

Multi-Level Monte Carlo Algorithm for Integrate-Fire Neural Network

Zhuocheng Xiao¹, Kevin Lin^{1,*}

¹ Department of Mathematics, University of Arizona

* klin@math.arizona.edu

Abstract

Multi-Level Monte Carlo algorithm is much more efficient to simulate SDEs, comparing with MC algorithm. The core idea for MLMC is using information from different levels of numerical results to constraint mean-square-error (MSE), so that we gain a better estimation on the expectation. For coarser discretization, we evolve more samples, and less samples for finer discretization. Comparing with MC, when we try to get the same order of accuracy of estimation, we could use the MSE produced by the finest grid and the largest group of samples, which has shown huge advantages on time cost. However, when we try to apply MLMC to I-F Neuron network, we need to face several questions which people didn't know the answers before, which drives us to write this work report...(need modification)

1 Introduction

Action potentials, which are the reactions of neurons to stimulation, play a critical role in neuron interaction by transmitting signals along connected neurons in central nervous system. Action potentials are also named as "neural spikes" or "firing events" due to the short millisecond-time-scales, comparing with the collective behavior of a group of neurons in several seconds. The "spike sequences" generated by a neural network are signals released to the network itself/other neural system, therefore usually viewed as the core study object of computational neuroscience. One can use spike sequences to describe the functions of a group of neurons, or examine relation between different brain areas, or event reconstruct a stimulus-driven network from the exact spike times.

As the central character of brain coding, the spike sequences should be captured by a successful numerical simulation of neural dynamics with satisfying time accuracy. A classical approximative model called Leaky-Integrate-Fire (LIF) Equations are commonly used in computation. A much simplified current-based LIF for a single neuron is like:

$$\frac{dv}{dt} = -g_L(v - V_r) + I, \quad (1a)$$

$$\tau \frac{dI}{dt} = -I + f \sum_{\mu} \delta(t - t_{\mu})$$

$$v(t^-) = V_T \Leftrightarrow v(t^+) = V_r. \quad (1b)$$

1a is a differential equation describing dynamics of the neuron's membrane potential and current, with jump boundary condition 1b (reset to 0 when v reach 1). Current I is driven by a stochastic δ -sequence μ which are the spikes received by this neuron. Therefore, such kind of LIF equations can be evolved by Monte Carlo method for large amount of samples. Monte Carlo path simulation methods are frequently employed in numerical simulations to gain statistics for a stochastic processes, especially stochastic differential equations(SDEs). However, even for the simplest stastic, expectation $\mathbb{E}(X)$, an estimation

with an $O(\epsilon)$ mean-square-error brings an $O(\epsilon)^{-3}$ time cost [1]. Thus, variance reduction methods are used extensively to cut down computation complexity to gain fast and accurate statistics. In 2008, M. Giles developed an multigrid method for variance reduction [1], which is called Multi-Level Monte Carlo algorithm (MLMC). Here, we are trying to apply this algorithm to evolving the dynamics of neural networks. As people are mostly using the classical Monte Carlo method (named as MC below) for the task, MLMC is capable to remarkably reduce the computational complexity from $O(\epsilon^3)$ to $O(\epsilon^2 \log(\epsilon))$, according to **Theorem 3.1** in [1].

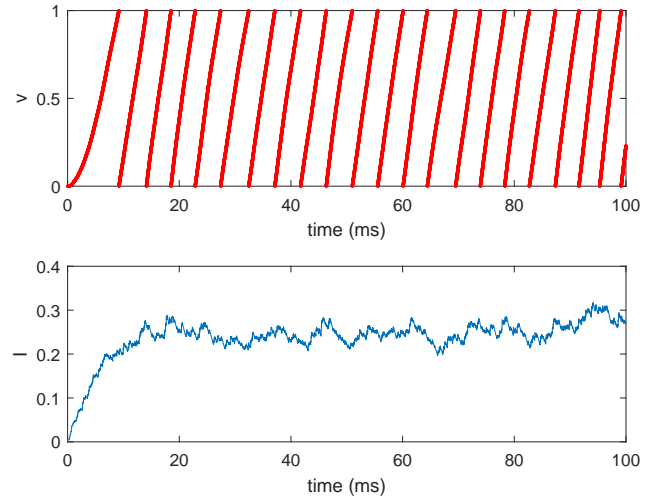


Figure 1. A 100-ms numerical sample for 1a, assuming input firing events μ to be Poisson. The upper bar for V and the lower bar for I . Driven by excitatory input firing events, V keeps crossing the boundary. The numerical result for $I(t)$ is just like a Ornstein-Uhlenbeck process, intuitively leading us to 5, i.e., the current jumps $\sum_{\mu} f \delta(t - t_{\mu})$ could be approximated by a Wiener process.

