptic spike times are

estimated by a linear interpolation and also have an error of order  $O(\Delta t^2)$ . After recalibration shown in Equations (12) and (13), the conductance variables  $G^Q$  and  $H^Q$ ,  $Q = E, I$  can achieve numerical accuracy of second-order at the end of the time step. Therefore, all the dependent variables  $V, m, h, n, G^E, G^I, H^E$ , and  $H^I$  that are numerically solved in each time step have an error of order  $O(\Delta t^2)$  (see below for verification of numerical results).

### 2.3. Exponential Time Differencing Method

Exponential time differencing method is proposed to solve the stiff problem in differential equations by decomposing the system into a linear stiff term and a nonlinear non-stiff term (Hochbruck et al., 1998; Cox and Matthews, 2002; Kassam and Trefethen, 2005; Nie et al., 2008). Following this idea, we propose the ETD schemes for HH Equations (1) and (2) below. As illustrated in Algorithm 1, each neuron in the HH network is evolved independently and their conductances are recalibrated at the end of time step. Thus, one can first derive an ETD scheme for a single HH neuron and then consider the spike interactions among neurons, and obtain an ETD scheme for the numerical evolution of an HH neural network.

Consider the evolution of a single HH neuron from  $t_k$  to  $t_{k+1}$ . The first step of the ETD method is to rewrite Equations (1) and (2) as

$$\frac{dz}{dt} = c_z z + F_z, \quad \text{for } z = V, m, h, n, \quad (14)$$

where

$$c_V = (-G_{Na} m_k^3 h_k - G_K n_k^4 - G_L) / C, \quad (15)$$

$$c_z = -\alpha_z(V_k) - \beta_z(V_k), \quad \text{for } z = m, h, n, \quad (16)$$

$$F_V(t, V, m, h, n) = [- (V - V_{Na}) G_{Na} m^3 h - (V - V_K) G_K n^4 - (V - V_L) G_L + I^{\text{input}}] / C - c_V V \quad (17)$$

and

$$F_z(t, V, m, h, n) = (1 - z) \alpha_z(V) - z \beta_z(V) - c_z z, \quad \text{for } z = m, h, n, \quad (18)$$

where  $z_k$  represents  $z(t_k)$  for  $z = V, m, h, n$  of this neuron. Here,  $F_z(t, V, m, h, n)$  is actually a function of  $t, V$ , and  $z$  for  $z = m, h, n$ , but we write in this way for ease of illustration. Note that the linear coefficient  $c_z$  in Equation (14) is a constant value in the  $k$ th time step  $[t_k, t_{k+1}]$  and is updated with respect to  $k$ . Multiplying Equation (14) by an integrating factor  $e^{-c_z t}$  and taking integral from  $t_k$  to  $t_{k+1}$ , we obtain

$$z_{k+1} = z_k e^{c_z \Delta t} + e^{c_z \Delta t} \int_0^{\Delta t} e^{-c_z \tau} F_z(t_k + \tau, V(t_k + \tau), m(t_k + \tau), h(t_k + \tau), n(t_k + \tau)) d\tau \quad (19)$$

for  $z = V, m, h$ , and  $n$ .

The next step of the ETD method is to derive proper approximations to the above integration. We take a second-order

ETD formula with RK time stepping which was proposed as ETD2RK method in Cox and Matthews (2002). Let,

$$a_{z,k} = z_k e^{c_z \Delta t} + F_{z,k} (e^{c_z \Delta t} - 1) / c_z, \quad (20)$$

and approximate  $F_z$  during the time interval  $[t_k, t_{k+1}]$  by

$$F_z(t_k + \tau, V(t_k + \tau), m(t_k + \tau), h(t_k + \tau), n(t_k + \tau)) = F_{z,k} + \tau (F_z(t_{k+1}, a_{V,k}, a_{m,k}, a_{h,k}, a_{n,k}) - F_{z,k}) / \Delta t + O(\Delta t^2), \quad (21)$$

for  $z = V, m, h$ , and  $n$ , where  $F_{z,k}$  represents  $F_z(t_k, V_k, m_k, h_k, n_k)$ . Substituting the above approximation into Equation (19) yields the ETD2 scheme (the ETD method which has second-order numerical accuracy) which is given by

$$z_{k+1} = a_{z,k} + [F_z(t_{k+1}, a_{V,k}, a_{m,k}, a_{h,k}, a_{n,k}) - F_{z,k}] (e^{c_z \Delta t} - 1 - c_z \Delta t) / c_z^2 \Delta t, \quad (22)$$

