J Comput Neurosci (2009) 27:369–390
DOI 10.1007/s10827-009-0151-9

Library-based numerical reduction of the Hodgkin–Huxley neuron for network simulation

Yi Sun · Douglas Zhou · Aaditya V. Rangan · David Cai

Received: 8 August 2008 / Revised: 13 February 2009 / Accepted: 18 March 2009 / Published online: 29 April 2009

Abstract We present an efficient library-based numerical method for simulating the Hodgkin–Huxley (HH) neuronal networks. The key components in our numerical method involve (i) a pre-computed high resolution data library which contains typical neuronal trajectories (i.e., the time-courses of membrane potential and gating variables) during the interval of an action potential (spike), thus allowing us to avoid resolving the spikes in detail and to use large numerical time steps for evolving the HH neuron equations; (ii) an algorithm of spike-spike corrections within the groups of strongly coupled neurons to account for spike-spike interactions in a single large time step. By using the library method, we can evolve the HH networks using time steps one order of magnitude larger than the typical time steps used for resolving the trajectories without the library, while achieving comparable resolution in statistical quantifications of the network activity, such as average firing rate, interspike interval distribution, power spectra of voltage traces. Moreover, our large time steps using the library method can break the stability requirement of standard methods (such as Runge–Kutta (RK) methods) for the original dynamics. We compare our library-based method with RK methods, and find that our method can capture very well phase-locked, synchronous, and chaotic dynamics of HH neuronal networks. It is important to point out that, in essence, our library-based HH neuron solver can be viewed as a numerical reduction of the HH neuron to an integrate-and-fire (I&F) neuronal representation that does not sacrifice the gating dynamics (as normally done in the analytical reduction to an I&F neuron).

Keywords Numerical algorithm · Hodgkin–Huxley neuron · Integrate-and-fire neurons · Neuronal network · Library method · Chaos

Action Editor: David Golomb

Y. Sun (✉) · D. Zhou · A. V. Rangan · D. Cai
Courant Institute of Mathematical Sciences,
New York University, New York, NY 10012, USA
e-mail: yisun@cims.nyu.edu

1 Introduction

Systems of conductance-based integrate-and-fire (I&F) neurons have been used to simulate the dynamics and study the properties of large scale neuronal networks (Somers et al. 1995; Troyer et al. 1998; McLaughlin et al. 2000; Cai et al. 2005; Rangan et al. 2005; Rangan and Cai 2007). But the I&F model does not account for the detailed generation of action potentials. We consider here the more realistic Hodgkin–Huxley (HH) model (Hodgkin and Huxley 1952; Dayan and Abbott 2001). This model of excitable membrane, originally introduced to describe the behavior of the squid's giant axon, provides a useful mechanism that accounts naturally for both the generation of spikes, due to voltage-dependent membrane conductances arising from ionic channels, and the existence of absolute refractory periods. This classical model serves as the foundation for other neuron models with more complicated behaviors, such as bursting. However, the complexity of the HH-like neuron model precludes detailed analytical studies of its properties, hence we often resort to numerical simulations to study them.

In this paper, we focus on a library-based numerical method for simulating the HH neuronal networks. In general, we cannot take large time steps to solve the HH neuron equations since they are stiff when the neuron is spiking. But in network simulations we frequently need to investigate the system's behavior for many different sets of parameters or to perform a statistical analysis over many trial conditions. It is therefore important to integrate the dynamics with an efficient algorithm that allows us to use as large a time step as possible.

Inspired by the simplicity of the I&F model, we propose a specialized numerical method which reduces the dynamics of the HH neuron model to an I&F-like model. This method, referred to as the "library method", can overcome the time step limitation due to the stiffness of the HH neuron model. In the temporal evolution of the dynamic variables of the HH neuron model during a single action potential (which onsets at the threshold, say, −50 mV), a large influx of the sodium ions produce a sharp spike of inward current which causes the membrane potential to rise very rapidly. During this brief period, the inward current is so large that the HH dynamics is very stiff, for which we need a sufficiently small time step to resolve the dynamics numerically. After the membrane potential reaches the peak value (near the sodium ions equilibrium potential, 50 mV), the potassium current drives it back down to negative values rapidly. The time interval of an action potential can last about 3 ms until the membrane potential drops down around its minimum value. The neuron cannot fire again during this period, i.e., in the absolute refractory period, which is explicitly imposed in the I&F model. Based on this observation, we take the following strategy. Once the membrane potential reaches the threshold value, we stop evolving the HH neuron model and restart the integration after the refractory period with the reset values of the membrane potential and other gating variables. However, unlike in the I&F model for which the membrane potential is fixed to be the reset value during the refractory period, here in our method we can recover the time-courses of the membrane potential (as well as other dynamic gating variables) from a precomputed high resolution data library. By this means we can circumvent numerical issues associated with stiffness, and use reasonably large time steps to evolve the HH neuron model.

We emphasize that this numerical I&F reduction of the HH neuron using our library method involves approximations in the spike dynamics and it is not expected to achieve the exact same trajectory-wise solution of the whole network evolved using the regular method (say, Range-Kutta methods) with very small time steps. Hence, our goal is to obtain accurate statistical information of the network—such as average firing rate and spike time distributions with large time steps, rather than accurate individual intracellular membrane potential time-courses for each and every neuron in the system over a long time. However, it should be stressed that our numerical I&F reduction of the HH neuron captures far more detailed dynamics (such as gating variable dynamics), and has a far wider regime of validity than the usual analytic I&F since it is a dynamical reduction adapted to all possible regimes of the HH neuron dynamics, as can be seen below.

As an I&F reduction, we need to address issues arising from large time steps. In particular, we need to take into account the causality of spiking events within a single large time step via spike-spike interactions. Usually, in the modified Runge–Kutta methods (Hansel et al. 1998; Shelley and Tao 2001), when one evolves the trajectory of each neuron in the network for a single time step, only the spikes of the feedforward input to the network within that time step can be used to evolve the system since there is no information about the synaptic spikes during this time interval (i.e., which neurons may fire and when to fire). We can only wait to the end of the step to consider the effect of all the synaptic spikes if any happened during this interval. This approach may become insufficient because the first few of these synaptic spikes induced within a large time step may substantially affect the rest of the network via spike-spike interactions in such a way that the rest of the numerically computed synaptic spikes within the time step are spurious. As a consequence, when used to evolve a network with strong coupling strengths, the modified Runge–Kutta methods with interpolated spikes need to take sufficiently small time steps to have only $O(1)$ number of spikes in the entire system within a single time step. For instance, in cortical networks, in a time interval of 0.1 ms a neuron would receive 1–5 synaptic inputs. To use a large time step, we need to further apply the algorithm of spike-spike corrections within our method for network simulations of HH neurons. This spike-spike correction is achieved by sorting the approximated spike times within each time step and applying an iterated correction procedure to account for the effects of each spike on all future spikes within the time step. This procedure, introduced in Rangan and Cai (2007), is similar to other event-driven approaches (Mattia and Del Giudice 2000; Reutimann et al. 2003; Rudolph and Destexhe 2007) and has been successfully incorporated into a method for simulating

large scale I&F networks (Cai et al. 2005; Rangan et al. 2005).

Extensive efforts have been made to study the impact of chaos not only on other biological and physical systems (Campbell and Rose 1983), but also on neuronal systems (Mainen and Sejnowski 1995; Hansel and Sompolinsky 1992, 1996; Kosmidis and Pakdaman 2003). Chaotic solutions have been observed in the study of a single HH neuron with several different types of external inputs (Aihara and Matsumoto 1986; Guckenheimer and Oliva 2002; Lin 2006). Here we study the issue in homogeneously all-to-all coupled HH neuronal networks under a sinusoidal external drive. We find three typical dynamical regimes as the synaptic coupling strength varies. When the HH neurons are weakly coupled, the network is in a phase-locked state, where each neuron fires repetitively with a train of regularly spaced spikes locked to the feedforward external input. When the coupling is relatively strong, the network operates in a synchronous state. For a moderately strong coupling between these two limits, the network dynamics exhibits chaotic behavior, as quantified by a positive largest Lyapunov exponent. It turns out that there is a strong consequence of these dynamical regimes on our numerical methods for evolving the network dynamics. In the nonchaotic dynamical regimes, i.e., the weak coupling or strong coupling limit, we show that there is a good numerical convergence of the solution in the trajectory-wise sense by using either the regular or library methods. For the chaotic dynamical regime, i.e., an intermediate strong coupling, our study shows that there is no numerical convergence of the solution and only statistical quantifications of the numerical results are reliable. To characterize the chaotic/nonchaotic regimes, we employ several measures, such as the largest Lyapunov exponent, the power spectrum analysis of voltage traces, the return maps of projected spike times and the interspike interval (ISI) distributions. We demonstrate that, in these quantifications, the numerical results using the library method are consistent with those obtained by the standard method. These results confirm that our library-based method can capture very well the HH neuronal network dynamics with good statistical accuracy.

The outline of the paper is as follows. We give a brief description of the HH neuronal network model in the following section. In Section 3, the general numerical methods for single neurons and networks are described. In Section 4, we provide a detailed description of the library method, especially about how to build and use the data library of the membrane potential and gating variables. In Section 5, we compare numerical results for both the standard and library methods, which illustrate the advantage of our method. We conclude in Section 6.

2 The model

2.1 Single neurons

The dynamics of a Hodgkin–Huxley (HH) neuronal network with N neurons is governed by

$$C\frac{d}{dt}V_i = -G_{\mathrm{Na}}m^3h(V_i - V_{\mathrm{Na}}) - G_{\mathrm{K}}n^4(V_i - V_{\mathrm{K}}) - G_{\mathrm{L}}(V_i - V_{\mathrm{L}}) + I_i^{\mathrm{input}}, \tag{1}$$

$$\frac{dm}{dt} = \alpha_m(V_i)(1-m) - \beta_m(V_i)m, \tag{2}$$

$$\frac{dh}{dt} = \alpha_h(V_i)(1-h) - \beta_h(V_i)h, \tag{3}$$

$$\frac{dn}{dt} = \alpha_n(V_i)(1-n) - \beta_n(V_i)n, \tag{4}$$

where the index i labels the neuron in the network, C is the cell membrane capacitance and V_i is its membrane potential, m and h are the activation and inactivation variables of the sodium current, respectively, and, n is the activation variable of the potassium current (Hodgkin and Huxley 1952; Dayan and Abbott 2001). In our network, for simplicity, all neurons are identical. The parameters G_{Na}, G_{K}, and G_{L} are the maximum conductances for the sodium, potassium, and leak currents, respectively, V_{Na}, V_{K}, and V_{L} are the corresponding reversal potentials. Functional forms and parameters values for the HH neuron equations are given in Appendix.

In our network model, I_i^{input} stands for the input current, which is given by $I_i^{\mathrm{input}} = I_i^{\mathrm{F}} + I_i^{\mathrm{S}}$ with

$$\begin{aligned} I_i^{\mathrm{F}} &= -G_i^{\mathrm{F}}(t)\big(V_i(t) - V_{\mathrm{G}}\big), \\ I_i^{\mathrm{S}} &= -G_i^{\mathrm{S}}(t)\big(V_i(t) - V_{\mathrm{G}}\big). \end{aligned} \tag{5}$$

It contains two parts: the current I_i^{F} generated by the feedforward input and the synaptic current I_i^{S} generated from the synaptic interactions of the neurons within the network. Both of these currents share the same reversal potential V_{G} (see Appendix) for excitatory synapses. The terms $G_i^{\mathrm{F}}(t)$ and $G_i^{\mathrm{S}}(t)$ denote their corresponding conductances which are defined below.

2.2 Synaptic interactions

Our network model is conductance-based and synaptic inputs are modeled as conductance changes. We say an action potential or emission of a spike occurs at time t if the membrane potential of a presynaptic neuron (say the jth neuron) reaches a threshold value V^{th} at that time. Then the spike triggers postsynaptic events in all the neurons the jth neuron is connected to and changes their conductances. On the other hand, for the postsynaptic ith neuron, its conductance $G_i^{\text{S}}(t)$ is changed by the sum of the total contributions at time t of all spikes in the past generated by the presynaptic neurons. The dynamics of $G_i^{\text{S}}(t)$ is governed by

$$\frac{d}{dt}G_i^{\text{S}}(t) = -\frac{G_i^{\text{S}}(t)}{\sigma_{\text{r}}} + \widetilde{G}_i^{\text{S}}(t), \tag{6}$$

$$\frac{d}{dt}\widetilde{G}_i^{\text{S}}(t) = -\frac{\widetilde{G}_i^{\text{S}}(t)}{\sigma_{\text{d}}} + \sum_{j\neq i}\sum_k S_{i,j}\delta\left(t - T_{j,k}^{\text{S}}\right), \tag{7}$$

where $\widetilde{G}_i^{\text{S}}(t)$ is an additional variable. The variable $G_i^{\text{S}}(t)$ has an impulse response with the form of an α-function with both a fast rise and a slow decay timescale, σ_{r} and σ_{d}, respectively. The time $T_{j,k}^{\text{S}}$ stands for the kth spike of neuron j prior to time t. The coefficient $S_{i,j}$ represents the coupling strengths, which can encode many different types of network architecture. For the sake of simplicity, we consider an all-to-all homogenous excitatory neuronal network, in which $S_{i,j}$ is a constant S/N. However, our method can readily be extended to a heterogeneous network or more complicated cases involving both excitatory and inhibitory neurons. The typical values or ranges of σ_{r}, σ_{d} and S can be found in Appendix.

2.3 Feedforward inputs

The system is driven by feedforward inputs with $G_i^{\text{F}}(t)$ as a continuous sinusoidal function of time t:

$$G_i^{\text{F}}(t) = \frac{1}{2}F_i\left(\sin\left[\left(\omega t + \frac{i}{N}\right)2\pi\right] + 1\right), \tag{8}$$

where ω is the oscillation angular frequency and F_i is the peak amplitude of the feedforward input. In our work, F_i assumes a constant F for all neurons in the homogenous network. Here, there is a phase difference of $2\pi/N$ between the feedforward input to the ith neuron and the feedforward input to the $(i+1)$th neuron. With this continuous feedforward input, our network model is a deterministic dynamical system.