To utilize numerical solutions from different grids to cancel the errors introduced by numerical grids, we need them converging to an actual solution (if exists!) when the grid size $h \rightarrow 0$. This is pretty natural if the stochastic differential equations (SDE) are continuous and satisfy Lipschitz condition. In LIF equation, however, this is generally not true according to the jumping boundary condition 1b. There are plenty of researches on SDE with jump, but none of them was about a jump boundary condition. That's to say, if an error ϵ produced at some step, the time boundness or convergence for such $\epsilon(t)$ is never guaranteed by any classical numerical approximation theorem. Without any further investigation into error estimation for LIF eqns, we can't trust the numerical solution to **a single neuron**, not to mention MLMC method which is based on the former.

Such obstacle drives us to look into the convergence of numerical error for LIF equations in detail. In this article, we present a "physical proof" for almost sure convergence of membrane potential v when evolved by same/different time grid sizes. Based on such convergence, we can derive convergence for every Feed-Forward Network (FFN), i.e. we can safely use MLMC methods for FFNs. As for recurrent network (neural network with feed back), however, the situation is much more complicated. We argue that for weakly coupled networks which don't produce multiple firing events (MFEs) [3], a.s. convergence still exists. But for strongly coupled networks with large MFEs, a.s. convergence collapses and numerical results are extremely sensitive to time error for even a single firing event. In this case, MLMC method is no more valid when simulations don't converge even if taking grid size $h \rightarrow 0$.

2 Background

2.1 Simplified LIF Equations

The membrane potential and current dynamics of a general current-based LIF neural network consists of Excitatory & Inhibitory neurons with various connectivity can be described as:

$$\begin{aligned} C \frac{dv_j}{dt} &= -g_L(v_j - V_r) + (I_j^E - I_j^I), \\ \sigma_E \frac{dI_j^E}{dt} &= -I_j^E + \sum_{\mu_j^E} f_j^E \delta(t - t_{\mu_j^E}) + \sum_{i \neq j} \sum_{\mu_{ij}^E} s_{ij}^E \delta(t - t_{\mu_{ij}^E}), \\ \sigma_I \frac{dI_j^I}{dt} &= -I_j^I + \sum_{\mu_j^I} f_j^I \delta(t - t_{\mu_j^I}) + \sum_{i \neq j} \sum_{\mu_{ij}^I} s_{ij}^I \delta(t - t_{\mu_{ij}^I}). \end{aligned} \quad (2a)$$

$$\begin{aligned} v_j &\in (-\infty, V_T), \quad I_j^{E,I} \in [0, +\infty); \\ v_j(t^-) &= V_T \Leftrightarrow v_j(t^+) = V_r. \end{aligned} \quad (2b)$$

In 2a, $\mu_j^{E,I}$ represent Excitatory & inhibitory input spiking sequences, which can be assumed as Poission processes in many cases. So $\mu_j^{E,I}$ is the source of stochasticity for the system. On the other hand, matrices $[\mu]_{ij}^{E,I}$ represent Excitatory or inhibitory connections from neuron i to neuron j .

We will start from a single neuron in **Coupling Single**

Neurons, as the dynamics for a neural network could be so sophisticated. Consider the I-F eqn for a single neuron:

$$\begin{aligned} C \frac{dv}{dt} &= -g_L v + (I^E - I^I), \\ \sigma_E \frac{dI^E}{dt} &= -I^E + \sum_{\mu^E} f^E \delta(t - t_{\mu^E}), \\ \sigma_I \frac{dI^I}{dt} &= -I^I + \sum_{\mu^I} f^I \delta(t - t_{\mu^I}). \end{aligned} \quad (3a)$$