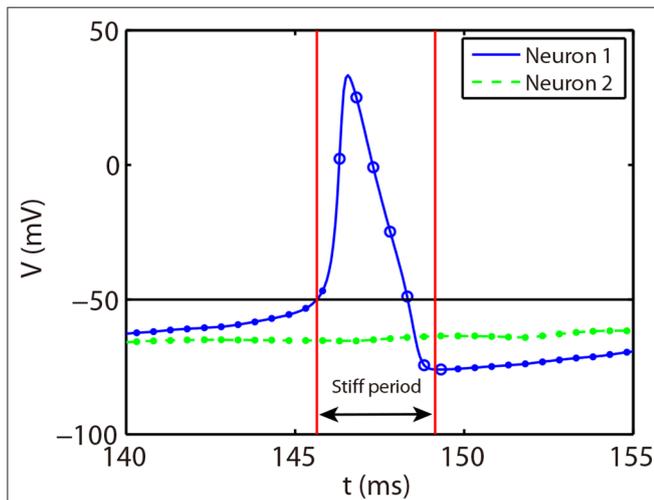
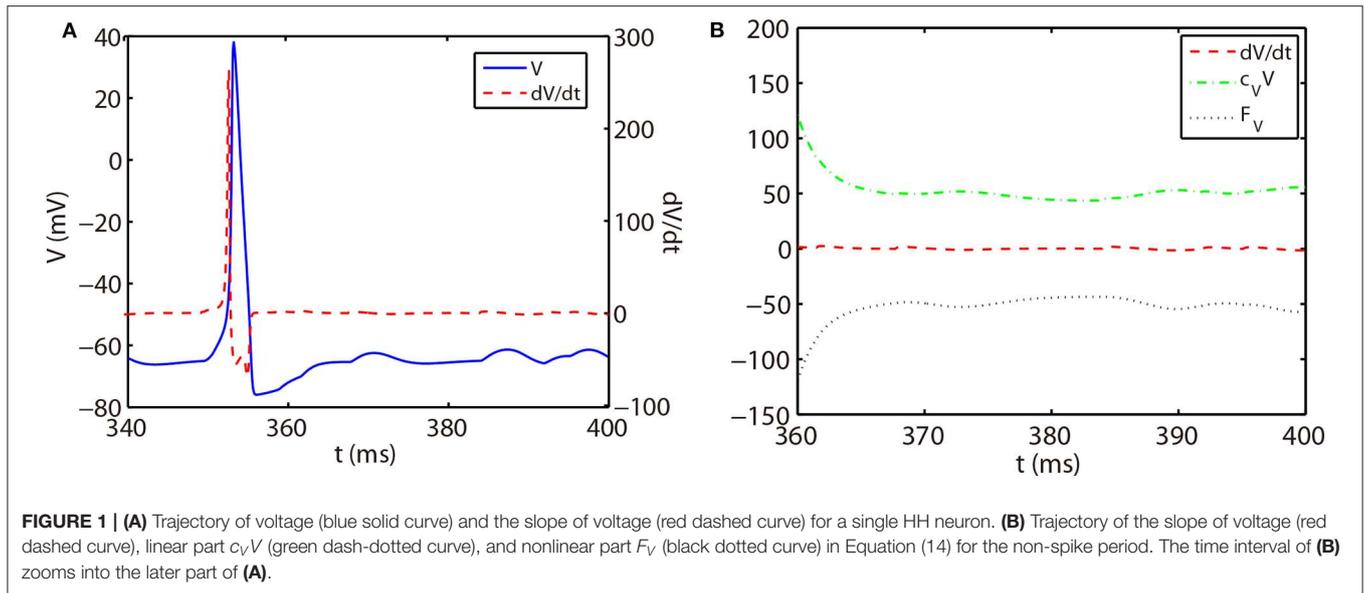
for  $z = V, m, h$ , and  $n$ . The procedure of the ETD2 algorithm for an HH neural network is similar to that of the RK2 algorithm given in Algorithm 1, but the RK2 scheme is replaced by the ETD2 scheme in Equation (22).

### 2.4. Adaptive Exponential Time Differencing Method

The ETD2 method can indeed use a large time step to improve computational efficiency, but we find that it will introduce relatively large error in the trajectories of neurons' membrane potentials and even lead to the missing of action potentials (see below for numerical results). In addition, the number of the missed action potentials in the ETD2 method can grow with the increase of time steps compared with the RK2 method using a small time step. Thus, it is important to design an efficient but also reliable ETD method to solve this issue.

As shown in **Figure 1A**, the slope of voltage has a very large value when the neuron generates an action potential (spike period) and quickly reduces to a value around zero in the non-spike period until the next spike time. Therefore, the stiffness of HH equations is quite different between spike and non-spike periods. In the non-spike period, the slope of voltage is almost zero, while the linear and nonlinear parts in Equation (14) have a much larger absolute value and nearly cancel each other out as shown in **Figure 1B**. Therefore, the decomposition in Equation (14) may not be appropriate in the non-spike period since both the linear and nonlinear parts become stiff while the summation of them is indeed non-stiff. Based on this, we propose a different decomposition in the non-stiff period from that in the stiff period: taking  $c_z = 0$  and the whole right hand side of Equation (14) as the nonlinear part. For such a decomposition, the ETD2 scheme reduces to the RK2 scheme in the non-stiff period.

The stiff period of HH equations can be clearly identified as shown in **Figure 1A** and is defined as follows. For each spike event, the starting point of the stiff period is determined as the spike time when the voltage reaches the firing threshold  $V_{th} = -50$  mV and the interval of stiff period is chosen as 3.5 ms which is sufficient long to cover the highly stiff region of



the spike. Based on the above observation, we give our AETD2 method for HH neural network as following: each neuron is evolved using ETD2 scheme if it is in the stiff period and use the reduced ETD2 scheme, the RK2 scheme, otherwise, as shown in **Figure 2**. Detailed AETD2 algorithm is given in **Algorithm 2**.

### 3. RESULTS

We consider an all-to-all connected network of 80 excitatory and 20 inhibitory neurons driven by Poisson feedforward

#### Algorithm 2: AETD2 algorithm

**Input:** an initial time  $t_k$ , feedforward input times  $\{s_{il}\}$

**Output:** Solutions at time  $t_{k+1}$

```

1 for  $i = 1$  to  $N$  do
2   Solve the HH equations for the  $i$ th neuron without
   considering spike input:
3   if The  $i$ th neuron is inside the stiff period then
4     use ETD2 scheme
5   else
6     use RK2 scheme
7   end
8   if  $V_i(t_k) < V^{th}$  and  $V_i(t_{k+1}) \geq V^{th}$  then
9     The  $i$ th neuron spiked in  $[t_k, t_{k+1}]$ .
10    Estimate the spike time  $\tau_{il}$  by Equation (9).
11  end
12 end
13 Recalibrate the conductance by Equations (12) and (13).