However, in some cases, we need to consider stochastic dynamics (Section 3), where we use a spike train sampled from a Poisson process with rate equal to ω as the feedforward input. Let $T_{i,k}^{\text{F}}$ denote the kth spike from the feedforward input to the ith neuron. Then the dynamics of $G_i^{\text{F}}(t)$ obeys equations similar to Eqs. (6) and (7):

$$\frac{d}{dt}G_i^{\text{F}}(t) = -\frac{G_i^{\text{F}}(t)}{\sigma_{\text{r}}} + \widetilde{G}_i^{\text{F}}(t), \tag{9}$$

$$\frac{d}{dt}\widetilde{G}_i^{\text{F}}(t) = -\frac{\widetilde{G}_i^{\text{F}}(t)}{\sigma_{\text{d}}} + \sum_k F_i\delta\left(t - T_{i,k}^{\text{F}}\right), \tag{10}$$

where we take the same rise and decay timescales σ_{r} and σ_{d} as in Eqs. (6) and (7). Then we can combine both of the conductances into one term $G_i(t) = G_i^{\text{F}}(t) + G_i^{\text{S}}(t)$ and its dynamics is given by

$$\frac{d}{dt}G_i(t) = -\frac{G_i(t)}{\sigma_{\text{r}}} + \widetilde{G}_i(t), \tag{11}$$

$$\frac{d}{dt}\widetilde{G}_i(t) = -\frac{\widetilde{G}_i(t)}{\sigma_{\text{d}}} + \sum_{j\neq i}\sum_k S_{i,j}\delta\left(t - T_{j,k}^{\text{S}}\right) + \sum_k F_i\delta\left(t - T_{i,k}^{\text{F}}\right). \tag{12}$$

We note that, both of the conductance terms $G_i^{\text{S}}(t)$ and $G_i^{\text{F}}(t)$ in Eqs. (6) and (9) are continuous functions, but their first derivatives are discontinuous due to the δ-function in Eqs. (7) and (10).

3 General numerical scheme

For network modeling, we need a stable and accurate numerical scheme to evolve the HH neuron equations coupled with the dynamics of conductances (Eqs. (1)–(10)) for each neuron. Since individual neurons interact with each other through conductance changes associated with spike times, it is also necessary to have numerical interpolation schemes that can determine the spike times accurately and efficiently (Hansel et al. 1998; Shelley and Tao 2001). In most of our study we use the Runge–Kutta fourth-order scheme (RK4) with fixed time step for integrating the system, along with a cubic Hermite interpolation for estimating spike times. The whole scheme is fourth-order accurate.

Here we provide the details of the numerical method for a single neuron. For simplicity, we use the vector X_i to represent all the variables in the solution of the ith neuron:

$$X_i(t) = \left(V_i(t), m_i(t), h_i(t), n_i(t), G_i(t), \widetilde{G}_i(t)\right),$$

where we consider the case of a δ function for the feedforward input (Eqs. (9) and (10)) and use the combined conductance term $G_i(t)$ (Eqs. (11) and (12)). If the continuous sinusoidal function (Eq. (8)) for $G_i^{\mathrm{F}}(t)$ is chosen instead, we certainly can use it to obtain the feedforward input directly.

Given an initial time t_0 and time step Δt, initial values $X_i(t_0)$, and spike times $T_{i,k}^{\mathrm{F}}$ and $T_{j\neq i,k}^{\mathrm{S}}$ from the rest of the network, our method calculates a numerical solution of all variables $X_i(t_0+\Delta t)$ as well as the intervening spike times $T_{i,k}^{\mathrm{S}}$ (if any occurred) for the ith neuron as follows:

Algorithm 1 Single neuron scheme

Step 1: *Input: an initial time t_0, a time step Δt, a set of spike times $T_{i,k}^{\mathrm{F}}$ and $T_{j\neq i,k}^{\mathrm{S}}$ and associated strengths F_i and $S_{i,j}$.*

Step 2: *Consider the time interval $[t_0, t_0+\Delta t]$. Let M denote the total number of feedforward and presynaptic spikes within this interval. Sort these spikes into an increasing list of M spike times T_m^{sorted} with corresponding spike strengths S_m^{sorted}. In addition, we extend this notation such that $T_0^{\mathrm{sorted}} := t_0$, $T_{M+1}^{\mathrm{sorted}} := t_0+\Delta t$ and $S_0^{\mathrm{sorted}} = S_{M+1}^{\mathrm{sorted}} := 0$.*

Step 3: *For $m = 1, \ldots, M+1$, advance the equations for HH neuron model and conductances (Eqs. (1)–(5) with (11) and (12)) from $T_{m-1}^{\mathrm{sorted}}$ to T_m^{sorted} using the standard RK4 scheme to obtain $X_i(T_m^{\mathrm{sorted}})$; Then, update the conductance $\widetilde{G}_i(T_m^{\mathrm{sorted}})$ by adding the appropriate strengths S_m^{sorted}.*

Step 4: *If the calculated values for $V_i(T_m^{\mathrm{sorted}})$ are each less than V^{th}, then we can accept $X_i(T_{M+1}^{\mathrm{sorted}})$ as the solution $X_i(t_0+\Delta t)$. We update $t_0 \leftarrow t_0+\Delta t$ and return to step 2 and continue.*

Step 5: *Otherwise, let $V_i(T_m^{\mathrm{sorted}})$ be the first calculated voltage greater than V^{th}. We know that the ith neuron fired somewhere during the interval $[T_{m-1}^{\mathrm{sorted}}, T_m^{\mathrm{sorted}}]$.*

Step 6: *In this case we use a high-order polynomial interpolation to find an approximation of the spike time t^{fire} in the interval $[T_{m-1}^{\mathrm{sorted}}, T_m^{\mathrm{sorted}}]$. For example, we can use the numerical values of $V_i(T_{m-1}^{\mathrm{sorted}})$, $V_i(T_m^{\mathrm{sorted}})$, $\frac{d}{dt}V_i(T_{m-1}^{\mathrm{sorted}})$, $\frac{d}{dt}V_i(T_m^{\mathrm{sorted}})$ to form a cubic polynomial. We record t^{fire} as the $(k+1)$th postsynaptic spike time $T_{i,k+1}^{\mathrm{S}}$ of the ith neuron. We update $t_0 \leftarrow t_0+\Delta t$ and return to step 2 and continue.*

Now we consider the general framework for simulating the network. A simple strategy, like the traditional clock-driven approach, is to evolve the trajectory of each neuron in the network from t_0 to $t_0+\Delta t$ by using Algorithm 1, while only considering the feedforward spikes within the time interval $[t_0, t_0+\Delta t]$, and the times of all synaptic spikes during this interval being assigned to the end of the step. In this approach we need to take sufficiently small time steps to account for the causality of the spiking events (Hansel et al. 1998; Shelley and Tao 2001). As mentioned in Rangan and Cai (2007), this approach may work well for certain systems, such as an all-to-all mean-driven weakly coupled excitatory network, since the contribution of each spike to the overall conductance of this system is negligible, and the errors introduced at each step do not substantially alter the network dynamics. However, it may fail disastrously when applied to a strongly recurrent system with a large time step. This is because the simple approach does not take into account the fact that the first few of these synaptic spikes induced within a large time step may substantially influence the trajectories of the other spiking neurons in the network via spike-spike interactions, which makes the computation for the rest of the spikes within the time step incorrect.

To outperform this simple approach, we choose a more precise strategy, similar to the event-driven approach (Mattia and Del Giudice 2000; Reutimann et al. 2003; Rudolph and Destexhe 2007). We take the spike-spike correction procedure proposed in Rangan and Cai (2007), which is equivalent to stepping through the sequence of all the synaptic spikes within one time step and computing the effect of each spike on all future spikes. We step through this correction process until the neuronal trajectories and spike times of neurons converge. We refer to Rangan and Cai (2007) for the details of this spike-spike correction algorithm and the general coupling strategy.

Here, we demonstrate the importance of the spike-spike correction procedure for the evolution of networks by applying it to an all-to-all coupled test network (Fig. 1). As shown in Fig. 1, without spike-spike corrections, the subtle errors in the spike times caused by the temporal binning procedure in the clock-driven strategy (that postpones the effects of all the

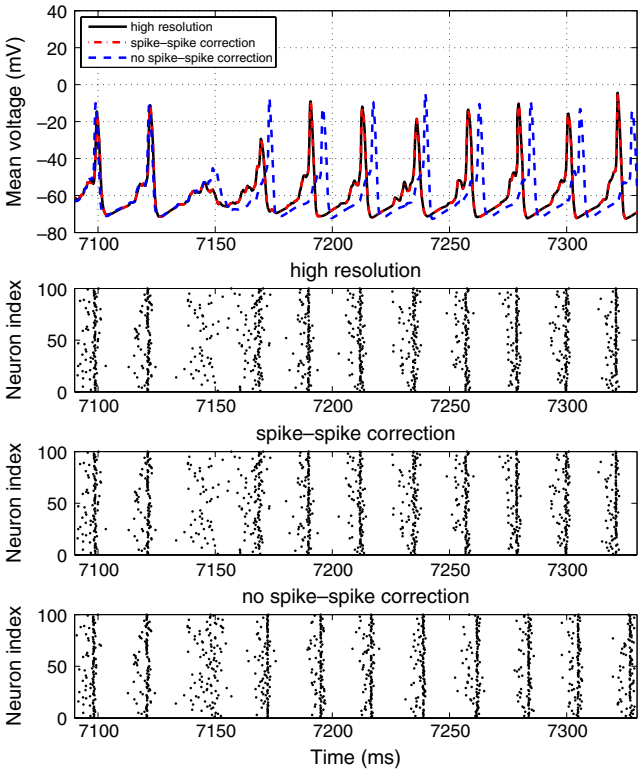

Fig. 1 (Color online) Importance of spike-spike corrections for evolution of networks. We show the mean voltage traces and raster plots of spike events for the general network solver (Algorithm 1 with or without spike-spike corrections) when tested on an all-to-all connected network of 100 excitatory neurons with the coupling strength $S = 0.5$ mS/cm^2. The network is driven by feedforward input of a particular realization of a Poisson process with the rate $\omega = 50$ Hz. The realization of Poisson input is fixed for all the cases shown here. The mean voltage trajectories of the system over the time interval from 7090 to 7330 ms (after the transient) are shown in the *top panel*. The *black solid line* corresponds to the high precision solution (as approximated with very small time step $\Delta t = 2^{-16} \approx 1.5 \times 10^{-5}$ ms), the *red dashed-dotted line* represents the trace computed with much larger time step ($\Delta t = 2^{-4} = 0.0625$ ms) by using Algorithm 1 with spike-spike corrections, and the *blue dashed line* is the solution computed (for the same time step $\Delta t = 0.0625$ ms) by using Algorithm 1 except without spike-spike corrections. Note that, the result with spike-spike corrections and large time step cannot be distinguished from the high precision solution. Below the *top panel* we show the raster plots for these three tests. Clearly, in the case without spike-spike corrections (*bottom panel*) the occurrence of synchronous events is delayed after 7150 ms with respect to the high precision result and the raster plot exhibits a different firing pattern from the high precision result

synaptic spikes during one time step to the end of the step) can accumulate and lead to noticeable delays in later spike times, compared to the strategy with spike-spike corrections. In Fig. 1, we can clearly observe that the occurrence of synchronous events is delayed after $t = 7150$ ms without spike-spike corrections.

4 Library method

4.1 Motivation

Although the general numerical scheme with the spike-spike corrections can evolve the HH neuronal networks quite accurately, there is still some limitation on the time step. It is because explicit Runge–Kutta methods have finite domains of stability, and will have stability problems solving Eq. (1) if the conductances G_i^{F} and G_i^{S} are high (i.e., the equations of the HH system are stiff) and the time step Δt is large. Standard linearly stable schemes like implicit Euler method tend to be of low order accuracy when applied to smooth ODEs (Gear 1971), and may not be very accurate if Δt is large. Moreover, we also have three Eqs. (2) to (4) for the gating variables m, h, n coupled with Eq. (1) to be solved simultaneously, so the implicit method may not be efficient since it requires extra computation for solving the system iteratively in each step.

To overcome the time step limitation due to the stiffness, we propose a specialized numerical method which approximates the trajectories of all the neurons in the network, and allows us to collect accurate statistical information with much lower computational cost. This *library* method, as we mentioned in the Introduction, treats the HH neuron like an I&F neuron. When the membrane potential $V(t)$ reaches the threshold value V^{th}, we stop evolving the HH neuron equations associated with this neuron since the total current (i.e., the right-hand side of Eq. (1)) becomes very large after this moment making these equations very stiff. Instead of resolving the spike by using a very small time step, we recover the spike from our pre-computed high resolution data library. Besides the potential $V(t)$, we also have the *intermediate replica*, i.e., the time courses of the gating variables m, h, n during the spiking period. The spike takes a certain time length like an absolute refractory period in the I&F model. At the end of this period, the potential $V(t)$ drops down and the total current returns to a sufficiently small value, which can allow us to evolve the HH neuron equations of this neuron using a large time step as before. Since the values of V, m, h, n at the end of this period are not fixed, we need to build the data library.

The main advantage of our library method is that we can evolve the HH neuronal networks using a much larger time step than the one used for resolving the whole trajectories without the library, while achieving comparable resolution in statistical quantifications. In Section 5, the numerical results will show that we can still obtain at least 2 or 3 digits of accuracy in the average firing rate by using a time step about 10 times

larger than the typical time step used by the standard RK methods. In the following, we provide the details on how to build the data library and apply it to our method.

4.2 Building the library

Since the dynamics of each neuron is driven by the feedforward input current I^{F} and connected with other neurons through the synaptic current I^{S}, the input current I^{input} (i.e., the sum of I^{F} and I^{S}, Eq. (5)) is the main force to drive the dynamics. Hence, we build the library based on different neuronal trajectories of a single neuron under different values of I^{input}.

Another key point is that the data library will be used to recover the spikes whenever a neuron fires, so the starting point of the time-course of membrane potential in the library is the threshold value V^{th}. In our library, and in the numerical tests of network dynamics, we take $V^{\mathrm{th}} = -50$ mV, which is sufficiently low to keep Eq. (1) not stiff before the spiking and allows us to use large time steps. Meanwhile, it is also sufficiently high that the neuron will definitely fire after its membrane potential reaches this threshold value. If V^{th} is set too low, some subthreshold potential fluctuation might be misidentified as a spike. We can choose other threshold values, but the library needs to be rebuilt.

The first step is to isolate a time-course of a spike. As mentioned above, we consider a single neuron and take the input current I^{input} in Eq. (1) as constant $I^{\mathrm{input}} = -G_0(V^{\mathrm{th}} - V_{\mathrm{G}}) = 50G_0$ (note that $V_{\mathrm{G}} = 0\mathrm{mV}$) and G_0 is a parameter constant within a certain range from 0.15 to 1.0 mS/cm^2. Then the range of input current is from 7.5 to 50.0 μA/cm^2, which can essentially cover all typical values of I^{input} of the spiking neurons *at the moments when they fire* in our network simulations, even for some special case of strong synchronization with a very large coupling strength S up to 1.5 mS/cm^2.