In 3a, we simply choose $C = 1, V_r = 0, V_T = 1, g_L = \frac{1}{20\text{ms}}$ and $\sigma_E = \sigma_I = 5\text{ms}$. Furthermore, we can simply assume the inhibitory current I^I is a constant for simplicity as firing dynamics is mainly determined by I^E like the case in Fig.1. Note that μ^E is supposed as Poission with frequency ν and could well-approximated by Gaussian processes when we take limit $f^E \rightarrow 0, \nu \rightarrow \infty$. Thus, we could rewrite 3a as:

$$\frac{dv}{dt} = -g_L v + (I^E - I^I), \quad (4a)$$

$$\sigma_E \frac{dI^E}{dt} = -I^E + f\nu + \frac{f^2\nu}{2} \frac{dW}{dt}. \quad (4b)$$

What's more, by taking $\sigma_E \rightarrow 0 \Rightarrow I^E = f\nu + \frac{f^2\nu}{2} \frac{dW}{dt}$, 4a and 4b are homogenized into a 1-variable SDE with jumping boundary condition, directly driven by a white noise:

$$\frac{dv}{dt} = -g_L v + (f\nu + \frac{f^2\nu}{2} \frac{dW}{dt} - I^I). \quad (5)$$

2.2 Couplings and Sensitivity

A valid numerical simulation for a sample from 5 requires convergent error. Consider sample v and a biased sample $u = v + \epsilon(t)$ and their coupling $P(v, u)$ which is a joint probability space for them, where both samples driven by the exactly same input. Therefore, finite coupling time and $\lim_{t \rightarrow \infty} |u - v| = 0$ leads to a valid numerical method.

For "real" solutions, as we have stated in **Introduction**, if in 5 v doesn't jump when hits the boundary V_T , $\epsilon(t) = \epsilon(0)e^{-g_L t}$ (Fig.2 left column). Think about u and v are receiving the same input $I + \frac{f^2\nu}{2} \frac{dW}{dt}$, where $f\nu - I^I = I$, we have:

$$\frac{du}{dt} = -g_L u + I + \frac{f^2\nu}{2} \frac{dW}{dt}, \quad (6a)$$

$$\frac{dv}{dt} = -g_L v + I + \frac{f^2\nu}{2} \frac{dW}{dt}. \quad (6b)$$

which leads to:

$$\frac{d(u - v)}{dt} = -g_L (u - v). \quad (7)$$

Thus, ?? means $\epsilon(t) \rightarrow 0$ exponentially if excluding jumping boundary condition. However, when there is jump from V_T to V_r , or from 1 to 0, the coupling dynamics could be very complicated, so could the error.

Let's consider a mean-field limit for 5, i.e., we let $f \rightarrow 0, \nu \rightarrow \infty$ and meanwhile hold $f\nu = \text{const}$. Thus, the stochastic term $\frac{f^2\nu}{2} \frac{dW}{dt}$ is small enough to be neglected and we suppose

current $I = I^E - I^I$ converges to a constant exponentially. So we can still use 7 but with boundary this time. When I is large enough to drive both (u, v) firing, it's easy to check that (u, v) never converges (Fig.2 right column), as $(u(t), v(t))$ should be both close to certain periodic functions with the same period but different phases. Therefore, in the case stochasticity diminishes, we can't estimate nor even give a bound to $\epsilon(t)$ (except an apparent one: $|u(t) - v(t)| \leq 1 = V_T - V_r$), and coupling time is $+\infty$. The comparison between these 2 cases gives us that, for LIF equations with jumping boundary conditions:

(1) For deterministic cases with jumping boundary conditions, numerical error is not bounded nor convergent. We guess that randomness possibly provides error convergence, and better convergence with larger randomness.

(2) For $u(0) - v(0) = \epsilon$ and (u, v) receiving the same input, We can't expect the error converges to zero in every case, as we could easily design a "weird" input to drive $|u(t) - v(t)|$ oscillates between 0 and 1. We should seek for probabilistic convergence for LIF equations such as **convergence in probability 1**, instead of absolute convergence in each case.

Therefore, our goal is to provide a physical "proof" for the one-neuron coupling (u, v) , and extend it to more complex cases.