```

input. For the ease of illustration, we choose the Poisson input strength  $f = 0.06 \text{ mS}\cdot\text{cm}^{-2}$  and input rate  $\nu = 300 \text{ Hz}$ , and the coupling strength between neurons are chosen as  $S = 0.2 \text{ mS}\cdot\text{cm}^{-2}$  throughout this work, unless indicated otherwise. However, our algorithm can be applied to HH neural networks under a variety of dynamical regimes.

First, we verify the second-order numerical accuracy by performing convergence tests. A high precision solution is obtained by using RK2 method with a sufficiently small time step  $\Delta t = 1 \times 10^{-6} \text{ ms}$  and is denoted by a superscript “high.” It is compared with the solutions computed by the RK2, ETD2, and AETD2 methods with various values of larger time steps  $\Delta t = 2^{-4}, 2^{-5}, \dots, 2^{-12} \text{ ms}$  which is denoted by a superscript

“ $\Delta t$ .” Errors of membrane potentials at final run time  $T = 2,000$  ms and the last spike time of each neuron are computed:

$$\text{Error}_V = \sqrt{\sum_i (V_i^{(\Delta t)}(T) - V_i^{(\text{high})}(T))^2}, \quad (23)$$

$$\text{Error}_\tau = \sqrt{\sum_i (\tau_{i|*}^{(\Delta t)} - \tau_{i|*}^{(\text{high})})^2}, \quad (24)$$

where  $\tau_{i|*}$  indicates the last spike time of the  $i$ th neuron during the run time interval. As shown in **Figure 3**, if one naively assigns the end of time step as the spike times in the RK2 method (naive RK2), the numerical accuracy of the membrane potentials and spike times can only be of the first-order. In contrast, if one determines the spike times by linear interpolation and recalibrate the conductances accordingly, all the RK2, ETD2, and AETD2 methods can achieve a second-order numerical accuracy. In addition, we find that the ETD2 method has much larger error compared with the RK2 and AETD2 methods using the same time step as shown in **Figure 3**. When using a time step larger than  $\Delta t = 2^{-6} = 0.0156$  ms, the ETD2 method performs even worse than the naive RK2 method. The underlying reason is that the HH equations are almost non-stiff in the non-spike period, but the decomposition in Equation (14) induces a relatively large stiffness for the nonlinear term as discussed previously.

We next discuss the numerical performance of our AETD2 method and compare it with other different numerical methods. As shown in the top panel of **Figure 4**, the AETD2 method with large time steps (maximum time step  $\Delta t = 0.277$  ms) can obtain the same high accuracy in membrane potentials as the RK2 method using a very small time step  $\Delta t = 0.01$  ms. The bottom panel of **Figure 4** shows the raster plots (neuron index vs. its spike time) of the spike events in the network. It can be seen that the spike times are well-captured by the AETD2 method with large time steps. In contrast, as shown in **Figures 5A,B**, the ETD2 method is highly inaccurate in terms of voltage traces and raster plots when the time step  $\Delta t = 0.277$  ms is used (the maximum time step in AETD2 method). **Figure 5C** shows the relative error in the mean firing rate (the average number of synaptic spikes per unit time) between the ETD2 and RK2 methods, and that between the AETD2 and RK2 methods over different values of coupling strength. It can be seen that the ETD2 method can achieve only one digit of numerical accuracy while the AETD2 method can robustly achieve more than two digits of numerical accuracy when the time step  $\Delta t = 0.277$  ms is used in both methods. Therefore, the ETD2 method has worse numerical performance compared with the AETD2 method.

To demonstrate the efficiency of our AETD2 method, we compare the simulation time that RK2, ETD2, and AETD2 methods take for a common total run time. We simulate the all-to-all connected network by the RK2, ETD2, and AETD2 methods on a Windows platform using an Intel i7 2.6 GHz processor (the weblink of the source codes is given in section 4), and the simulation time and numerical accuracy of mean firing rate are given in **Table 1**. The AETD2 method can achieve over an

order of magnitude of speedup compared with the RK2 method while achieving the same high accuracy in terms of the mean firing rate.

In addition, we define the efficiency ratio of the AETD2 method over the RK2 method as

$$E = \frac{T_{\text{RK2}}}{T_{\text{AETD2}}} \quad (25)$$

where  $T_{\text{RK2}}$  and  $T_{\text{AETD2}}$  indicate the simulation times of the RK2 and AETD2 methods, respectively, for the HH neural network to evolve the run time  $T$ . Note that the RK2 and ETD2 methods take almost the same simulation time when using the same small time step as shown in **Table 1**. Thus, the above efficiency ratio can be approximated by the ratio of the total number of time steps each method requires as