We choose N_{I} different values of I^{input} equally distributed in its range. With each I^{input}, we evolve the HH neuron equations of the single neuron on very fine time step to obtain a high resolution trajectory. After a transient, the neuron fires periodically and we isolate one spike that starts *exactly* from a spike time t^{fire} (not the time step point t_{n+1} immediately after t^{fire}) to a later time where the potential drops down *around* its minimum (Fig. 2). A technical detail we should point out is that since the spike time t^{fire} can rarely happen at a precise time step point, there is always a time difference $t_{n+1} - t^{\mathrm{fire}}$ between the numerical trajectory on the time step points and the data of a single spike time-course that we need. Therefore, for high accuracy it is also necessary to interpolate the membrane potential

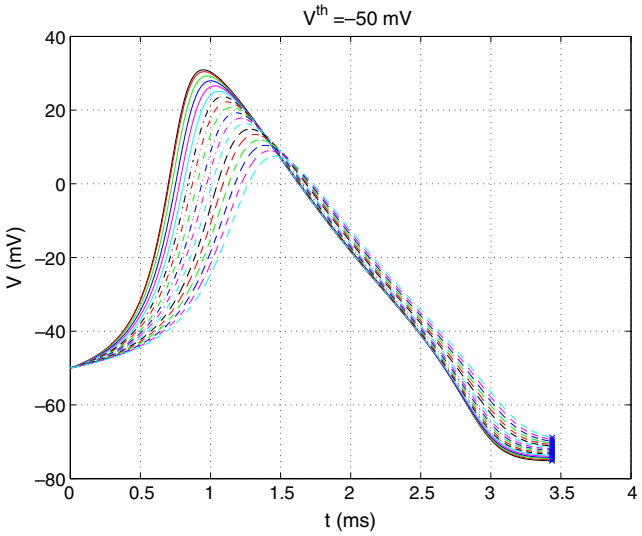

Fig. 2 (Color online) Isolated time-courses of single spikes obtained by applying different input currents I^{input} to a single neuron. Each *curve* corresponds to one spike. Shown are total $N_{\mathrm{I}} = 18$ spikes corresponding to different I^{input} ranging from 7.5 to 50.0 μA/cm^2. All time-courses start exactly from the same threshold value $V^{\mathrm{th}} = -50$ mV and have the same time interval $T^{\mathrm{ref}} = 3.4375$ ms. The time step $\delta t = 2^{-13} \approx 1.22 \times 10^{-4}$ms is used to obtain these solutions

in each time step due to this time shift, even though the numerical error can be very small by using a very fine time step δt. We used a cubic Hermite interpolation to obtain the time-courses.

We refer to the time interval of this isolated time-course as the absolute refractory period T^{ref} which should not be confused with the true refractory period of the HH neuron and which can be interpreted as the refractory period in the sense of the I&F neuron. In our calculation, we fix T^{ref} for all cases with different input currents I^{input}, therefore, the end point of spike in each case will have different values of potential (Fig. 2). We denote the value by V^{re} where the superscript -re stands for reset value. Therefore, for each value of G_0 corresponding to an input current I^{input}, we thus isolate a time-course of a spike to build the library for the membrane potential V.

Although the neuron always fires at the same threshold V^{th}, the gating variable may have different values at different spike times. Therefore, in the next step, we build the library for each gating variable. Once we know the spike time t^{fire}, we can also use numerical interpolation to obtain the values of gating variables m, h, n at that time. We denote them by m^{th}, h^{th}, n^{th}, respectively. Equations (2)–(4) show that the dynamics of m, h, n only depend on the membrane potential $V(t)$ and each of them has no direct effect on the others.

Table 1 The data structures of I^{input}, V and m in the library

Those of h and n are similar to that of m. We omit the superscript -input from the current I^{input}. Note that the output of V and m has another dimension in time t

input: $(m_1, \ldots, m_k, m_{k+1}, \ldots, m_{N_{\text{lib}}})$ $\Downarrow$		
input:	output:	output:
$\begin{pmatrix} I_1 \\ \ldots \\ I_j \\ I_{j+1} \\ \ldots \\ I_{N_{\text{I}}} \end{pmatrix} \Rightarrow$	$\begin{pmatrix} V_1 \\ \ldots \\ V_j \\ V_{j+1} \\ \ldots \\ V_{N_{\text{I}}} \end{pmatrix}$	$\begin{pmatrix} m_{1,1}, \ldots, m_{1,k}, m_{1,k+1}, \ldots, m_{1,N_{\text{lib}}} \\ \ldots \\ m_{j,1}, \ldots, m_{j,k}, m_{j,k+1}, \ldots, m_{j,N_{\text{lib}}} \\ m_{j+1,1}, \ldots, m_{j+1,k}, m_{j+1,k+1}, \ldots, m_{j+1,N_{\text{lib}}} \\ \ldots \\ m_{N_{\text{I}},1}, \ldots, m_{N_{\text{I}},k}, m_{N_{\text{I}},k+1}, \ldots, m_{N_{\text{I}},N_{\text{lib}}} \end{pmatrix}$

With this key fact, we can simply use the isolated time-course of membrane potential to compute the intermediate replica of m, h, n individually over the time course of T^{ref}. In particular, this way we can obtain the reset values $m^{\text{re}}, h^{\text{re}}, n^{\text{re}}$ at the end point of the spike for each case of I^{input}.

For example, in order to build the data library of m, we equally distribute N_{lib} points of m^{input} in the parameter range $[m^{\text{input}}_{\min}, m^{\text{input}}_{\max}]$. This range should cover all typical values of m^{th} of a neuron when it fires in a HH network. Then, starting from the beginning of the isolated time-course of the spike with a chosen value of m^{input}, we evolve Eq. (2) of m on the same time step δt only for a time interval of the isolated time-course, i.e., T^{ref}. We emphasize that in each time step t_{n+1} of RK4 scheme, the values of $V(t_n)$ and $V(t_{n+1})$ are known from the isolated time-course and $m(t_n)$ has been obtained from previous step t_n, then, we just need to compute the value of $m(t_{n+1})$ that is slaved to the potential V due to the fact that the dynamics of m only depends directly on V (Eq. (2)). In the end, the reset value of m^{re} is computed. Meanwhile, we refer to the whole trace of m as the output m^{output}. By repeating this process for all different values of m^{input}, we can obtain a total of N_{lib} sets of the intermediate replica for m as the output.

The intermediate replica for h and n can be obtained in the same way. After we obtain a data suite of V, m, h, n for one case of I^{input}, we repeat the whole procedure for all the other cases of I^{input}. Finally, we can get N_{I} suites of data libraries. In each suite, we have one trace of V and N_{lib} traces of m, h, n, respectively (see Table 1 for the data structures and Fig. 3 for the ranges of the reset values V^{re}, m^{re}, h^{re}, n^{re}). The total size of the library is $(3N_{\text{lib}} + 1)N_{\text{I}}N_{\delta t}$ where $N_{\delta t} = T^{\text{ref}}/\delta t$ is total time steps to resolve the isolated time-course of the spike. We note that when building the library, we make the approximation that all of V, m, h, n are the functions of the input current I^{input}. In Table 1, for simplicity, we omit the parts of h and n since their data structures are similar to the one of m. In summary, since V depends on I^{input}, the data structure of V is one dimensional. But the data structures of m, h, n are two dimensional since their reset values depend on I^{input} and their own input values.

Now we comment on the choice of N_{I} and N_{lib}. Larger values of N_{I} and N_{lib} mean more cases of the input current and more interpolation points of m^{input}, h^{input}, n^{input} to compute, increasing the accuracy of the data library as well as the size of the library. In our simulation, we take $N_{\text{I}} = 18$ and $N_{\text{lib}} = 31$ with $\delta t = 2^{-13}$ms, which can make the library sufficiently accurate as is shown below. In terms of memory usage, the size of the library is approximately 45 megabyte in binary form, and it can be easily handled by current computers.

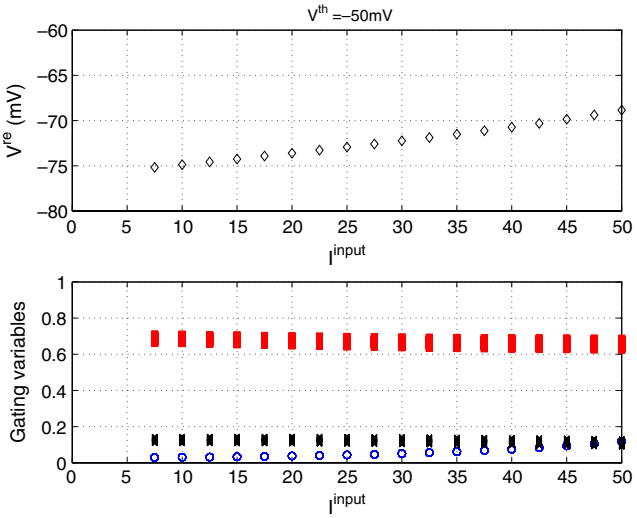

Fig. 3 (Color online) The ranges of the reset values $V^{\text{re}}, m^{\text{re}}, h^{\text{re}}, n^{\text{re}}$ at the end of the time-course of spike versus different external currents I^{input} ranging from 7.5 to 50.0μA/cm^2 ($N_{\text{I}} = 18$). In the *top panel*, we show the reset values V^{re} labeled by *diamonds*. In the *bottom panel*, the reset values m^{re}, h^{re}, n^{re} are labeled by *circles*, *crosses*, and *squares*, respectively. Note that there are $N_{\text{lib}} = 31$ data points for each type of reset value at every I^{input}. The *circles* of m^{re} are so close that only one circle can be seen at each I^{input}. For the data shown, the ranges of the input values $m^{\text{input}} \in [0.15, 0.3]$, $h^{\text{input}} \in [0.25, 0.55]$, $n^{\text{input}} \in [0.3, 0.54]$

4.3 Use of the library

Once we have the data library, we can evolve the HH neuronal networks using a large time step Δt. Here, we illustrate how to use the library by considering a single neuron in the network. For brevity of notation, we omit the index i for the time being. First, the membrane potential $V(t)$ and the gating variables $m(t)$, $h(t)$, $n(t)$ of this neuron evolve continuously until the potential reaches the threshold V^{th}. At this point in time, the neuron produces a spike which can be recovered from the library by numerical interpolation. Usually, we simply stop evolving the HH neuron equations (Eqs. (1)–(4)) for this neuron at this time and restart after T^{ref} ms with the reset values V^{re}, m^{re}, h^{re}, n^{re} interpolated from the library. We only recover the action potential when the data points of membrane potential are needed for some analysis, e.g., power spectra analysis of time-courses. We note that during the absolute refractory period the conductance terms $G^{\text{F}}(t)$ and $G^{\text{S}}(t)$ still evolve as usual. Therefore, in our library method, it is straightforward to use other models to describe the dynamics of the conductance terms, such as 2-state kinetic models, instead of the form resembling an α-function in Eqs. (11) and (12).

We now describe how to obtain the reset values by interpolation from the library. First we need to calculate the input current I^{input} at the spike time t^{fire} by: $I^{\text{th}} = -\left(G^{\text{F}}(t^{\text{fire}}) + G^{\text{S}}(t^{\text{fire}})\right)(V^{\text{th}} - V_{\text{G}})$ as well as the gating variables m^{th}, h^{th}, and n^{th} by high order interpolation consistent with the RK4 scheme. Then we do a linear interpolation for I^{th} in the current/potential (I-V) table and compute the corresponding reset value of potential V^{re}. From the input data set of current we need to find two data points I_j^{input} and I_{j+1}^{input} between which I^{th} falls, and calculate the interpolation coefficient λ_I which satisfies the following relationship (see Table 1):

$$I^{\text{th}} = \lambda_I I_j^{\text{input}} + (1-\lambda_I) I_{j+1}^{\text{input}}, \tag{13}$$

then, we evaluate the corresponding reset value of potential:

$$V^{\text{re}} = \lambda_I V_j^{\text{output}} + (1-\lambda_I) V_{j+1}^{\text{output}}. \tag{14}$$

For the gating variable m, given that it is between two data points m_k^{input} and m_{k+1}^{input} in the input data set, we need to do a similar interpolation for m^{th} with the interpolation coefficient λ_m satisfying (see Table 1):

$$m^{\text{th}} = \lambda_m m_k^{\text{input}} + (1-\lambda_m) m_{k+1}^{\text{input}}. \tag{15}$$

Then in two output data sets $m_{j,\bullet}^{\text{output}}$ and $m_{j+1,\bullet}^{\text{output}}$ with j coming from the relationship (13), we can compute two interpolation values m_j^{re} and m_{j+1}^{re} according to the relationship (15):

$$m_j^{\text{re}} = \lambda_m m_{j,k}^{\text{output}} + (1-\lambda_m) m_{j,k+1}^{\text{output}}, \tag{16}$$

$$m_{j+1}^{\text{re}} = \lambda_m m_{j+1,k}^{\text{output}} + (1-\lambda_m) m_{j+1,k+1}^{\text{output}}. \tag{17}$$

Finally, we use the relationship (13) again to obtain the reset value

$$m^{\text{re}} = \lambda_I m_j^{\text{re}} + (1-\lambda_I) m_{j+1}^{\text{re}}. \tag{18}$$

The values of h^{re} and n^{re} can be obtained in the same way. Moreover, by doing the interpolation in the time direction from the library, we can also recover the values of $V(t)$, $m(t)$, $h(t)$, and $n(t)$ for t inside the refractory period.

We point out that occasionally the values of I^{th}, m^{th}, h^{th} and n^{th} may be out of their input ranges. In this situation we may do linear extrapolation instead of interpolation to compute their reset values.

We emphasize that there is an assumption made for our method: once a neuron fires, its input current I^{input} keeps the same value I^{th} throughout the entire time-course of the refractory period T^{ref}. Generally, this is not true since, from Eq. (5), we can clearly see that the conductance terms $G^{\text{F}}(t)$ and $G^{\text{S}}(t)$ of this neuron are changing due to both the feedforward input and synaptic spikes from other neurons in the network during this period. This approximation is based on the following observations: (1) although I^{input} varies during the refractory period, the maximum of its range is usually about 20–30 μA/cm^2; (2) the intrinsic current (i.e., the sum of ionic and leakage currents) is about 50–60 μA/cm^2 at the spike time t^{fire}; and (3) the intrinsic current increases very rapidly to 200–300 μA/cm^2 after the spike time. From observations (1) and (2), we perform an accurate interpolation of I^{input} in the library because the contribution of I^{input} to the dynamics of the membrane potential is comparable to the contribution of the intrinsic current. From observation (3), the intrinsic current is dominant in the dynamics, therefore, we may keep I^{input} as constant during T^{ref}. These are the approximations underlying our library method.