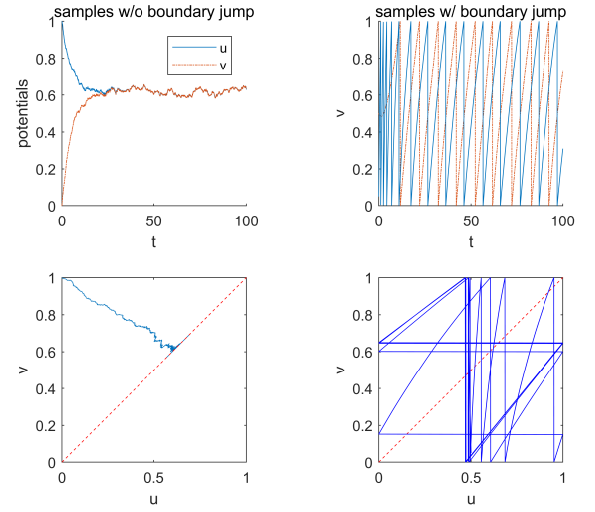


Figure 2. Comparison between non-bound-crossing (low I , left column) and bound-crossing (high I , right column). **Left column.** When both u, v never cross the bound $V_T = 1, V_{0,1}$, $\epsilon(t) = u - v$ soon converges to 0, i.e. the trajectory goes to the diagonal $u = v$ on the $u - v$ plane. **Right column.** However, when both samples keep firing (crossing bound and jumping back to 0), we can easily design non-converging ϵ . See upper-right, $I_{u,v}$ have different initial conditions, but receiving the same input, thus both ultimately converge to the same constant level. But constant $I_{u,v}$ make both u, v periodic functions with same period but different phases, thus the error $\epsilon(t) = u - v$ never converge, i.e., the trajectory never goes to the diagonal $u = v$, but travels periodically on the $u - v$ plane (lower-right).

2.3 Multilevel Monte Carlo Methods

To apply MLMC reasonably, we need $\lim_{n \rightarrow \infty} |u_{\Delta t}(n) - v_{\Delta t}(n)| \rightarrow 0$ to ensure stochastic error can be canceled by large amount of samples in the same time-grid level. On the other hand, we also need

$$\lim_{n \rightarrow \infty, \Delta t \rightarrow 0} |u_{\Delta t}(n) - v_{\frac{\Delta t}{2}}(2n)| \rightarrow 0$$

s.t. we can use different level simulations to cancel systematic error for the same sample.