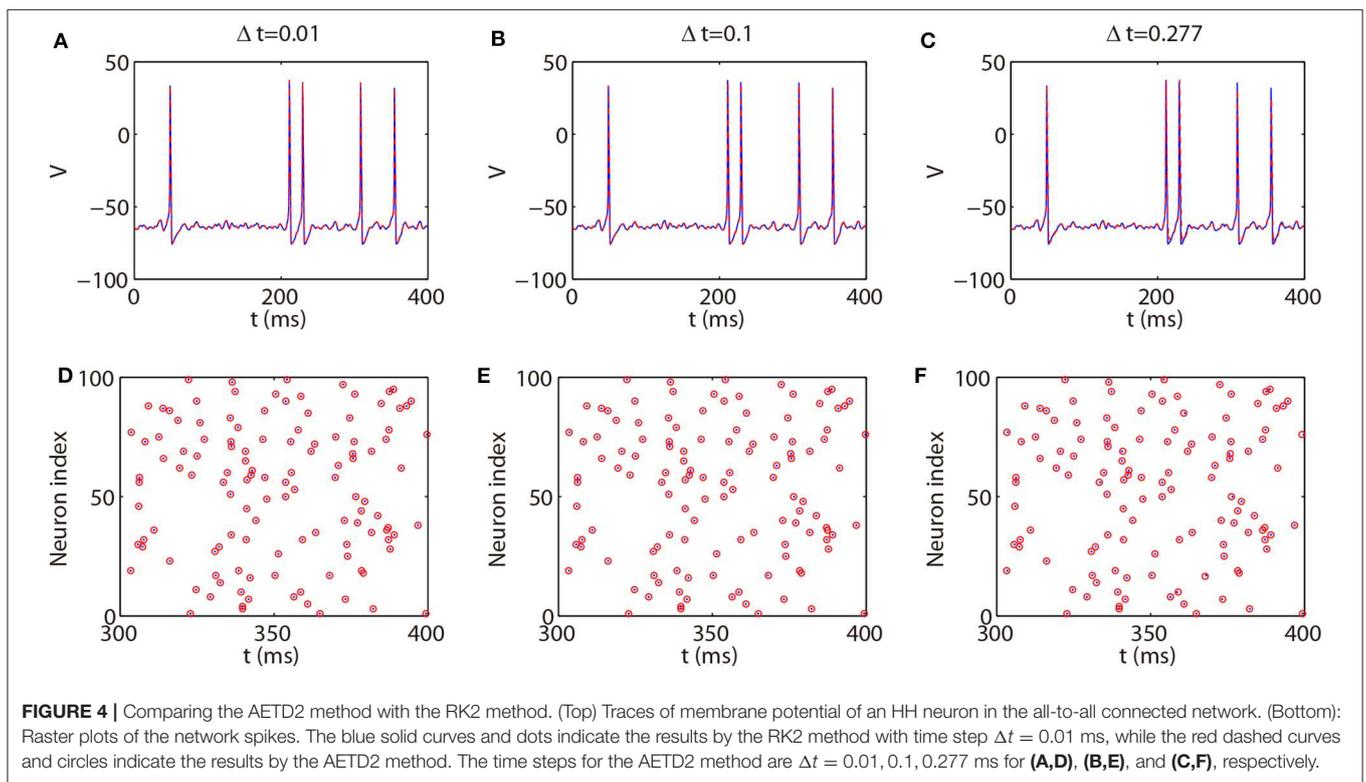
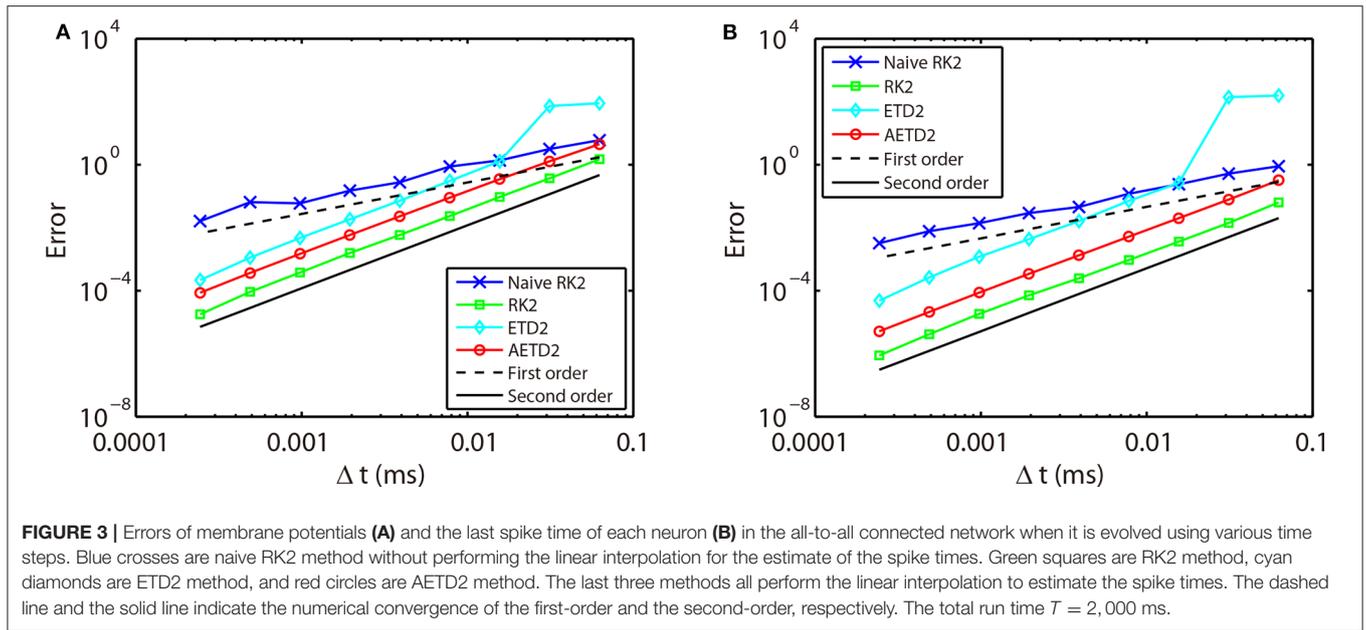
$$E \approx \frac{T/\Delta t_{\text{RK2}}}{T/\Delta t_{\text{AETD2}}} = \frac{\Delta t_{\text{AETD2}}}{\Delta t_{\text{RK2}}}, \quad (26)$$