We conclude this section by briefly outlining a variation, Algorithm 2, of Algorithm 1 in Section 3. Here, Algorithm 2 uses the library to recover the spike. Given an initial time t_0 and time step Δt, initial values of each state variable in the HH neuron equations at time t_0, and spike times $T_{i,k}^{\text{F}}$ and $T_{j\neq i,k}^{\text{S}}$ from the rest of the

network, our method calculates a numerical solution of all variables at the next time step $t_0 + \Delta t$ as well as the intervening spike times $T^{\rm S}_{i,k}$ (if any occurred) for the ith neuron as follows:

Algorithm 2 Library algorithm

Step 0: *Pre-compute the data library of V, m, h, n for $N_{\rm I}$ different values of the constant input current $I^{\rm input}$ using very fine time step δt. For each $I^{\rm input}$, we isolate a spike that starts exactly from $V^{\rm th}$ to a later time (after $T^{\rm ref}$ms) where the membrane potential drops down around its minimum. Then we use this spike (i.e., the intermediate replica of the membrane potential) with different values of $m^{\rm input}, h^{\rm input}, n^{\rm input}$ as initial values to compute the intermediate replica for m, h, n individually. In particular, we can obtain the reset values $m^{\rm re}$, $h^{\rm re}$, $n^{\rm re}$ at the end point of the spike. For each case of $I^{\rm input}$, there are $N_{\rm lib}$ data sets of m, h, n, respectively.*

Step 1: *Input: the library, an initial time t_0, a large time step Δt, a set of spike times $T^{\rm F}_{i,k}$ and $T^{\rm S}_{j\neq i,k}$ and associated strengths F_i and $S_{i,j}$.*

Step 2: *Consider the time interval $[t_0, t_0 + \Delta t]$. Let M denote the total number of feedforward and presynaptic spikes within this interval. Sort these spikes into an increasing list of M spike times $T^{\rm sorted}_m$ with corresponding spike strengths $S^{\rm sorted}_m$. In addition, we extend this notation such that $T^{\rm sorted}_0 := t_0$, $T^{\rm sorted}_{M+1} := t_0 + \Delta t$ and $S^{\rm sorted}_0 = S^{\rm sorted}_{M+1} := 0$.*

Step 3: *For $m = 1, \ldots, M+1$, advance the equations for HH neuron model and conductances (Eqs. (1)–(5) with (11) and (12)) from $T^{\rm sorted}_{m-1}$ to $T^{\rm sorted}_m$ using the standard RK4 scheme to obtain the solution $X_i\left(T^{\rm sorted}_m\right)$; Then, update the conductance $\widetilde{G}_i\left(T^{\rm sorted}_m\right)$ by adding the appropriate strengths $S^{\rm sorted}_m$.*

Step 4: *If the calculated values for $V_i\left(T^{\rm sorted}_m\right)$ are each less than $V^{\rm th}$, then we can accept $X_i\left(T^{\rm sorted}_{M+1}\right)$ as the solution $X_i(t_0 + \Delta t)$. We update $t_0 \leftarrow t_0 + \Delta t$ and return to step 2 and continue.*

Step 5: *Otherwise, let $V_i\left(T^{\rm sorted}_m\right)$ be the first calculated voltage greater than $V^{\rm th}$. We know that the ith neuron fired somewhere during the interval $\left[T^{\rm sorted}_{m-1}, T^{\rm sorted}_m\right]$.*

Step 6: *In this case we use a high-order polynomial interpolation to find an approximation to the spike time $t^{\rm fire}$ in the interval $\left[T^{\rm sorted}_{m-1}, T^{\rm sorted}_m\right]$. For example, we can use the numerical values of $V_i\left(T^{\rm sorted}_{m-1}\right)$, $V_i\left(T^{\rm sorted}_m\right)$, $\frac{d}{dt}V_i\left(T^{\rm sorted}_{m-1}\right)$, $\frac{d}{dt}V_i\left(T^{\rm sorted}_m\right)$ to form a cubic polynomial. We record $t^{\rm fire}$ as the $(k+1)$th postsynaptic spike time $T^{\rm S}_{i,k+1}$ of the ith neuron.*

Step 7: *We compute the values of $I^{\rm th} = -\left(G^{\rm F}(t^{\rm fire}) + G^{\rm S}(t^{\rm fire})\right)(V^{\rm th} - V_{\rm G})$ as well as the gating variables $m^{\rm th}$, $h^{\rm th}$ and $n^{\rm th}$ at this time. Then, we perform a linear interpolation to find the corresponding reset values of $V^{\rm re}, m^{\rm re}, h^{\rm re}, n^{\rm re}$ in the library. Meanwhile, we stop evolving Eqs. (1)–(5) for the next $T^{\rm ref}$ms, but evolve Eqs. (11) and (12) for the conductance terms $G^{\rm F}(t)$ and $G^{\rm S}(t)$ as usual. We update $t_0 \leftarrow \min(t^{\rm fire} + T^{\rm ref}, t + \Delta t)$ and return to step 2 and continue with the reset values $V^{\rm re}, m^{\rm re}, h^{\rm re}, n^{\rm re}$ as the initial values $V_i(t_0)$, $m_i(t_0)$, $h_i(t_0)$ and $n_i(t_0)$ (Fig. 4).*

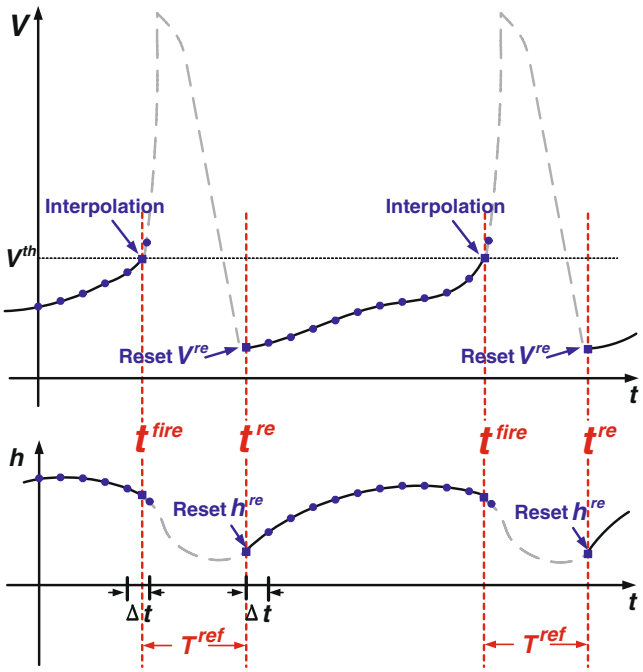

Fig. 4 (Color online) Illustration of the library method, especially for step 7 in Algorithm 2. Note that only the trajectories of $V(t)$ and $h(t)$ are shown in the *top* and *bottom panels*, respectively, and those of $m(t)$ and $n(t)$ are similar. First we use a high-order interpolation to find the spike time $t^{\rm fire}$ when $V(t)$ reaches the threshold $V^{\rm th}$ (the *horizontal dotted line* in the *top panel*). We compute the values of $I^{\rm th}$ as well as the gating variables $m^{\rm th}$, $h^{\rm th}$ and $n^{\rm th}$ at this time. Then we do linear interpolation to find the corresponding reset values of $V^{\rm re}, m^{\rm re}, h^{\rm re}, n^{\rm re}$ in the library. After stopping the evolution of Eqs. (1)–(5) during the refractory period $T^{\rm ref}$, we restart it with these reset values. Equations (11) and (12) for the conductance terms $G^{\rm F}(t)$ and $G^{\rm S}(t)$ are solved as usual. By doing the interpolation in the time direction from the library, we can also recover the values of $V(t)$ and $h(t)$ in the refractory period when needed (the *dashed curves*). For clarity, the size of Δt shown in the figure is actually much larger than what is used in simulations

5 Numerical results

5.1 Three regimes of the network

We again consider an all-to-all connected network of 100 excitatory neurons, but the feedforward input is the continuous sinusoidal function of time t (Eq. (8)) with oscillation angular frequency $\omega = 50$ Hz. Hence, the entire network model is a deterministic dynamical system. Other parameters are given in Appendix. We perform simulations of this network for synaptic coupling strength S ranging from 0.2 to 0.4 mS/cm^2 with an increment of $\Delta S = 0.005$ mS/cm^2. The results reveal three typical dynamical regimes—a state phase-locked to the external drive, a chaotic state, and a state exhibiting nearly synchronous activity. The systematic scanning results will be presented in Section 5.3.

(i) *Phase-locked state* For small values of the coupling strength, the current due to the synaptic spikes I^{S} is so weak that the dynamics of each neuron is largely *phase locked* to the sinusoidal feedforward input. Each neuron fires periodically with the ISI = 20 ms. A raster plot of the case $S = 0.2$ mS/cm^2 is shown in Fig. 5(a). The spike times of any given neuron in the network reflect the phase associated with that neuron's feedforward input. This phase-locked state exists for $0.2 \lesssim S \lesssim 0.245$ mS/cm^2.

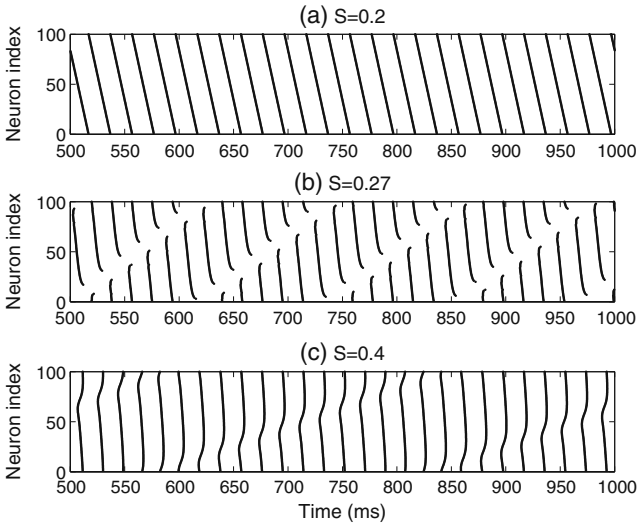

Fig. 5 Raster plots of spike events in an all-to-all connected network of 100 excitatory neurons driven by the continuous sinusoidal feedforward input with oscillation angular frequency $\omega = 50$ Hz. The plots from *top* to *bottom* show three typical cases with the coupling strength $S = 0.2$, 0.27, and 0.4 mS/cm^2, respectively

This case is also illustrated through the return map of projected spike times and the power spectrum, averaged over all neurons, of membrane potential trace (Fig. 6(a) and (d)). To construct the return map, we record the kth spike time $T^{\mathrm{S}}_{i,k}$ of a chosen neuron in the network, say the ith neuron, and project it into a time window with the same length as the ISI. The projected spike time $\widetilde{T}^{\mathrm{S}}_{i,k}$ is given as

$$\widetilde{T}^{\mathrm{S}}_{i,k} = T^{\mathrm{S}}_{i,k} \mod \omega^{-1}. \tag{19}$$

In Fig. 6(a), we plot the return map of projected spike times of the first neuron in the network, i.e., $T^{\mathrm{S}}_{1,k+1}$ versus its previous spike time $T^{\mathrm{S}}_{1,k}$. We see that the spike times always stay at same location, signifying a periodic firing. The power spectrum, averaged over all neurons, of membrane potential trace, shown in Fig. 6(d), contains peaks clearly located at integer multiples of the fundamental frequency 50 Hz, indicating that the membrane potential evolves periodically.

(ii) *Chaotic regime* For intermediate coupling strength, in this network for $0.245 \lesssim S \lesssim 0.318$ mS/cm^2, the statistical results for long time simulation show that the dynamics of the network is truly *chaotic* since its largest Lyapunov exponent is measured to be positive, as will be discussed below. A raster plot of the case $S = 0.27$ mS/cm^2 is shown in Fig. 5(b).

As shown in Fig. 6(b), the return map of projected spike times of the first neuron in the network is a complicated geometric object with data points spreading over it due to the fractal dimension of the attractor, which is typical of a chaotic system. The power spectrum, averaged over all neurons, of membrane potential trace in Fig. 6(e) exhibits a noisy signal and broad-band nature, which is also characteristic of a chaotic system. The spectrum contains peaks indicating the predominant frequencies of the solution.

(iii) *"Nearly synchronous" state* When the coupling is strong, $S \gtrsim 0.318$ mS/cm^2, a large proportion of neurons in the network fire almost synchronously after a few of the neurons fire in advance. This is shown in a raster plot of the case $S = 0.4$ mS/cm^2 in Fig. 5(c).

Moreover, the network is in a *quasi-periodic* phase, as indicated by the return map of projected spike times of the first neuron shown in Fig. 6(c), which forms a circle map with strictly monotonic increasing branches, i.e., no chaotic trajectories for the map (Schuster and Just 2005). The power spectrum, averaged over all neurons, of membrane potential trace of the quasi-periodic phase (Fig. 6(f)) consists of not only the main spectrum of the periodic motion, but also many tightly spaced sidebands due to the relatively slow modulation.