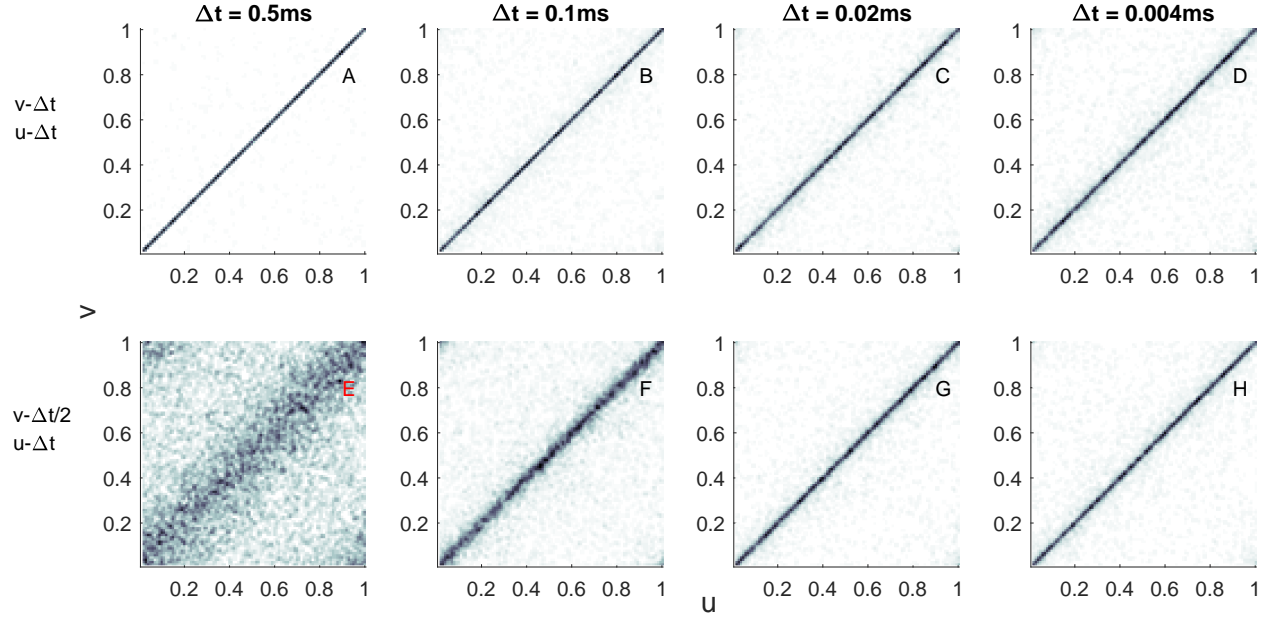


Figure 3. Probability density distributions for coupling (u, v) . For each case, we evolve 10,000 samples that initially uniformly distributed on the plane $\mathbb{T}^2 = [0, 1) \times [0, 1)$ for 900ms with different Δt time grids. Thus, each panel can be regarded as a numerical approximation for $\rho(u, v, t = 900)$ in 8a evolved from $\rho(u, v, 0) \equiv 1$. For **A-D**, the corresponding u, v s are evolved by the same time grids for " $\Delta t - \Delta t$ " couplings, where $\Delta t_{u,v} = 0.5, 0.1, 0.02, 0.004$ ms, while u, v using different time grids in **E-H** for " $\Delta t - \Delta t/2$ " couplings, where $\Delta t_u = 0.5, 0.1, 0.02, 0.004$ and $\Delta t_v = 0.5/2, 0.1/2, 0.02/2, 0.004/2$. For " $\Delta t - \Delta t/2$ " couplings, ρ more and more accumulates around diagonal when $\Delta t \rightarrow 0$, while for " $\Delta t - \Delta t$ " couplings larger Δt provides better accumulations. However, when comparing **D** and **H**, $\rho_{0.004}^{\Delta t_v}$ and $\rho_{0.004}^{0.002}$ are pretty similar, implying that for all couplings, $\Delta t_{u,v} \rightarrow 0 \Rightarrow \rho_{\Delta t_u}^{\Delta t_v} \rightarrow \rho$, regardless the relation between $\Delta t_{u,v}$.

3 Coupling Single Neurons

Coupling (u, v) for single neurons is the simplest case, and is the base of couplings for more complex neural network. To investigate the coupling time and convergent rate, we consider a Fokker-Planck equation for the joint distribution $\rho(u, v, t)$. Assuming u, v satisfies 6a and 6b, the corresponding Fokker-Planck equation should be (see details in Appendix):

$$\begin{aligned} \partial_t \rho &= \partial_u [(g_L u - I)\rho] + \frac{f^2 \nu}{2} (\partial_u + \partial_v) \rho \\ &+ \partial_v [(g_L v - I)\rho] + \frac{f^2 \nu}{2} (\partial_u + \partial_v) \rho \\ &= \partial_u [(g_L u - I)\rho] + \partial_v [(g_L v - I)\rho] + \frac{f^2 \nu}{2} (\partial_u + \partial_v)^2 \rho \end{aligned} \quad (8a)$$

According to 8a, there are advections of probability on both u, v -directions. The diffusion term, however, is only on the diagonal $u = v$. We guess

$$\lim_{t \rightarrow \infty} \rho(u, v, t) = \delta(u - v),$$

which leads to convergence with probability 1. A rigorous proof for continuous t turns out to be difficult, but what we really care for is numerical convergence. Consider 2 different kinds of couplings for discrete time and their corresponding joint distributions $\rho_{\Delta t_u}^{\Delta t_v}$: a) When (u, v) are evolved

by the same time grid, i.e. $\Delta t_u = \Delta t_v$, we should have $\lim_{t \rightarrow \infty} \rho_{\Delta t}^{\Delta t}(u, v, t) = \delta(u - v)$; b) When (u, v) are evolved by different time grids, such as $\Delta t_u = 2\Delta t_v$, we should also have $\lim_{t \rightarrow \infty} \rho_{\Delta t}^{\Delta t/2}(u, v, t) = \delta(u - v)$. See Fig. 3 for a time section when t is relatively large.

4 Coupling Neural Network

We still need an error estimator for spikes as it is the jump of membrane voltage making the whole dynamics highly nonlinear and does not fit the classical numerical theory for simulation. Before we get started, let's keep one important fact in mind:

Proposition 1 The the numerical simulation to a neuron network system, no matter how sophisticatedly connected, does obey the classical numerical convergence, as long as none of the V_j s hit the threshold, with the error $O(\Delta t)$ when we use forward Euler method.