where  $\Delta t_{\text{RK2}}$  and  $\Delta t_{\text{AETD2}}$  indicate the time steps used in the RK2 and AETD2 methods, respectively. To demonstrate that the above efficiency ratio is independent of the network connectivity, size, and dynamical regimes, we evolve the all-to-all connected network of 80 excitatory and 20 inhibitory neurons and a randomly connected network of 800 excitatory and 200 inhibitory neurons with a variety choice of coupling strength. Not surprisingly, the efficiency ratio approximated by Equation (26) agrees well with the one measured by the ratio of simulation time between the RK2 and AETD2 methods in both two networks as shown in **Figure 6**. Hence, the efficiency ratio of the AETD2 method relies on only the size of evolved time steps.

## 4. DISCUSSION

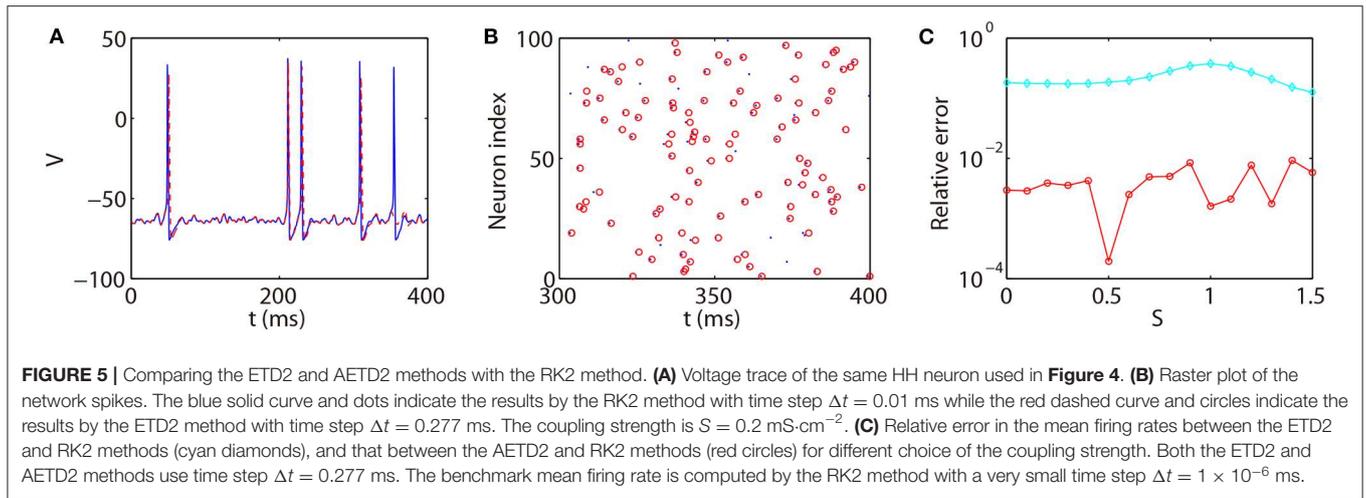
We have presented an adaptive second-order ETD method to evolve the pulse-coupled HH neural network. Our AETD2 method can solve the stiff problem in the HH equations when an HH neuron generates an action potential (spike period). It can use a large time step to raise computational efficiency while accurately capturing dynamical properties of HH neurons such as the trace of membrane potentials, spike times of each neuron, and the mean firing rate. We point out that our AETD2 method can robustly enlarge time steps and raise computational efficiency over one order of magnitude compared with the RK2 method. This high efficiency seems to be independent of parameter choice of connectivity structure, dynamical regimes, or network size.

Our adaptive ideas of ETD methods can be applied to dynamical systems with stiff and non-stiff periods. In addition, we point out that the ETD scheme in our AETD2 algorithm can be chosen in a variety of forms according to the properties of dynamical systems. Here, we use the ETD2 scheme derived by approximating the integration in Equation (19) with RK time stepping. Other forms of numerical schemes can also be used to approximate the integration. For example, one can use a linear interpolation to approximate the nonlinear part in Equation (14) to obtain another form of ETD2 scheme. Besides, one can derive an ETD scheme following the idea proposed in Börgers and



Nectow (2013) by linearly approximating the HH equations. The derived ETD scheme is proven to be unconditionally stable for HH system in Börgers and Nectow (2013). All these different ETD schemes can be easily embedded into our AETD2 method in the same way as given in Algorithm 2. For example, we can embed the ETD formula proposed in Börgers and Nectow (2013) into the AETD2 method to evolve the reduced Traub