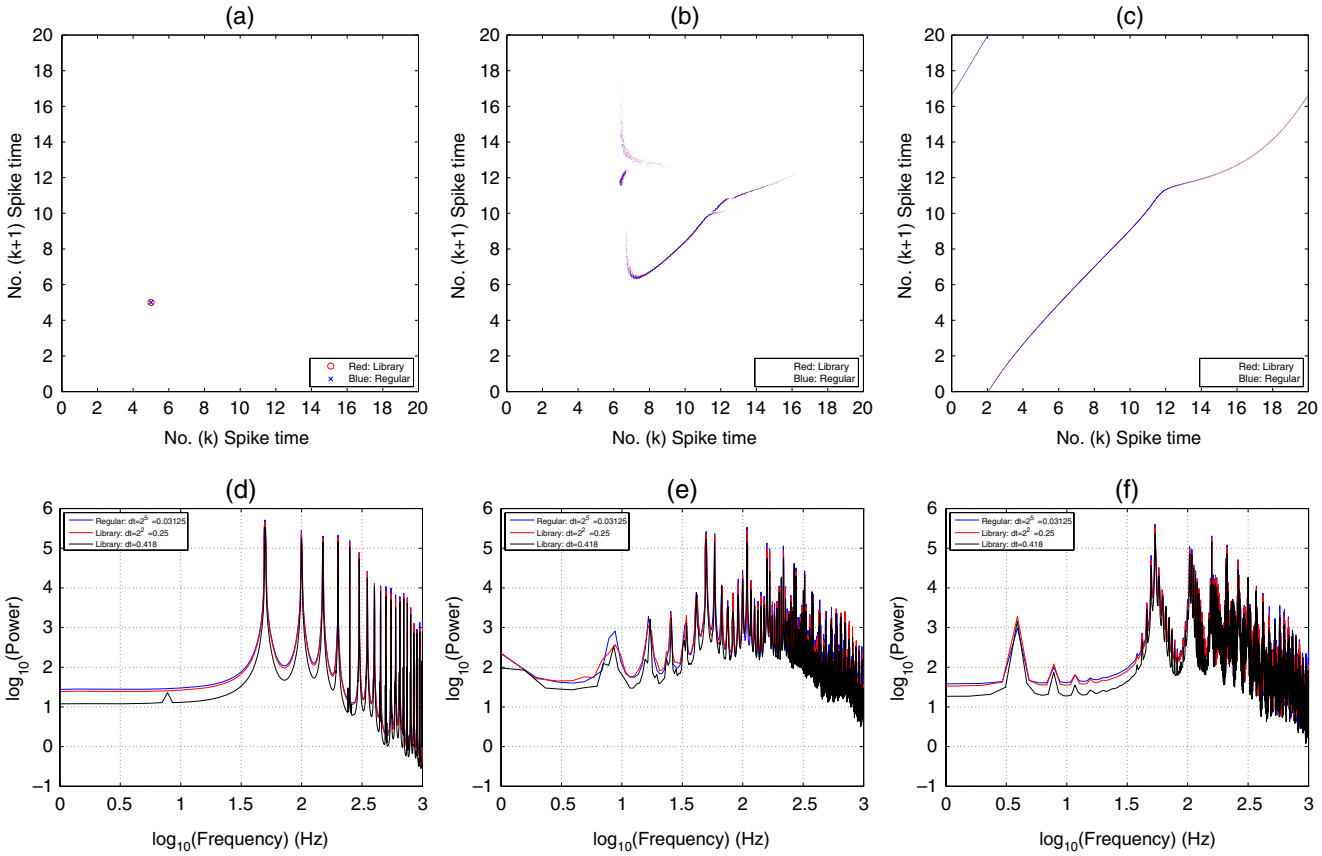

Fig. 6 (Color online) (*Top panel*): return maps of projected spike times of the first neuron in an all-to-all connected network of 100 excitatory neurons driven by the continuous sinusoidal feedforward input with oscillation angular frequency $\omega = 50$ Hz. The plots from (**a**) to (**c**) show three cases with the coupling strength $S = 0.2, 0.27$, and $0.4\ \mathrm{mS/cm^2}$, respectively. In each case, we compare the results obtained from both the regular method (*blue dots*) and the library method (*red dots*) with the same time step ($\Delta t = 2^{-5} = 0.03125$ ms). In particular, for the case $S = 0.2\ \mathrm{mS/cm^2}$ in (**a**), we use *circles* and *crosses* to represent the spike times since they all appear at the same location. The results of both methods match each other surprisingly well. (*Bottom panel*): The power spectrum, averaged over all neurons, of membrane potential trace in the network with different coupling strengths. In each plot, the *blue line* corresponds to the result using the regular method with time step ($\Delta t = 0.03125$ ms); the *red line* represents the spectrum computed with much larger time step ($\Delta t = 0.25$ ms) by using the library method, and the *black line* is the solution computed for the maximum time step ($\Delta t = 0.418$ ms) with the library method. We note that the library method with large time step can capture those dominant frequencies as well as the regular method does with a small time step

We remark that both the regular method (Algorithm 1 with spike-spike corrections) and the library method (Algorithm 2 with spike-spike corrections) were used in the numerical tests of three typical coupling strengths shown above. The patterns of raster plots exhibit similar dynamic regimes for both methods, therefore, we only show the raster plots obtained by using the regular method (Fig. 5). In Fig. 6, the results from both methods are compared. As we explain about the library method in Section 4, if the values of $V(t)$ in the refractory period are needed for computing the power spectrum of membrane potential trace, we can obtain them by appealing to the library and interpolating as necessary. We emphasize that the library method with a large time step can achieve results as good as those from the regular method using a small time step. These comparison results are in excellent agreement, which presents strong evidence for the success of our method.

5.2 Convergence tests

To verify the accuracy of our numerical methods—both the regular and library methods, we perform convergence tests of the numerical solutions for the same network as described in the previous section. For each method, we obtain a high precision solution at time $t = 1024$ ms with a time step ($\Delta t = 2^{-16} \approx 1.5 \times 10^{-5}$ ms) which is sufficiently small so that the solutions by using the algorithm with or without spike-spike corrections

produce the same convergent solution. We take the convergent solution as a representation of the high precision solution $\mathbf{X}^{\text{high}}(t)$. Here, for simplicity of notation, we use a vector $\mathbf{X}(t) = [X_1(t), \ldots, X_i(t), \ldots, X_N(t)]$ to represent all variables in the solution of all neurons. The sub-vector $X_i(t)$ stands for the solution of the ith neuron:

$$X_i(t) = \big(V_i(t), m_i(t), h_i(t), n_i(t), G_i(t)\big), \tag{20}$$

where $G_i(t)$ is the total conductance from both of the feedforward input and synaptic input. We compare the high precision solution $\mathbf{X}^{\text{high}}(t)$ with the trajectories $\mathbf{X}^{\Delta t}(t)$ calculated with larger time steps $\Delta t = 2^{-9} \rightarrow 2^{-4}$ ms. We measure the numerical error in the L^2-norm as follows:

$$E \equiv \left\| \mathbf{X}^{\Delta t} - \mathbf{X}^{\text{high}} \right\|. \tag{21}$$

As shown in Fig. 7(a), the regular method can achieve fourth-order accuracy when $S = 0.2$ (phase-locked state) and 0.4 mS/cm^2 (nearly synchronous state). However, for $S = 0.27$ and 0.31 mS/cm^2, due to the chaotic dynamics, we cannot achieve convergence of the numerical solutions. A similar phenomenon also occurs when we use the library method in Fig. 7(b).

We note that in the library method, we recover the trajectories of a neuron by linear interpolation from the library whenever this neuron is in the refractory period. Therefore, there is a numerical error associated with the interpolation and the accuracy of the library. Fortunately, the number of spikes is finite and much smaller than the total number of time steps (a ratio of $1 : 10^3 \sim 10^5$). Moreover, since we use a cubic Hermite interpolation for estimating the spike times, the full method can still be fourth-order accurate in some cases. For the case $S = 0.4$ mS/cm^2, where the network is in a nearly synchronous state, it occurs that almost all of the neurons are out of the refractory period at $t = 1024$ ms (Fig. 5(c)), which allows most of the solution values to be computed from the regular numerical integration at this moment, not by interpolation from the library. So in this situation we can achieve fourth-order accuracy. In the other case $S = 0.2$ mS/cm^2, where the network is in the phase-locked regime, there are always some neurons in the refractory period (Fig. 5(a)). Hence, we need to do interpolation from the library for part of the solution values, which introduces extra numerical error. When the time step is small ($\Delta t \lesssim 2^{-6}$ ms, i.e., $\log_{10}(\Delta t) \lesssim -1.8$), this type of numerical error is dominant and the convergence cannot be achieved as the time step becomes smaller. That is why the curve of numerical errors for $S = 0.2$ mS/cm^2 becomes flat when Δt is less than 2^{-6}ms (Fig. 7(b)).

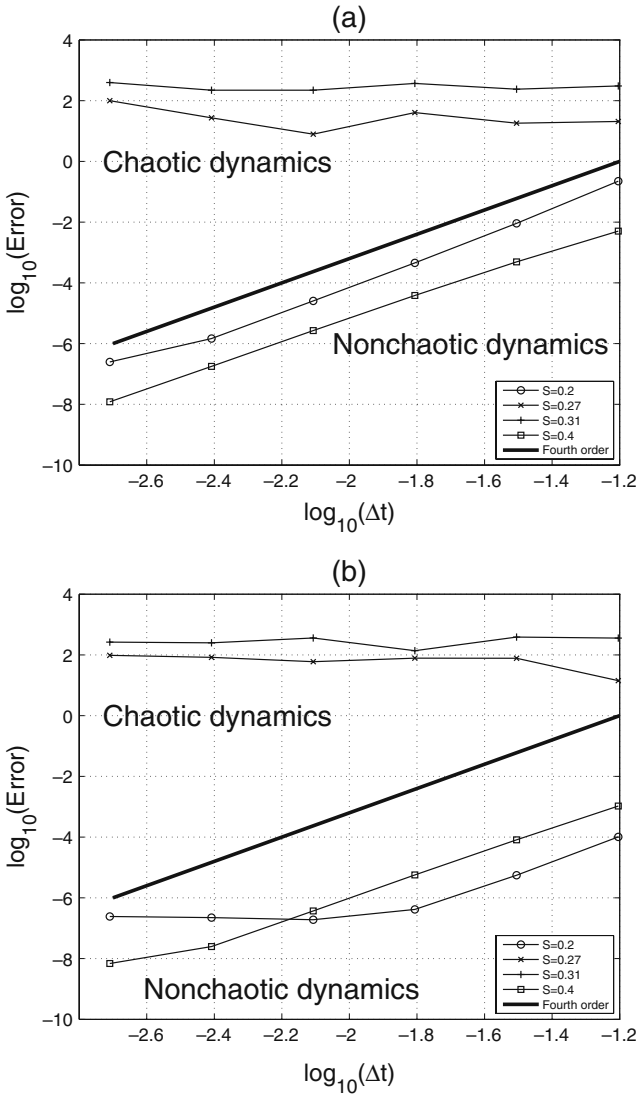

Fig. 7 Comparison of accuracy between (**a**) the regular method and (**b**) the library method. The convergence tests are performed on the same network as the one in Fig. 6 by using the RK4 scheme in both methods with a final time of $t = 1024$ ms. In each plot we show four cases with the coupling strength $S = 0.2$ (*circles*), 0.27 (*crosses*), 0.31 (*pluses*) and 0.4 mS/cm^2 (*squares*), respectively

In summary, both the regular and library methods can achieve numerical convergence for the networks within the phase-locked and nearly synchronous regimes. For the chaotic regime, there is no convergence of the solutions.

5.3 Lyapunov exponent

A useful tool for characterizing chaos in a dynamical system is the set of Lyapunov exponents, in particular, the largest one, which measures the rate of exponential divergence or convergence from perturbed initial states of the system. The chaotic dynamics is signified

by a positive largest Lyapunov exponent. Generally, the largest Lyapunov exponent λ can be obtained by following two nearby trajectories $\mathbf{X}(t)$ and $\mathbf{X}'(t)$ and calculating their average logarithmic rate of separation:

$$\lambda = \lim_{t\to\infty}\lim_{\epsilon\to 0}\frac{1}{t}\ln\frac{\|\mathbf{Z}(t)\|}{\|\mathbf{Z}_0\|}, \tag{22}$$

where $\mathbf{Z}(t) = \mathbf{X}(t) - \mathbf{X}'(t)$, $\|\mathbf{Z}_0\| = \epsilon$ and $\mathbf{Z}(0) = \mathbf{Z}_0$ is the initial separation. However, for a chaotic system, at least one Lyapunov exponent is positive which implies that $\|\mathbf{Z}(t)\|$ is unbounded as $t \to \infty$. Therefore, a practical approach to avoid overflow is to scale back one of the trajectories, say $\mathbf{X}'(t)$, to the vicinity of the other $\mathbf{X}(t)$ along the direction of separation whenever they become too far apart. We refer to this step as *renormalization*. In our study we calculate the divergence of nearby trajectories with finite time steps τ, and after each step we renormalize $\|\mathbf{Z}(n\tau)\|$ to a fixed ϵ and take the average of separation rate after sufficiently long time to obtain λ via:

$$\begin{aligned}
&\mathbf{Z}(\tau) = \mathbf{Z}(0)\exp(\lambda_1\tau);\\
&\mathbf{Z}(2\tau) = \mathbf{Z}(\tau)\frac{\epsilon}{\|\mathbf{Z}(\tau)\|}\exp(\lambda_2\tau);\ldots.\\
&\mathbf{Z}(k\tau) = \mathbf{Z}((k-1)\tau)\frac{\epsilon}{\|\mathbf{Z}((k-1)\tau)\|}\exp(\lambda_k\tau);\ldots.
\end{aligned} \tag{23}$$

and

$$\lambda = \lim_{n\to\infty}\frac{1}{n}\sum_{k=1}^{n}\lambda_k = \lim_{n\to\infty}\frac{1}{n\tau}\sum_{k=1}^{n}\ln\frac{\|\mathbf{Z}(k\tau)\|}{\epsilon}. \tag{24}$$

The details of this algorithm and how to choose a suitable value of ϵ can be found in Parker and Chua (1989) and Schuster and Just (2005).

We remark that traditionally the Lyapunov exponent is defined for a smooth dynamical system. Here, we use an extension of the notion of the Lyapunov exponents, which employs only continuous dynamical variables (Sun et al., submitted). Hence, the additional variable of the synaptic conductance term $\widetilde{G}_i^{\mathrm{S}}$ is excluded because it is discontinuous due to the δ-function in Eq. (7), i.e., we only consider:

$$X_i(t) = \big(V_i(t), m_i(t), h_i(t), n_i(t), G_i^{\mathrm{S}}(t)\big). \tag{25}$$

In addition, in measurement of the largest Lyapunov exponent, we make the network an autonomous system by replacing the phase term ωt in the sinusoidal feedforward input (Eq. (8)) with a single variable p and including it into the solution vector $\mathbf{X}(t) = [X_1(t), \ldots, X_i(t), \ldots, X_N(t), p]$. Its dynamics is described by an additional equation

$$\frac{dp}{dt} = \omega. \tag{26}$$

More details about the calculation of Lyapunov exponents for different types of networks will be published elsewhere (Sun et al., submitted).

The largest Lyapunov exponents obtained by using both the regular and library methods for coupling strength S ranging from 0.2 to 0.4 mS/cm^2 over a long time interval of $T = 2^{16} = 65536$ ms are shown in Fig. 8. The regular method reveals that there is a chaotic regime of the network in $0.245 \lesssim S \lesssim 0.318$ mS/cm^2 since the largest Lyapunov exponent is positive at most of data points in this range. The left part ($0.2 \lesssim S \lesssim 0.245$ mS/cm^2) and the right part ($0.318 \lesssim S \lesssim 0.4$ mS/cm^2) correspond to the phase-locked state and

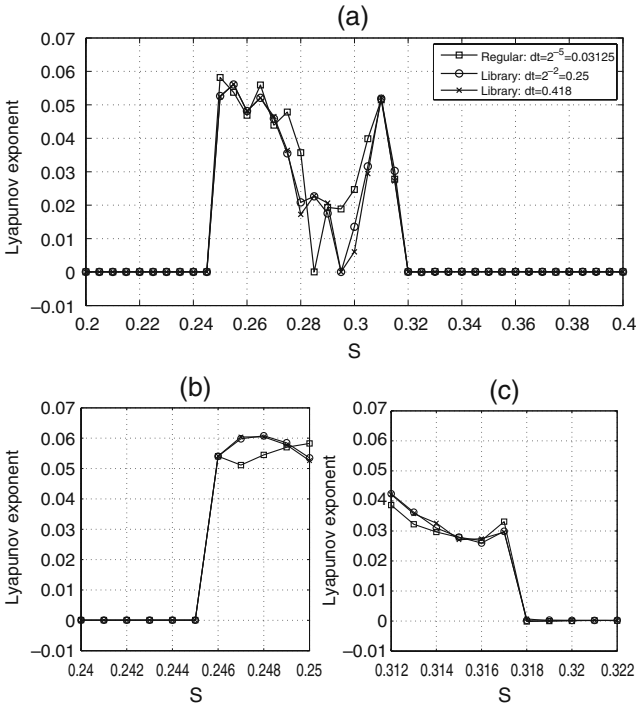

Fig. 8 (**a**): Largest Lyapunov exponent of the network versus the coupling strengths S. The squares correspond to the result using the regular method with time step ($\Delta t = 0.03125$ ms); the *circles* represent the one computed with much larger time step ($\Delta t = 0.25$ ms) by using the library method, and the crosses are the solution computed for the maximum time step ($\Delta t = 0.418$ ms) with the library method. (**b**): A fine scanning result on $[0.24, 0.25]$ mS/cm^2. (**c**): A fine scanning result on $[0.312, 0.322]$ mS/cm^2. We note that the library method with large time steps can capture the chaotic regime as well as the regular method does with a small time step. The total run time for following the trajectories is sufficiently long (65536 ms) in order to obtain convergent results for the largest Lyapunov exponent

nearly synchronous state, respectively. For these non-chaotic systems, the largest Lyapunov exponent is zero since the system is autonomous. As seen in Fig. 8, the library method can capture the chaotic regime well and the boundaries of the parameters that separate chaotic and nonchaotic regimes are consistent with the result of the regular method, although the values of the largest Lyapunov exponent in the chaotic regime are different between two methods.