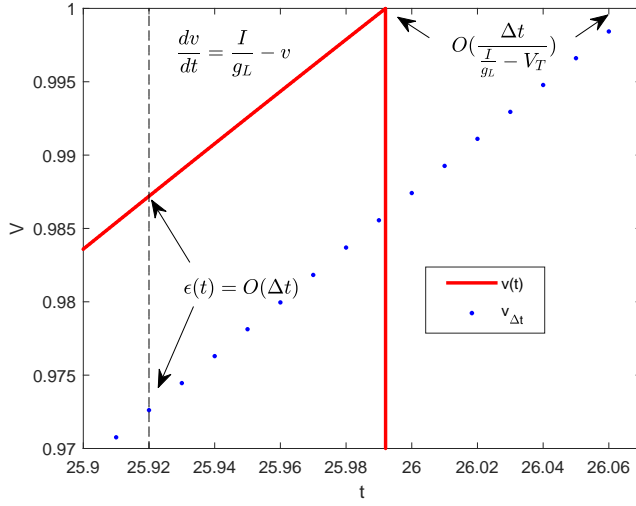


Figure 4. "Real" solution $V(t)$ (of course it's also an numerical solution but with much finer grids) and numerical solution $V(n\Delta t)$. We can choose the time steps much smaller that the frequency of input Poisson process λ , and the decay speed of I which is $\sigma_{E,I}$, thus I is approximately a constant and the trajectories for $V(t)$ and $V(n\Delta t)$ can be regarded as straight lines with slope to be $I/g_L - V_{th}$.

First, we study the spikes from only one neuron. As illustrated in Fig.4, let's consider the actual solution $V(t)$ to 10 and numerical solution $V_{\Delta t}(n)$ to 11. Before any firing event, we have $|V(n\Delta t) - V_{\Delta t}(n)| = O(\Delta t)$, and the error converge to zero in probability when $t \rightarrow \infty$. Thus, it's safe to say $|V(n\Delta t) - V_{\Delta t}(n)| = O(\Delta t)$ in the whole time period

$$\begin{aligned} \frac{dV}{dt} &= -g_L V - G^E(V - V^E) - G^I(V - V^I), \\ \sigma_E \frac{dG^E}{dt} &= -G^E + \sum_{\mu^E} f^E \delta(t - t_{\mu^E}), \\ \sigma_I \frac{dG^I}{dt} &= -G^I + \sum_{\mu^I} f^I \delta(t - t_{\mu^I}). \end{aligned} \quad (9)$$

So far we have investigated convergence for feed-forward networks for current based and conductance based, which is a large category of neural network. However, a biologically-realistic neural network usually consists of rich recurrent, i.e., feed-back connections. We now move to a 1024-neuron case with recurrent, which is broadly used in V1 model. One can see that the convergence still holds. However, comparing with feed-forward cases, the variance of error is much larger. This is because the MC method only works when the neurons firing pattern is homogenized, i.e., with weak correlation. If we consider an extreme case, that several connected neurons firing

we simulate. If there is a firing event after time point $n\Delta t$, $V(n\Delta t)$ should be very close to 1 and so is $V_{\Delta t}(n)$. Also, the variation of I is bounded by $O(\Delta t)$ in a small period of Δt . That's to say, the difference between the real spiketime t_s and the numerical spiketime t_{ns} is $O(\Delta t / (I/g_L - V_{th}))$, as both trajectories can be approximately regarded as straight lines before the firing event. Now we have another important fact:

Proposition 2 The time errors for spike series produced by one neuron are $O(\Delta t / (I/g_L - V_{th}))$, thus of magnitude $O(\Delta t)$ as long as $I/g_L - V_{th}$ has a lower bound, which is necessary for a neuron to fire.

Now let's consider an application of Proposition 2 to a 2-neuron feed-forward case. We input Poisson-like series and constant inhibitory current, for the sake of increasing the variation of I^1 , to neuron 1. Excitatory neuron 1 produce spikes as the only input to neuron 2. Proposition 1 gives us that the spiking time errors for neuron 1 is $O\Delta t$ as the numerical results for I^1 and V^1 are both $O\Delta t$ accurate, which means I^2 is $O\Delta t$ accurate except for several narrow time bins brought by some spikes with larger time errors. That's to say, V^2 and spiking series time for neuron 2 are also $O\Delta t$ accurate in general. The details for the system is illustrated in Figure 5.