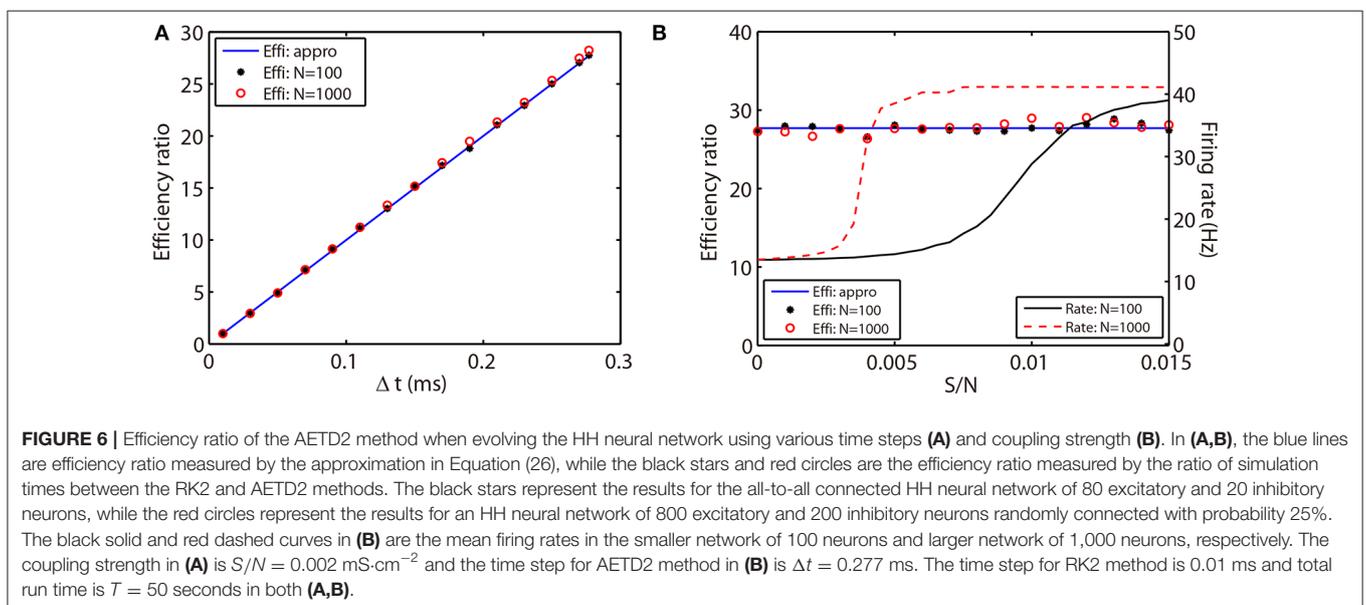
Miles (RTM) neural networks (Ermentrout and Kopell, 1998; Olufsen et al., 2003; Börgers and Nectow, 2013). The dynamical equations for an RTM neuron is almost the same as that for an HH neuron except that the gating variable  $m$  is described by  $m_i = \alpha_m(V_i)/(\alpha_m(V_i) + \beta_m(V_i))$ . The forms of  $\alpha$  and  $\beta$  for the RTM neurons are set as:  $\alpha_m(V_i) = 0.32(V_i + 54)/(1 - \exp(-(V_i + 54)/4))$ ,  $\beta_m(V_i) = 0.28(V_i + 27)/(\exp((V_i + 27)/5) - 1)$ ,  $\alpha_h(V_i) =$