In summary, the library method can capture the chaotic dynamics well, even when using a much larger time step compared to the regular method. This success of simulating chaotic dynamics presents strong evidence for the robustness of this numerical I&F reduction of HH neurons to study statistical long time dynamics of HH neuronal networks.

5.4 Firing rate and ISI distribution tests

In many applications, it is not necessary to resolve every single trajectory of all neurons in the system. For example, many real experiments (Koch 1999) only record the statistical properties of subpopulation of neurons (or the entire system), such as firing rate and ISI distributions. For example, an experiment involving firing rate statistics may only be concerned with the ISI distribution aggregated for many neurons in the system over a long time. To numerically model these experiments, we only need to obtain an accurate ISI distribution. In this case, the statistical properties of the network can be accurately resolved with much less computational effort by using the library method (with a large time step) than it would take to accurately resolve every neuronal trajectory with the regular method.

To further demonstrate the accuracy of the library method, we compare the statistical results of firing rate and ISI distribution with those obtained by the regular method. Figure 9(a) shows the average firing rate R, which is the number of firing events per neuron per second, for different values of coupling strength S in the test network as before. In the phase-locked regime ($0.2 \lesssim S \lesssim 0.245$ mS/cm^2), the firing rate stays exactly at the same value 50 spikes/s due to the fact that the dynamics of each neuron is phase locked to the sinusoidal feedforward input with oscillation angular frequency $\omega = 50$ Hz. Once the network enters the

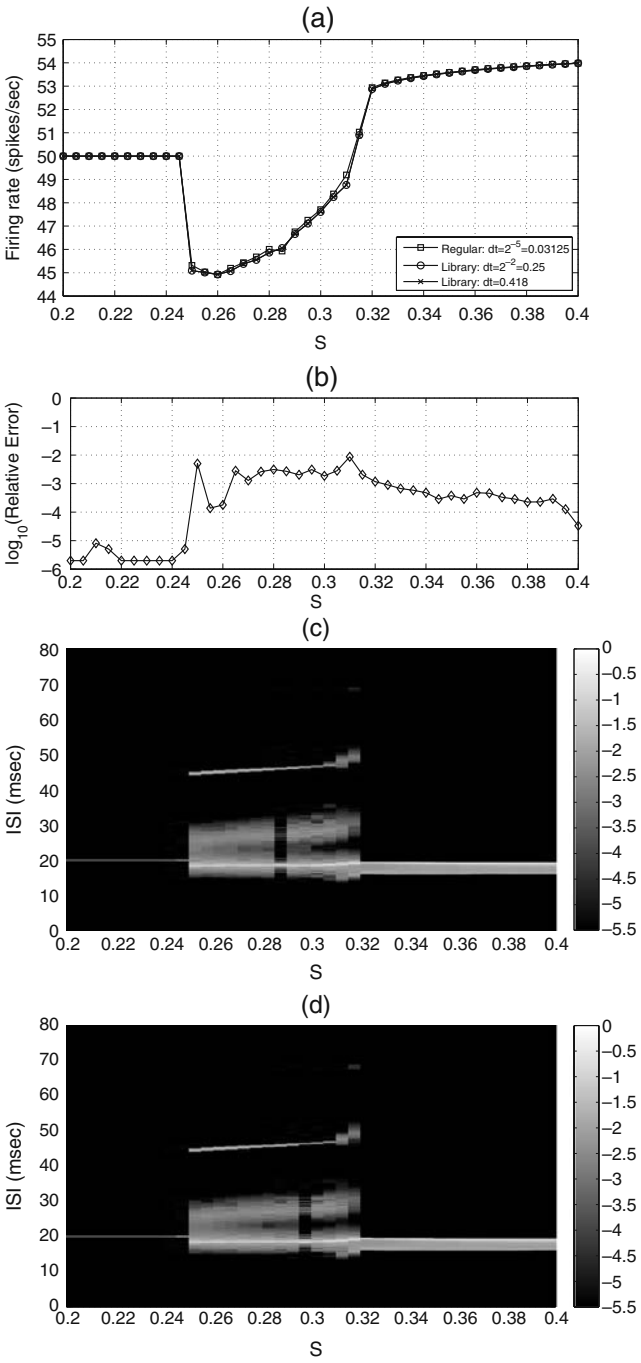

Fig. 9 (**a**): Average firing rate versus the coupling strength S. The *squares* correspond to the result using the regular method with time step ($\Delta t = 0.03125$ ms); the *circles* represent the one computed with much larger time step ($\Delta t = 0.25$ ms) by using the library method, and the *crosses* are the solution computed for the maximum time step ($\Delta t = 0.418$ ms) with the library method. (**b**): The relative error in the average firing rate between the library method on maximum time step ($\Delta t = 0.418$ ms) and the regular method on small time step ($\Delta t = 0.03125$ ms) versus S. (**c**): The ISI histograms computed with $\Delta t = 0.03125$ ms using the regular method versus S. Note that the probability density is represented by different gray scales as shown on the right of the figure. (**d**): The ISI histograms computed with $\Delta t = 0.418$ ms using the library method versus S. Note that we take the logarithm of the ISI histograms in (**c**) and (**d**) for clarity. The ISI data is binned into 0.1 ms time bins. The total run time is 65536 ms

chaotic regime $0.245 \lesssim S \lesssim 0.318$ mS/cm^2, the firing rate abruptly drops down to about 45 spikes/s and stays around there until $S \approx 0.265$ mS/cm^2, then grows as S increases. When the network becomes nearly synchronous ($0.318 \lesssim S \lesssim 0.4$ mS/cm^2), the firing rate again abruptly changes to about 53 spikes/s and then increases slowly.

Figure 9(b) shows the relative error in the average firing rate between the library method for large time steps ($\Delta t = 0.418$ ms) and the regular method for small time steps ($\Delta t = 0.03125$ ms), which is defined as follows:

$$E^{\mathrm{R}} = \left(R^{\mathrm{regular}} - R^{\mathrm{library}}\right)/R^{\mathrm{regular}}. \tag{27}$$

The library method can achieve more than 2 digits of accuracy using time steps 10 times larger than those used by the regular method for all values of S.

In Fig. 9(c) and (d), we plot the logarithm of the ISI histograms by using the regular method with small time steps ($\Delta t = 0.03125$ ms) and the library method with large time steps ($\Delta t = 0.418$ ms), respectively. The ISI data is binned into 0.1 ms time bins. Both of the results are consistent with each other except at a few points of S inside the chaotic regime. When the network is in the phase-locked regime ($0.2 \lesssim S \lesssim 0.245$ mS/cm^2), the ISI stays exactly at the same value 20 ms. In the nearly synchronous state ($0.318 \lesssim S \lesssim 0.4$ mS/cm^2), the ISI spreads over one band from about 15 to 20 ms. In the chaotic regime ($0.245 \lesssim S \lesssim 0.318$ mS/cm^2), the ISI distribution exhibits complicated structures of multiple bands, reflecting the chaotic nature of the dynamics.

5.5 Extensions of network simulations

To extend the applications of the library method, we test another two models. In the first one, we consider an all-to-all connected network of 100 excitatory neurons, which is driven by a feedforward input (Eqs. (9) and (10)) of a particular realization of a Poisson process with the rate $\omega = 50$ Hz, like the model used in Fig. 1. We find that the library method performs well in capturing the different dynamic regimes and computing the average firing rate. The library method can achieve not only more than 2 digits of accuracy in the average firing rate using time steps 10 times larger than those used by the regular method, but also good agreement between the largest Lyapunov exponents obtained by using the regular and library methods. In Fig. 10, we show the systematic scanning results between two methods for the model with the coupling strength S ranging from 0.025 to 1.0 mS/cm^2 over a long time interval of $T = 65536$ ms. The results also reveal three typical

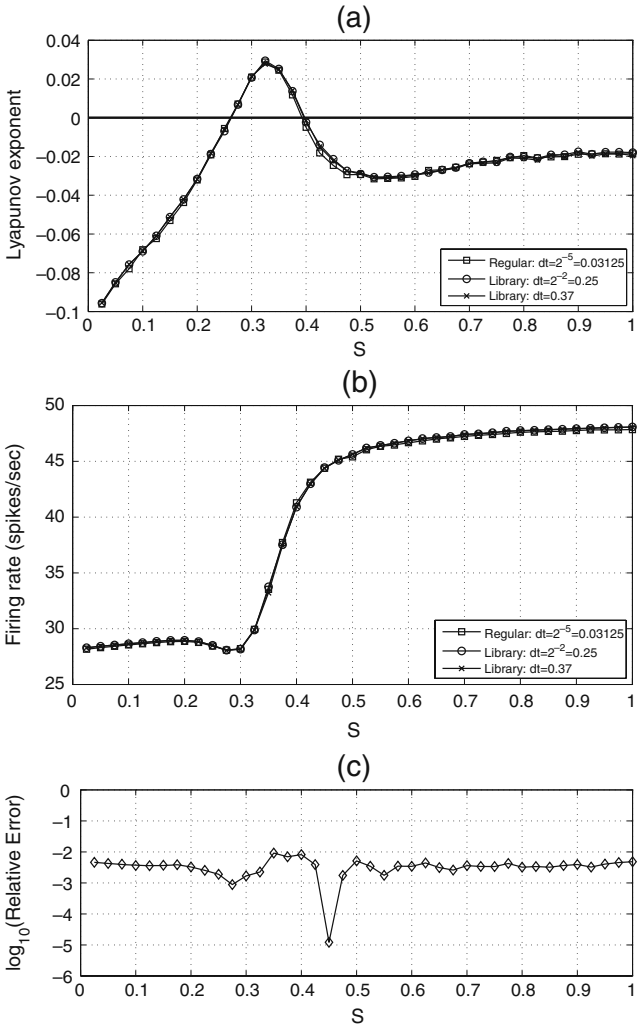

Fig. 10 The comparison results of an all-to-all connected network of 100 excitatory neurons driven by the feedforward input of a particular realization of a Poisson process with the rate $\omega = 50$ Hz. (**a**): Largest Lyapunov exponent versus the coupling strengths S. The *squares* correspond to the result using the regular method with time step ($\Delta t = 0.03125$ ms); the *circles* represent the one computed with much larger time step ($\Delta t = 0.25$ ms) by using the library method, and the crosses are the solution computed for the maximum time step ($\Delta t = 0.37$ ms) with the library method. We note that the library method with large time steps can capture the chaotic regime as well as the regular method does with a small time step. (**b**): Average firing rate versus the coupling strength S. (**c**): The relative error in the average firing rate between the library method on maximum time step ($\Delta t = 0.37$ ms) and the regular method on small time step ($\Delta t = 0.03125$ ms) versus S. The total run time is 65536 ms

dynamical regimes—an asynchronous, a chaotic, and a synchronous regime. The chaotic regime of the network exists in $0.262 \lesssim S \lesssim 0.395$ mS/cm^2 in the sense that the largest Lyapunov exponent is positive in this range. The left part ($0.025 \lesssim S \lesssim 0.262$ mS/cm^2) and the right part

($0.395 \lesssim S \lesssim 1.0$ mS/cm^2) correspond to the asynchronous state and synchronous state, respectively.

In the second extension test, we incorporate inhibitory neurons into the network. For each neuron, its input current term I_i^{input} is given by

$$I_i^{\text{input}} = -\sum_{\text{Q}} G_i^{\text{Q}}(t)\left(V_i(t) - V_{\text{G}}^{\text{Q}}\right), \tag{28}$$

where $G_i^{\text{Q}}(t)$ are the conductances with the index Q running over the types of conductances used, i.e., inhibitory and excitatory, and V_{G}^{Q} are their corresponding reversal potentials (see Appendix). The dynamics of $G_i^{\text{Q}}(t)$ are governed by

$$\frac{d}{dt}G_i^{\text{Q}}(t) = -\frac{G_i^{\text{Q}}(t)}{\sigma_{\text{r}}^{\text{Q}}} + \widetilde{G}_i^{\text{Q}}(t), \tag{29}$$

$$\frac{d}{dt}\widetilde{G}_i^{\text{Q}}(t) = -\frac{\widetilde{G}_i^{\text{Q}}(t)}{\sigma_{\text{d}}^{\text{Q}}} + \sum_{j\neq i}\sum_{k} S_{i,j}^{\text{Q}}\delta\left(t - T_{j,k}^{\text{S}}\right) + \sum_{k} F_i^{\text{Q}}\delta\left(t - T_{i,k}^{\text{F}}\right), \tag{30}$$

which are similar to Eqs. (11) and (12). Here, we consider an all-to-all connected network of 80 excitatory neurons and 20 inhibitory neurons. The stochastic feedforward input is generated by a particular realization of a Poisson process with the input rate $\omega = 50$ Hz and other parameters are given in Appendix.

In particular, we fix the coupling strength for inhibitory (excitatory) synapses onto excitatory (inhibitory) neurons $S^{\text{EI}} = S^{\text{IE}} = 0.1$ mS/cm^2 and vary the recurrent excitatory coupling strength S^{EE} ranging from 0.025 to 1.0mS/cm^2 to perform two systematic scanning tests for two different values of recurrent inhibitory coupling strength $S^{\text{II}} = 0.1$ and 0.2 mS/cm^2, respectively. We find again that there is a chaotic regime in both tests (Figs. 11 and 12). As shown in Fig. 11(a), for the case of $S^{\text{II}} = 0.1$ mS/cm^2, the chaotic regime exists in $0.224 \lesssim S^{\text{EE}} \lesssim 0.447$ mS/cm^2 since the largest Lyapunov exponent is positive in this range; but for $S^{\text{II}} = 0.2$ mS/cm^2 shown in Fig. 12(a), the chaotic regime exists for $0.025 \lesssim S^{\text{EE}} \lesssim 0.411$ mS/cm^2. We note that the library method performs well in capturing the different dynamic regimes. The average firing rates for both cases are monotonically increasing as S^{EE} increases (Figs. 11(b) and 12(b)). Again, the library method can achieve more than 2 digits of accuracy in the average firing rate using time steps 10 times larger than those used by the regular method (Figs. 11(c) and 12(c)).