Therefore, **Proposition 2** can be easily extended to *all feed-forward networks* by induction, and the simulation would be trivial. To convince our reader, we now simulate a feed-forward network, but with a more biological setup, i.e., each neuron is driven by a conductance-based I-F equation:

Note that the the role I plays in current-based model is now substituted by G^E , which receives input spikes and drives a neuron to fire. The convergence theorem for 9 shouldn't be dramatically different from **Theorem 1**, as $\frac{G^E(V - V^E)}{I^E}$ is bounded. We show the numerical results for a conductance-based feed-forward network with different grids in Figure. 6, from which we can see **Proposition 2** also holds, even if we change current-based model to conductance-based.

together, which are synchronized, it's clear that no numerical method can control the error when $\Delta t \rightarrow 0$.

Discussion

Acknowledgments

We thank just about everybody.

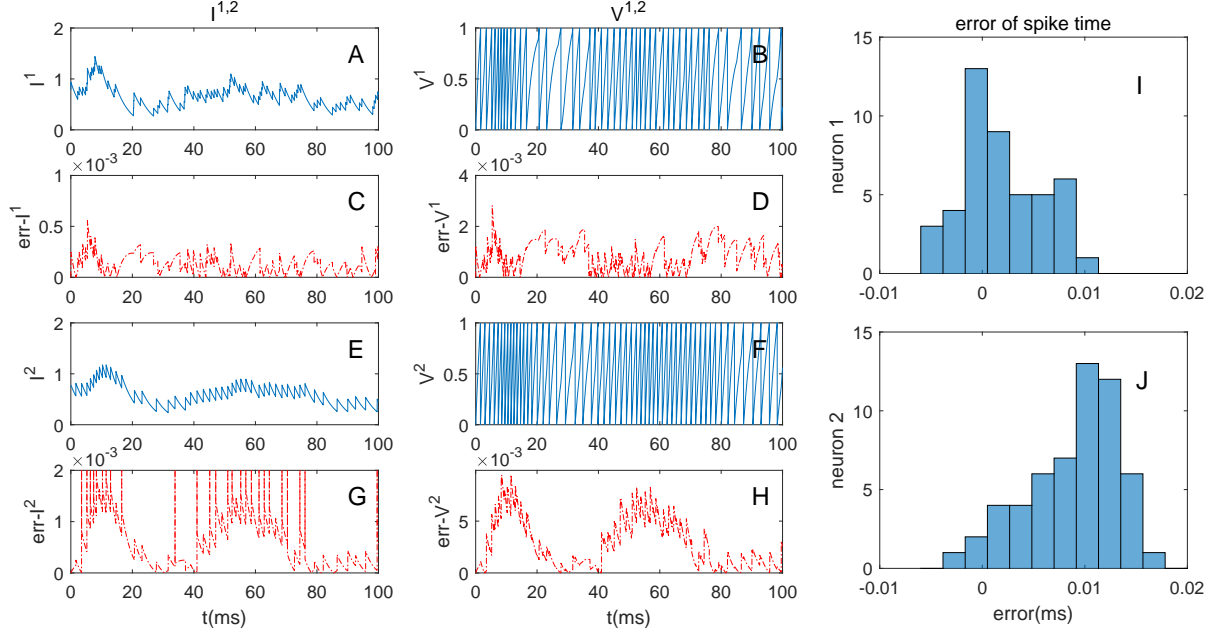


Figure 5. This figure describes the numerical simulations for a 2-neuron network by $\Delta t = 0.01\text{ms}$ time grids. Blue curves (A,B,E,F) stand for the numerical results of $I^{1,2}$ and $V^{1,2}$ with 10^{-2}ms time grid, whereas (C,D,G,H) red curves represent errors (given by the difference between these results with the numerical results by much finer grids) for corresponding variables. Sub-figures in the upper half describe variables for neuron 1 (A,B,C,D,I), while the lower half for neuron 2 (E,F,G,H,J). From C,D,I we can see that, the errors for I^1 , V^1 and spiking series t_{μ^1} are all bounded by $O(\Delta t)$, which is no surprise from **Proposition 2**. Thus, as $I^2 = \sum_{\mu^1} f e^{-(t-t_{\mu^1})/\sigma}$ is a linear combination of a series of exponential decay functions, the error of I^2 (G) is made up by the “spike errors” from t_{μ^1} and calculation errors introduced when we integrate I^2 . As V^2 is also integrated from I^2 , we can see that the error for V^2 (H) is also bounded by $O(\Delta t)$, thus the same for spiking time points as t_{μ^2} (J).