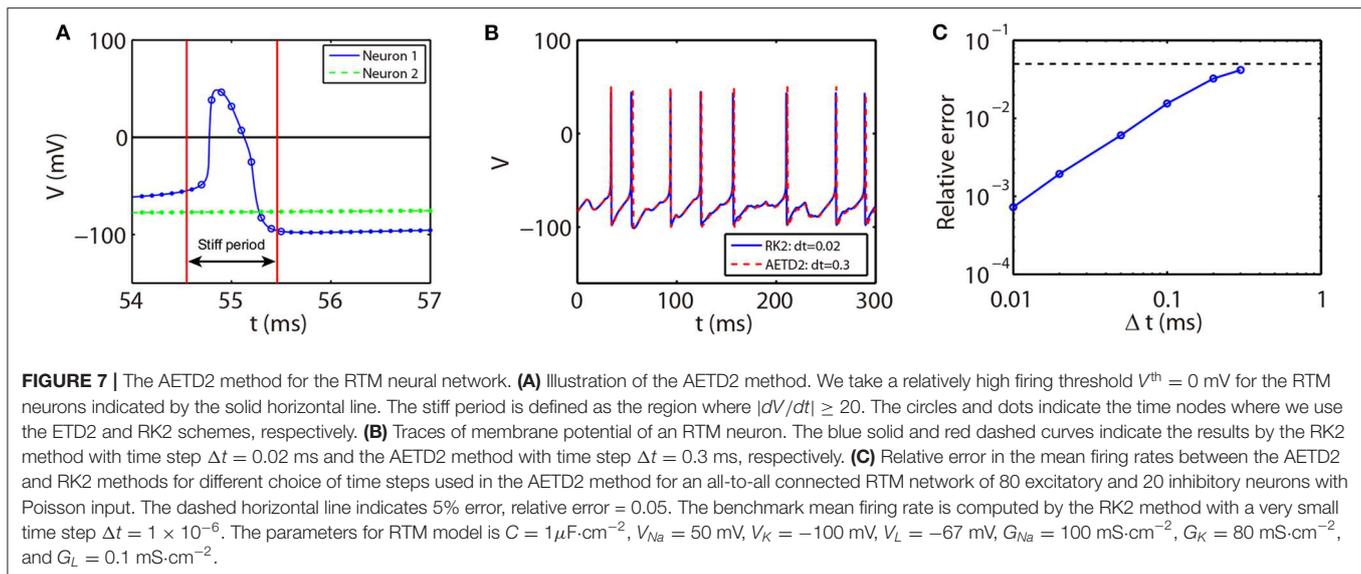


**TABLE 1 |** Simulation of the all-to-all connected network with a total run time  $T = 10$  s.

$\Delta t$ (ms)	RK2			ETD2			AETD2		
	CPU	Relative error		CPU	Relative error		CPU	Relative error	
0.005	60.56 s	0	(13.61 Hz)	60.07 s	0	(13.61 Hz)	60.82 s	0	(13.61 Hz)
0.01	30.22 s	0	(13.61 Hz)	30.05 s	0	(13.61 Hz)	30.55 s	0	(13.61 Hz)
0.02	14.99 s	0	(13.61 Hz)	15.03 s	0.074%	(13.60 Hz)	15.30 s	0	(13.61 Hz)
0.05	***	***	***	5.95 s	0.66%	(13.52 Hz)	6.16 s	0.074 %	(13.62 Hz)
0.1	***	***	***	2.94 s	2.65%	(13.25 Hz)	3.11 s	0.074 %	(13.62 Hz)
0.2	***	***	***	1.48 s	9.99%	(12.25 Hz)	1.57 s	0.15 %	(13.63 Hz)
0.277	***	***	***	1.09 s	12.56%	(11.29 Hz)	1.15 s	0.59 %	(13.69 Hz)
0.5	***	***	***	0.57 s	41.59%	(7.95 Hz)	***	***	***
1	***	***	***	0.29 s	87.07%	(1.76 Hz)	***	***	***

The simulation time is measured in seconds. The relative error in the mean firing rate between each method using different time steps and the RK2 method using a very small time step  $\Delta t = 1 \times 10^{-6}$  ms is measured in percentage and the mean firing rate is measured in Hz given inside the parentheses. Asterisks indicate overflow errors.





$0.128 \exp(-(V_i + 50)/18)$ ,  $\beta_h(V_i) = 4/(1 + \exp(-(V_i + 27)/5))$ ,  $\alpha_n(V_i) = 0.032(V_i + 52)/(1 - \exp(-(V_i + 52)/5))$ , and  $\beta_n(V_i) = 0.5 \exp(-(V_i + 57)/40)$ .

Note that the rising phase of action potentials for the RTM neurons is extremely short, around 0.03 ms as shown in **Figure 7A**. In such a situation, it may not be appropriate to choose the spike time (when the voltage reaches the firing threshold) as the starting point of the stiff period in the AETD2 method as shown in **Figure 7A** since large numerical error will be introduced, especially when a large time step is used, e.g., time step  $\Delta t = 0.3$  ms. This is because the system is evolved by the RK2 scheme during the time step that contains the rapid rising region of the neuron's action potential. Therefore, to achieve high numerical accuracy, the interval of the stiff period should cover the rapid rising region. To achieve this, we then define the stiff period of the RTM neurons as the region where the magnitude of the slope of voltage is over a proper threshold as shown in **Figure 7A**. We point out that our AETD2 method using a large time step can still accurately capture the membrane potential traces and the mean firing rates compared with the RK2 method using a small time step as shown in **Figures 7B,C**. Therefore, the definition of the stiff period in our AETD2 method can be flexibly determined based on dynamical properties of studied systems.

In this work, the numerical accuracy of our AETD2 method is second-order. In some situations, high accurate traces of membrane potentials may be required, especially the accurate shapes of action potentials (Traub et al., 2001; Kopell and Ermentrout, 2004). Therefore, one future work may be the design of the fourth-order ETD method. As illustrated above, due to the discontinuity arising from the pulse-coupled dynamics, an even more careful recalibration needs to be designed to achieve fourth-order numerical accuracy.

Finally, we point out that our AETD2 method can be easily extended to networks of other HH type neurons (Pospisichil et al., 2008). And our AETD2 method can also robustly achieve high numerical accuracy and efficiency. In addition, our method is naturally a parallel algorithm which can be applied to simulations of large-scale neural network dynamics. For reproducibility of our results by other researchers, all the source codes written in C++ can be accessed at [http://github.com/KyleZhongqi/ETD2\\_HH](http://github.com/KyleZhongqi/ETD2_HH).

## DATA AVAILABILITY STATEMENT

The datasets generated for this study are available on request to the corresponding author.

## AUTHOR CONTRIBUTIONS

ZT and DZ designed the research, wrote and code, ran the simulations, analyzed the results, and wrote the paper.

## FUNDING

This work was supported by National Science Foundation of China with Grant Nos. 11671259, 11722107, and SJTU-UM Collaborative Research Program, and by the Student Innovation Center at Shanghai Jiao Tong University (DZ).

## ACKNOWLEDGMENTS

This manuscript has been released as a Pre-Print at arXiv repository (Tian and Zhou, 2019). We dedicate this paper to our late professor David Cai.