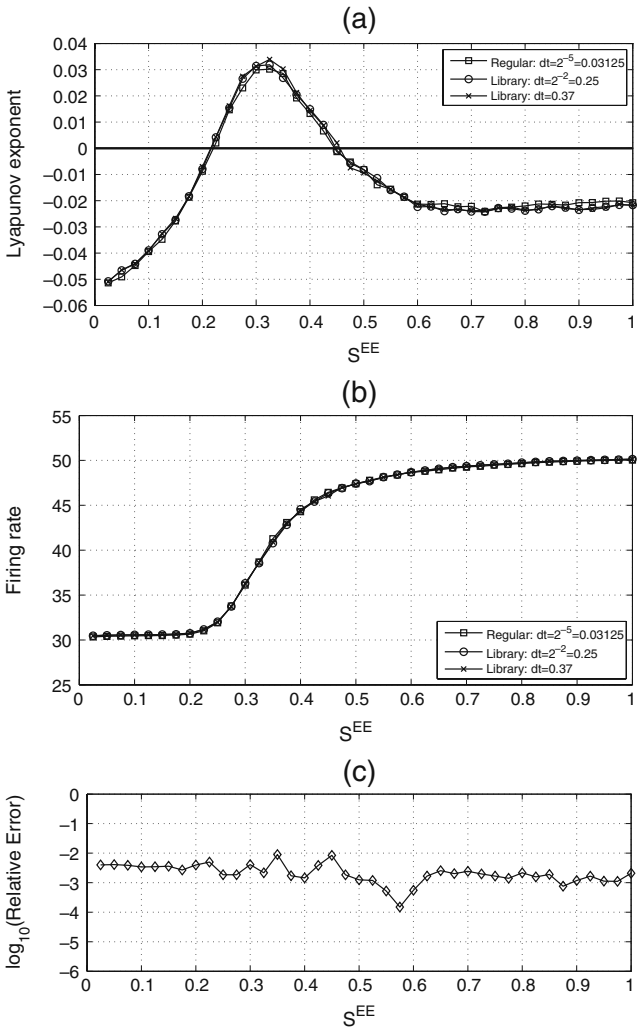

Fig. 11 The comparison results of an all-to-all connected network of 80 excitatory neurons and 20 inhibitory neurons driven by the feedforward input of a particular realization of a Poisson process with the rate $\omega = 50$ Hz. The value of recurrent inhibitory coupling strength $S^{\text{II}} = 0.1$ mS/cm^2. (**a**): Largest Lyapunov exponent versus the coupling strengths S^{EE}. The *squares* correspond to the result using the regular method with time step ($\Delta t = 0.03125$ ms); the *circles* represent the one computed with much larger time step ($\Delta t = 0.25$ ms) by using the library method, and the crosses are the solution computed for the maximum time step ($\Delta t = 0.37$ ms) with the library method. (**b**): Average firing rate versus the coupling strength S^{EE}. (**c**): The relative error in the average firing rate between the library method on maximum time step ($\Delta t = 0.37$ ms) and the regular method on small time step ($\Delta t = 0.03125$ ms) versus S^{EE}. The total run time is 65536 ms

5.6 Type II dynamics of individual HH neurons

Since our numerical I&F reduction of the HH neuron keeps the original HH dynamics (including gating variable dynamics) below the threshold and uses the intermediate replica for the action potentials, we can still capture some type II dynamical property of individual

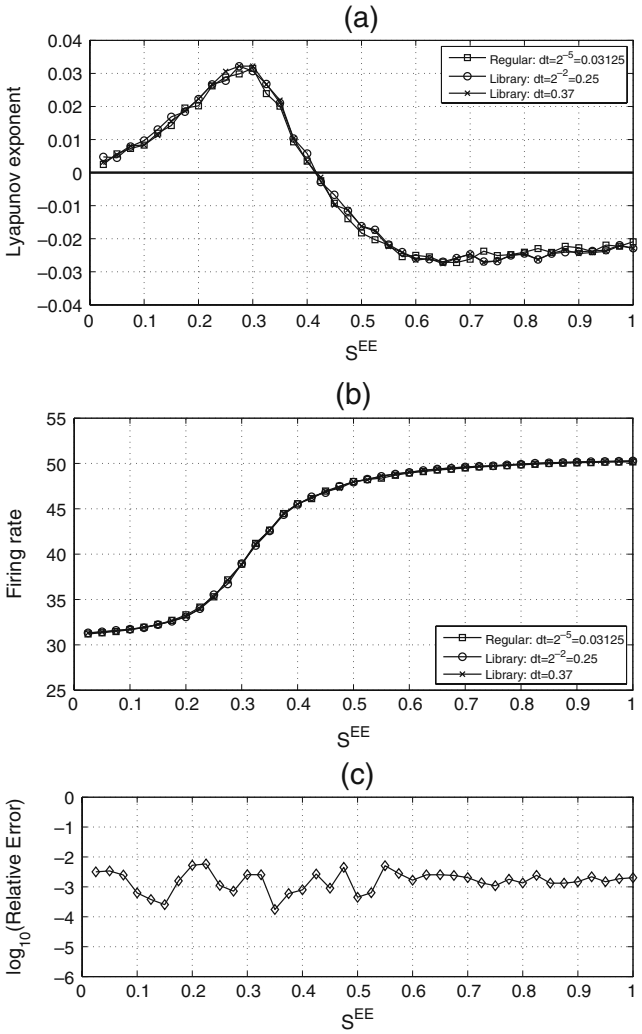

Fig. 12 The comparison results of the same network as the one in Fig. 11. The only difference is $S^{\mathrm{II}} = 0.2$ mS/cm^2. (**a**): Largest Lyapunov exponent versus the coupling strengths S^{EE}. (**b**): Average firing rate versus the coupling strength S^{EE}. (**c**): The relative error in the average firing rate between the library method on maximum time step ($\Delta t = 0.37$ ms) and the regular method on small time step ($\Delta t = 0.03125$ ms) versus S^{EE}. The total run time is 65536 ms

HH neurons, such as the phase advance and delay pattern in the phase-resetting curve (PRC) (Winfree 2001). The PRC describes the phase shift of the oscillation in response to a perturbing pulse of variable amplitude at each phase of the oscillation and can reveal useful information about the complex dynamical mechanisms underlying the periodic activity with hidden variables not accessible to observation (Galan et al. 2005). The perturbation is assumed to be sufficiently weak that its effect on the amplitude and intrinsic period is negligible. For the repetitively firing neurons (on a stable limit cycle), a small current pulse delays or advances the next spike without changing its shape or average firing frequency.

The PRC can be calculated by studying how the time of the next spike is shifted as a function of the pulse time t relative to the previous spike:

$$\Delta(\phi, I_0) = \frac{T - T'(\phi, I_0)}{T}, \tag{31}$$

where T is the natural period, $T'(\phi)$ is the time of the spike given a stimulus at time t after the last spike, and $\phi = \frac{2\pi t}{T} \in [0, 2\pi)$ is the instantaneous phase. The parameter I_0 is the magnitude of the current pulse.

In Fig. 13(a) and (b), we show the PRC by using both of the regular method and the library method under a current pulse with magnitude $I_0 = 0.2\ \mu$A/cm^2 for a short duration 0.5 ms and a longer duration 4.0 ms, respectively. The pulse is superimposed on the constant current $I = 7.5\ \mu$A/cm^2. We note that the library method with a large time step ($\Delta t = 0.25$ ms) can produce the PRC as accurate as that from the regular method using a small time step ($\Delta t = 0.03125$ ms). Moreover, as shown in Fig. 13, the PRC can be positive over certain phases and negative over others since a small positive perturbation of the membrane potential may delay or advance the next spike in a type II neuron, depending on the phase at which the pulse is delivered.

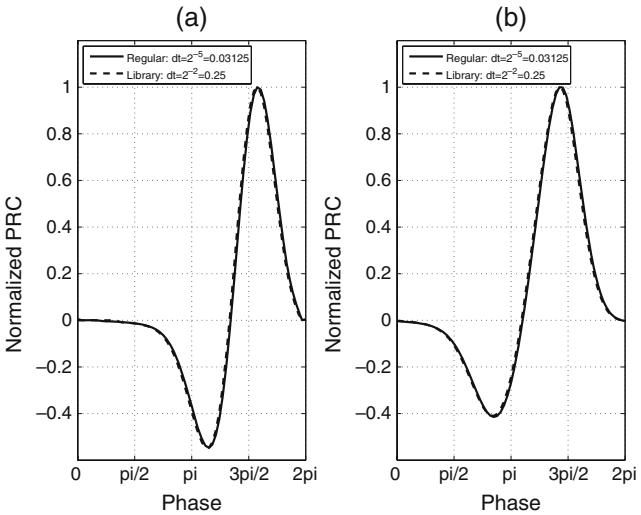

Fig. 13 (**a**): The PRC under a current pulse for a short duration 0.5 ms. (**b**): The PRC under a current pulse for a longer duration 4.0 ms. The pulse with magnitude $I_0 = 0.2\ \mu$A/cm^2 is superimposed on the constant current $I = 7.5\ \mu$A/cm^2. The *solid line* corresponds to the result using the regular method with time step ($\Delta t = 0.03125$ ms) and the *dashed line* represents the one computed with much larger time step ($\Delta t = 0.25$ ms) by using the library method

We point out that there is another type-II behavior in a single HH neuron—as a function of a constant input current I, the HH model undergoes a sudden jump from the fixed point solution (the membrane potential at rest) to repetitive firing with nonzero frequency at $I = I_H$ due to a subcritical Hopf bifurcation (Rinzel and Ermentrout 1998). As shown in Fig. 14(a), the library method with a large time step can capture this feature as well as the regular method does using a smaller time step.

Moreover, just below the Hopf bifurcation, there is a coexistence of a stable fixed point with a stable limit cycle and their basins of attraction being separated by an unstable limit cycle. The stable and unstable limit cycles are created via a saddle-node bifurcation at $I = I_S < I_H$ (Rinzel and Ermentrout 1998). As demonstrated in Roa et al. (2007), if the fixed point at zero current is perturbed by the application of a constant current near but below the critical current I_S, a number of spikes may appear before the HH neuron returns to the new resting state, as shown in Fig. 14(b). The number of spikes in the transient depends on how close the current is to I_S. The slow passage effect may exhibit very long transients in response to the perturbed current. However, since our library is built with the data obtained in the dynamic regime of stable repetitive firing, it is difficult for the library method to capture this transient effect near the saddle-node bifurcation, as evidenced in Fig. 14(b).

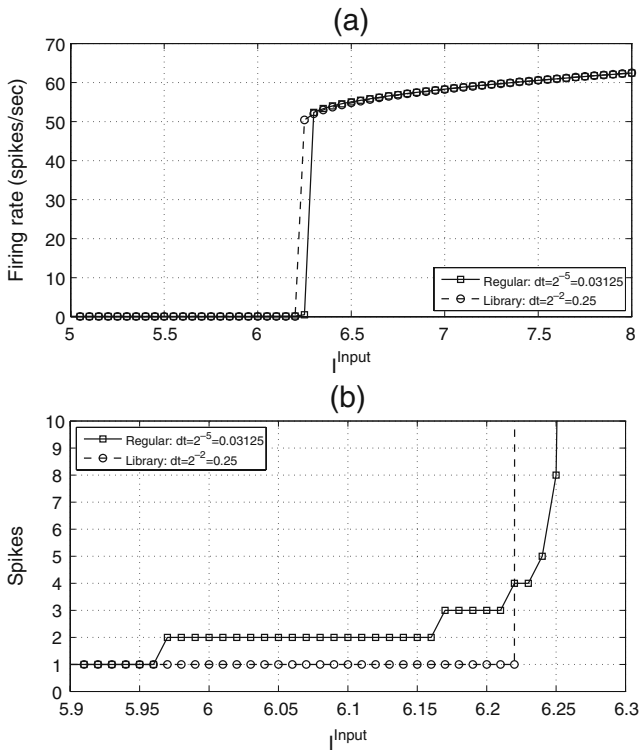

Fig. 14 (**a**): The firing rate versus the input current I^{input}. Both of the regular and the library methods can capture the Hopf bifurcation. (**b**): The number of spikes during the transient period (see text). Note that the difference in the label of y-axis between (**a**) and (**b**). The *solid line with squares* corresponds to the result using the regular method with time step ($\Delta t = 0.03125$ ms) and the *dashed line with circles* represents the one computed with much larger time step ($\Delta t = 0.25$ ms) by using the library method. The total run time is 16384 ms

5.7 Computational efficiency

To demonstrate the efficiency of library method, we compare the maximum time steps allowed for different schemes in both the regular method and the library method before the schemes become unstable numerically. Table 2 shows these maximum time steps for simulating the network driven by the continuous sinusoidal feedforward input in Section 5.1 with the strong coupling strength $S = 0.4$ mS/cm^2. In each method (regular or library), the maximum time steps for Euler and RK2 schemes are almost same. We use a linear interpolation to find the spike times with both schemes. RK4 scheme with a cubic Hermite interpolation for estimating spike times can allow a larger maximum time step than Euler and RK2 schemes. We emphasize that the large time steps using the library method can break the stability requirement of the regular method. As shown in Table 2, the maximum step for the regular method is $\Delta t = 0.094$ ms. At the time step resolution used in the library method, such as the maximum step $\Delta t = 0.418$ ms, the regular method becomes unstable and fails to run due to the stability requirement.

The advantage of library method is that it allows one to use much larger maximum time steps (4 times larger), in comparison with the regular method. Moreover, as we show in Fig. 6 (the return map and the power spectrum) and Fig. 9 (average firing rate and ISI histograms), the library method can achieve more than 2 digits accuracy with maximum time step $\Delta t = 0.418$ ms (10 times larger) as the results of the regular method on $\Delta t = 0.03125$ ms. In Figs. 8, 10–12, it is also demonstrated that the library method can robustly capture the chaotic dynamics using a large time step.

Table 2 Maximum time step (ms) allowed for different methods

Method	Regular method	Library method
Euler	7.40×10^{-2}	3.10×10^{-1}
RK2	7.50×10^{-2}	3.19×10^{-1}
RK4	9.40×10^{-2}	4.18×10^{-1}

We run the code on a Linux platform using an Intel Pentium IV 2.3 MHz processor. For the systematic scanning test shown in Figs. 8 and 9, it takes the library method about 5 hours to run all of 41 points of the synaptic coupling strength S, ranging from 0.2 to 0.4mS/cm^2 with the maximum time step $\Delta t = 0.418$ ms, whereas the regular method with $\Delta t = 0.03125$ ms requires nearly 26 h to resolve the computation or nearly 15 h with the maximum time step $\Delta t = 0.094$ ms. Note that when we use large time steps to run the code, it requires extra computation to take into account the causality of spiking events within a single time step via the spike-spike correction (Rangan and Cai 2007). Therefore, the computational speed is not increased linearly as we increase the time step. However, we emphasize that the library method can achieve more than 2 digits accuracy with more than 5 times computational speedup compared to the regular method as we show above.

We make further comments about computational efficiency. First, we comment on the comparison between our library method and the higher order implicit methods. In Fig. 2 we have an observation: for the action potential profile we isolated in the time interval $T^{ref} = 3.4375$ ms, it requires at least 7 to 10 data points to resolve the spike. This means that even we choose any implicit method, it still needs the time step to be sufficiently small to resolve the spike, say about 0.35 to 0.5 ms. But this time step is comparable to the maximum time step with the library method ($\Delta t = 0.418$ ms). Moreover, in principle, the explicit RK scheme employed in our library method should be faster than the implicit method in one time step because the implicit method requires computing the Jacobian of the dynamical equations of the entire system and solving them iteratively. If the implicit method is chosen with adaptive time steps, it may work well for a single neuron or a few neurons. However, we emphasize that since here we deal with network simulations, neurons may fire randomly over time and it is difficult to make adaptive time steps suitable for all neurons' trajectories at the same time since they are usually at different dynamical states. Second, it is quite involved to incorporate the spike-spike correction algorithm for handling the causality of spiking events in one time step into the implicit method.

Finally we address the issue of whether we can use a simpler library method, which ignores the difference in the reset values induced by the different values of the input current and uses fixed values for each of the reset values $V^{\rm re}$, $m^{\rm re}$, $h^{\rm re}$, $n^{\rm re}$. For simplicity, we take a data set of $V^{\rm re} = -72$ mV, $m^{\rm re} = 0.05, h^{\rm re} = 0.15, n^{\rm re} = 0.65$. With these fixed "naive" reset values, we perform the same systematic scanning test as the one in Figs. 8 and 9. As shown in Fig. 15(a), the borders between the chaotic regime and the other two regimes are mismatched, respectively, between the naive library method and the regular method. There is obvious disagreement in the average firing rate shown in Fig. 15(b) between the naive library method and the regular method. Moreover, by comparing Fig. 15(c) with Fig. 9(b), we find that the accuracy in the average firing rate changed from 2 digits to only 1 digit in most of the chaotic regime and from 3 digits to 2 digits in the nearly synchronous regime. On the other hand, we emphasize that it does

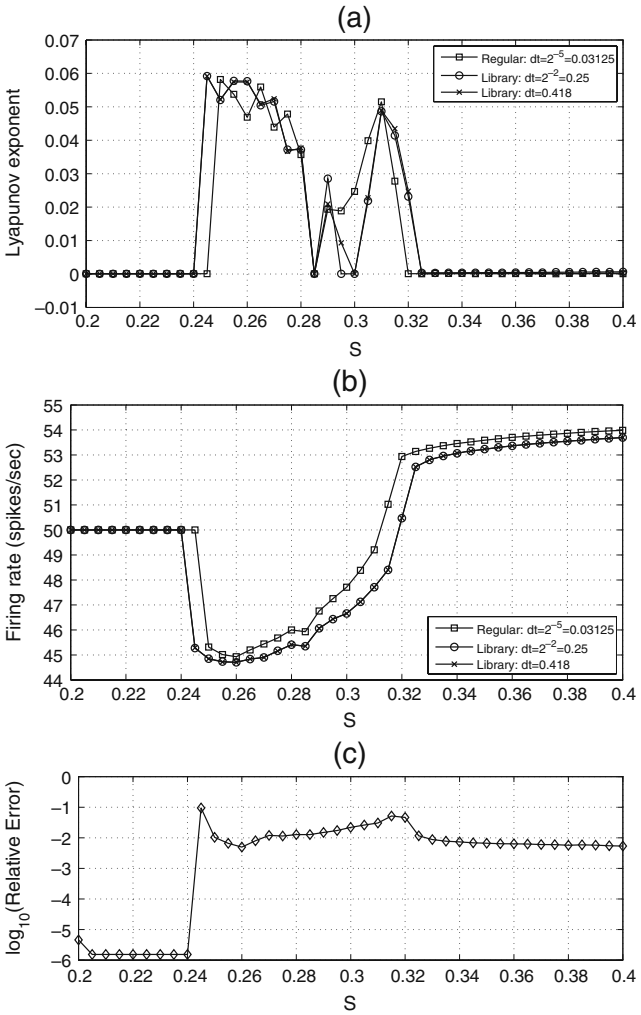

Fig. 15 The comparison results of the same network as the one in Fig. 8 between the naive library method (see text) and the regular method. All the parameters and the labels are the same as in Fig. 8. (**a**): Largest Lyapunov exponent versus the coupling strengths S. (**b**): Average firing rate versus the coupling strength S. (**c**): The relative error in the average firing rate between the naive library method using maximum time step ($\Delta t = 0.418$ ms) and the regular method using small time step ($\Delta t = 0.03125$ ms) versus S. The total run time is 65536 ms

not cost much more computation for our method using the elaborate data library than the naive library method because we only need to perform extra computation (e.g. the linear interpolations) to obtain the reset values from the data library. The number of operations for using the library only depends on the total number of spikes during the network simulation, instead of the total number of time steps. Therefore, there is a clear advantage to use our data library to keep the difference in the reset values to obtain more accurate results.

6 Conclusion

We have presented a new method that reduces the dynamics of HH neurons to that of I&F neurons numerically. Our method not only retains most of the detailed properties of the original HH neurons in a network, but also is more efficient than fully resolving the HH equations. We overcome the stability restriction associated with firing events by using a precomputed high resolution data library. This allows us to use much larger time steps for evolving the neuronal trajectories even if the conductances are high and the HH equations are stiff. By using this method, we can collect accurate statistical information with much lower computational cost.

We remark that the idea of our library method can be extended to more complicated HH-type equations which may contain more ionic channel currents. We can also build the intermediate replica for these channel gating variables into the library and use them with the same interpolation technique during the refractory period.

The results presented in this article have been obtained for networks of $N = 100$ neurons, but increasing the size of the network does not change our conclusions. In addition to deterministic models using the continuous type of feedforward input as shown in last section, our method performs well when we use stochastic feedforward inputs and add inhibitory neurons into the network.

We emphasize that the library method can be used to describe the chaotic regime of network dynamics, as signified by a positive largest Lyapunov exponent. Moreover, there is no numerical convergence for both the regular and library methods in this chaotic regime.

When we use the method to simulate the neuronal network, we also employ the spike-spike corrections procedure (Rangan and Cai 2007), which allows us to accurately estimate spiking sequences for strongly coupled neurons without the usual time step restriction imposed by synaptic interactions in modified Runge–Kutta methods (Hansel et al. 1998; Shelley and Tao 2001).

Finally, we mention that the library method can be practicable for simulating large-scale neuronal networks by incorporating a clustering procedure of firing events in networks to take advantage of localized architectures, such as spatial scales of strong local interactions, which are often present in large-scale computational models—for example, those of the primary visual cortex (Rangan and Cai 2007).

Acknowledgements The work was supported by NSF grant DMS-0506396 and a grant from the Swartz foundation.

Appendix: Parameter values for the Hodgkin–Huxley equations

Parameter values or ranges and function definitions of the Hodgkin–Huxley model are as follows (Dayan and Abbott 2001):

$$G_{\mathrm{Na}} = 120\ \mathrm{mS/cm^2},\quad V_{\mathrm{Na}} = 50\ \mathrm{mV},$$

$$G_{\mathrm{K}} = 36\ \mathrm{mS/cm^2},\quad V_{\mathrm{K}} = -77\ \mathrm{mV},$$

$$G_{\mathrm{L}} = 0.3\ \mathrm{mS/cm^2},\quad V_{\mathrm{L}} = -54.387\ \mathrm{mV},$$

$$C = 1\ \mu\mathrm{F/cm^2},\quad V_{\mathrm{G}}^{\mathrm{E}} = 0\ \mathrm{mV},\quad V_{\mathrm{G}}^{\mathrm{I}} = -80\ \mathrm{mV},$$

$$F^{\mathrm{E}} = 0.05 \sim 0.1\ \mathrm{mS/cm^2},\quad S^{\mathrm{E}} = 0.05 \sim 1.0\ \mathrm{mS/cm^2},$$

$$F^{\mathrm{I}} = 0.01 \sim 0.05\ \mathrm{mS/cm^2},\quad S^{\mathrm{I}} = 0.05 \sim 1.0\ \mathrm{mS/cm^2},$$

$$\sigma_{\mathrm{r}}^{\mathrm{E}} = 0.5\ \mathrm{ms},\quad \sigma_{\mathrm{d}}^{\mathrm{E}} = 3.0\ \mathrm{ms},$$

$$\sigma_{\mathrm{r}}^{\mathrm{I}} = 0.5\ \mathrm{ms},\quad \sigma_{\mathrm{d}}^{\mathrm{I}} = 7.0\ \mathrm{ms},$$

$$\alpha_m(V) = 0.1(V+40)/(1-\exp(-(V+40)/10)),$$

$$\beta_m(V) = 4\exp(-(V+65)/18),$$

$$\alpha_h(V) = 0.07\exp(-(V+65)/20),$$

$$\beta_h(V) = 1/(1+\exp(-(35+V)/10)),$$

$$\alpha_n(V) = 0.01(V+55)/(1-\exp(-(V+55)/10)),$$

$$\beta_n(V) = 0.125\exp(-(V+65)/80).$$

References

Aihara, K., & Matsumoto, G. (1986). Chaotic oscillations and bifurcations in squid giant axons. In *Chaos, nonlinear science: Theory and applications*. Manchester: Manchester University Press.

Cai, D., Rangan, A. V., & McLaughlin, D. W. (2005). Architectural and synaptic mechanisms underlying coherent spontaneous activity in v1. *Proceedings of the National Academy of Sciences, USA, 102*, 5868–5873.

Campbell, D., & Rose, H. (Eds.) (1983). Order in chaos. *Physica D, 7*, 1–362.

Dayan, P., & Abbott, L. F. (2001). *Theoretical neuroscience: Computational and mathematical modeling of neural systems*. Cambridge: MIT Press.

Galan, R. F., Ermentrout, G. B., & Urban, N. N. (2005). Efficient estimation of phase-resetting curves in real neurons and its significance for neural-network modeling. *Physical Review Letters, 94*, 158101–158104.

Gear, C. W. (1971). *Numerical initial value problems in ordinary differential equations*. Englewood Cliffs: Prentice Hall.

Guckenheimer, J., & Oliva, R. A. (2002). Chaos in the Hodgkin.Huxley model. *SIAM Journal on Applied Dynamical Systems, 1*, 105–114.

Hansel, D., Mato, G., Meunier, C., & Neltner, L. (1998). On numerical simulations of integrate-and-fire neural networks. *Neural Computation, 10*, 467–483.

Hansel, D., & Sompolinsky, H. (1992). Synchronization and computation in a chaotic neural network. *Physical Review Letters, 68*, 718–721.

Hansel, D., & Sompolinsky, H. (1996). Chaos and synchrony in a model of a hypercolumn in visual cortex. *Journal of Computational Neuroscience, 3*, 7–34.

Hodgkin, A. L., & Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology, 117*, 500–544.

Koch, C. (1999). *Biophysics of computation*. Oxford: Oxford University Press.

Kosmidis, E. K., & Pakdaman, K. (2003). An analysis of the reliability phenomenon in the FitzHugh-Nagumo model. *Journal of Computational Neuroscience, 14*, 5–22.

Lin, K. (2006). Entrainment and chaos in a pulse-driven Hodgkin–Huxley oscillator. *SIAM Journal on Applied Dynamical Systems, 5*, 179–204.

Mainen, Z., & Sejnowski, T. (1995). Reliability of spike timing in neocortical neurons. *Science, 268*, 1503–1506.

Mattia, M., & Del Giudice, P. (2000). Efficient event-driven simulation of large networks of spiking neurons and dynamical synapses. *Neural Computation, 12*, 2305–2329.

McLaughlin, D., Shapley, R., Shelley, M., & Wielaard, J. (2000). A neuronal network model of macaque primary visual cortex (V1): Orientation selectivity and dynamics in the input layer 4Cα. *Proceedings of the National Academy of Sciences, USA, 97*, 8087–8092.

Parker, T. S., & Chua, L. O. (1989). *Practical numerical algorithms for chaotic systems*. New York: Springer.

Rangan, A. V., & Cai, D. (2007). Fast numerical methods for simulating large-scale integrate-and-fire neuronal networks. *Journal of Computational Neuroscience, 22*, 81–100.

Rangan, A. V., Cai, D., & McLaughlin, D. W. (2005). Modeling the spatiotemporal cortical activity associated with the line-motion illusion in primary visual cortex. *Proceedings of the National Academy of Sciences, USA, 102*, 18793–18800.

Reutimann, J., Giugliano, M., & Fusi, S. (2003). Event-based simulation of spiking neurons with stochastic dynamics. *Neural Computation, 15*, 811–830.

Rinzel, I., & Ermentrout, G. B. (1998). Analysis of neuronal excitability (2nd ed.). In *Methods in neuronal modeling: From ions to networks*. Cambridge: MIT Press.

Roa, M. A. D., Copelli, M., Kinouchi, O., & Caticha, N. (2007). Scaling law for the transient behavior of type-II neuron models. *Physical Review E, 75*, 021911.

Rudolph, M., & Destexhe, A. (2007). How much can we trust neural simulation strategies? *Neurocomputing, 70*, 1966–1969.

Schuster, H. G., & Just, W. (2005). *Deterministic chaos*. Weinheim: Wiley-VCH.

Shelley, M. J., & Tao, L. (2001). Efficient and accurate time-stepping schemes for integrate-and-fire neuronal networks. *Journal of Computational Neuroscience, 11*, 111–119.

Somers, D., Nelson, S., & Sur, M. (1995). An emergent model of orientation selectivity in cat visual cortical simple cells. *Journal of Neuroscience, 15*, 5448–5465.

Troyer, T., Krukowski, A., Priebe, N., & Miller, K. (1998). Contrast invariant orientation tuning in cat visual cortex with feedforward tuning and correlation based intracortical connectivity. *Journal of Neuroscience, 18*, 5908–5927.

Winfree, A. T. (2001). *The geometry of biological time*. New York: Springer.