## REFERENCES

- Aihara, K. (1986). Chaotic oscillations and bifurcations in squid giant axons. *Chaos* 257–269. doi: 10.1515/9781400858156.257
- Börgers, C., Epstein, S., and Kopell, N. J. (2005). Background gamma rhythmicity and attention in cortical local circuits: a computational study. *Proc. Natl. Acad. Sci. U.S.A.* 102, 7002–7007. doi: 10.1073/pnas.0502366102
- Börgers, C., and Nectow, A. R. (2013). Exponential time differencing for Hodgkin-Huxley-like odes. *SIAM J. Sci. Comput.* 35, B623–B643. doi: 10.1137/120883657
- Brette, R., Rudolph, M., Carnevale, T., Hines, M., Beeman, D., Bower, J. M., et al. (2007). Simulation of networks of spiking neurons: a review of tools and strategies. *J. Comput. Neurosci.* 23, 349–398. doi: 10.1007/s10827-007-0038-6
- Cox, S. M., and Matthews, P. C. (2002). Exponential time differencing for stiff systems. *J. Comput. Phys.* 176, 430–455. doi: 10.1006/jcph.2002.6995
- Dayan, P., and Abbott, L. (2003). Theoretical neuroscience: computational and mathematical modeling of neural systems. *J. Cogn. Neurosci.* 15, 154–155. doi: 10.1162/089892903321107891
- Dayan, P., and Abbott, L. F. (2001). *Theoretical Neuroscience, Vol. 806*. Cambridge, MA: MIT Press.
- de la Hoz, F., and Vadillo, F. (2008). An exponential time differencing method for the nonlinear schrödinger equation. *Comput. Phys. Commun.* 179, 449–456. doi: 10.1016/j.cpc.2008.04.013
- Ermentrout, G. B., and Kopell, N. (1998). Fine structure of neural spiking and synchronization in the presence of conduction delays. *Proc. Natl. Acad. Sci. U.S.A.* 95, 1259–1264. doi: 10.1073/pnas.95.3.1259
- Guckenheimer, J., and Oliva, R. A. (2002). Chaos in the Hodgkin-Huxley model. *SIAM J. Appl. Dyn. Syst.* 1, 105–114. doi: 10.1137/S1111111101394040
- Hansel, D., Mato, G., Meunier, C., and Neltner, L. (1998). On numerical simulations of integrate-and-fire neural networks. *Neural Comput.* 10, 467–483. doi: 10.1162/089976698300017845
- Hansel, D., and Sompolinsky, H. (1996). Chaos and synchrony in a model of a hypercolumn in visual cortex. *J. Comput. Neurosci.* 3, 7–34. doi: 10.1007/BF00158335
- Hassard, B. (1978). Bifurcation of periodic solutions of the Hodgkin-Huxley model for the squid giant axon. *J. Theor. Biol.* 71, 401–420. doi: 10.1016/0022-5193(78)90168-6
- Hochbruck, M., Lubich, C., and Selhofer, H. (1998). Exponential integrators for large systems of differential equations. *SIAM J. Sci. Comput.* 19, 1552–1574. doi: 10.1137/S1064827595295337
- Hochbruck, M., and Ostermann, A. (2010). Exponential integrators. *Acta Numer.* 19, 209–286. doi: 10.1017/S0962492910000048
- Hodgkin, A. L., and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol.* 117:500. doi: 10.1113/jphysiol.1952.sp004764
- Kassam, A.-K., and Trefethen, L. N. (2005). Fourth-order time-stepping for stiff pdes. *SIAM J. Sci. Comput.* 26, 1214–1233. doi: 10.1137/S1064827502410633
- Kopell, N., and Ermentrout, B. (2004). Chemical and electrical synapses perform complementary roles in the synchronization of interneuronal networks. *Proc. Natl. Acad. Sci. U.S.A.* 101, 15482–15487. doi: 10.1073/pnas.0406343101
- Lin, K. K. (2006). Entrainment and chaos in a pulse-driven Hodgkin-Huxley oscillator. *SIAM J. Appl. Dyn. Syst.* 5, 179–204. doi: 10.1137/050631677
- Nie, Q., Wan, F. Y., Zhang, Y.-T., and Liu, X.-F. (2008). Compact integration factor methods in high spatial dimensions. *J. Comput. Phys.* 227, 5238–5255. doi: 10.1016/j.jcp.2008.01.050
- Olufsen, M. S., Whittington, M. A., Camperi, M., and Kopell, N. (2003). New roles for the gamma rhythm: population tuning and preprocessing for the beta rhythm. *J. Comput. Neurosci.* 14, 33–54. doi: 10.1023/A:1021124317706
- Pospischil, M., Toledo-Rodriguez, M., Monier, C., Piwkowska, Z., Bal, T., Frégnac, Y., et al. (2008). Minimal Hodgkin-Huxley type models for different classes of cortical and thalamic neurons. *Biol. Cybern.* 99, 427–441. doi: 10.1007/s00422-008-0263-8
- Shelley, M. J., and Tao, L. (2001). Efficient and accurate time-stepping schemes for integrate-and-fire neuronal networks. *J. Comput. Neurosci.* 11, 111–119. doi: 10.1023/A:1012885314187
- Somers, D. C., Nelson, S. B., and Sur, M. (1995). An emergent model of orientation selectivity in cat visual cortical simple cells. *J. Neurosci.* 15, 5448–5465. doi: 10.1523/JNEUROSCI.15-08-05448.1995
- Sun, Y., Zhou, D., Rangan, A. V., and Cai, D. (2009). Library-based numerical reduction of the Hodgkin-Huxley neuron for network simulation. *J. Comput. Neurosci.* 27, 369–390. doi: 10.1007/s10827-009-0151-9
- Tian, Z.-Q. K., and Zhou, D. (2019). Exponential time differencing algorithm for pulse-coupled Hodgkin-Huxley neuronal networks. *arXiv [preprint] arXiv:1910.08724*.
- Traub, R. D., Kopell, N., Bibbig, A., Buhl, E. H., LeBeau, F. E., and Whittington, M. A. (2001). Gap junctions between interneuron dendrites can enhance synchrony of gamma oscillations in distributed networks. *J. Neurosci.* 21, 9478–9486. doi: 10.1523/JNEUROSCI.21-23-09478.2001

**Conflict of Interest:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Tian and Zhou. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.