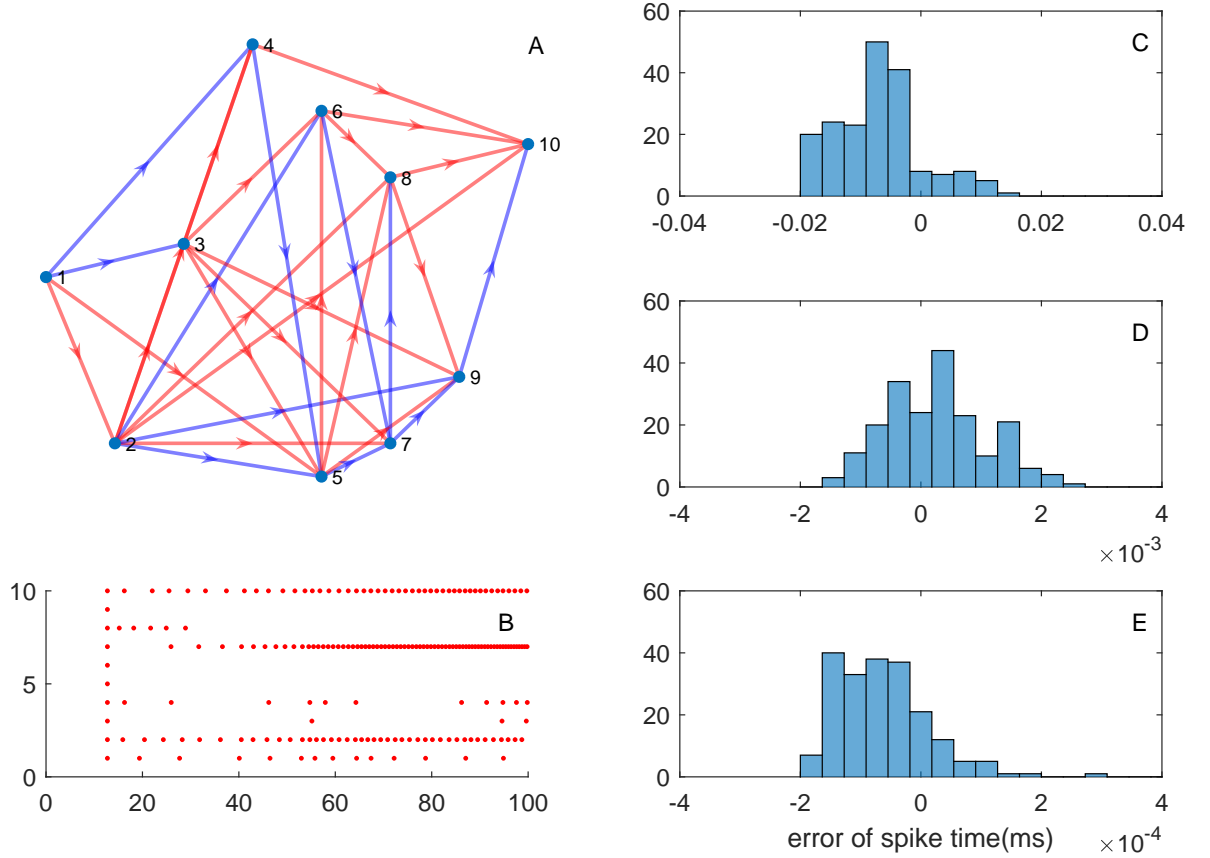


Figure 6. We use the same Poisson-like input to simulate a 10-neuron feed-forward neural network with 4 different grids ($\Delta t = 10^{-2}, 10^{-3}, 10^{-4}$ and 10^{-5} ms). **(A)**, the topological graph for a 10-neuron feed-forward neural network, where red arrows represent excitatory connections and blue arrows for inhibitory connections. One can see from the graph that this is a network without any feedback loop. **(B)**, each red dot stands for a firing event for corresponding neuron in this network. **(C, D, E)**, we spike time differences between 10^{-m} and 10^{-m-1} grids for the whole network, where $m = 2, 3, 4$. We can see the numerical results for different numerical grids do converge when grids become finer and finer.

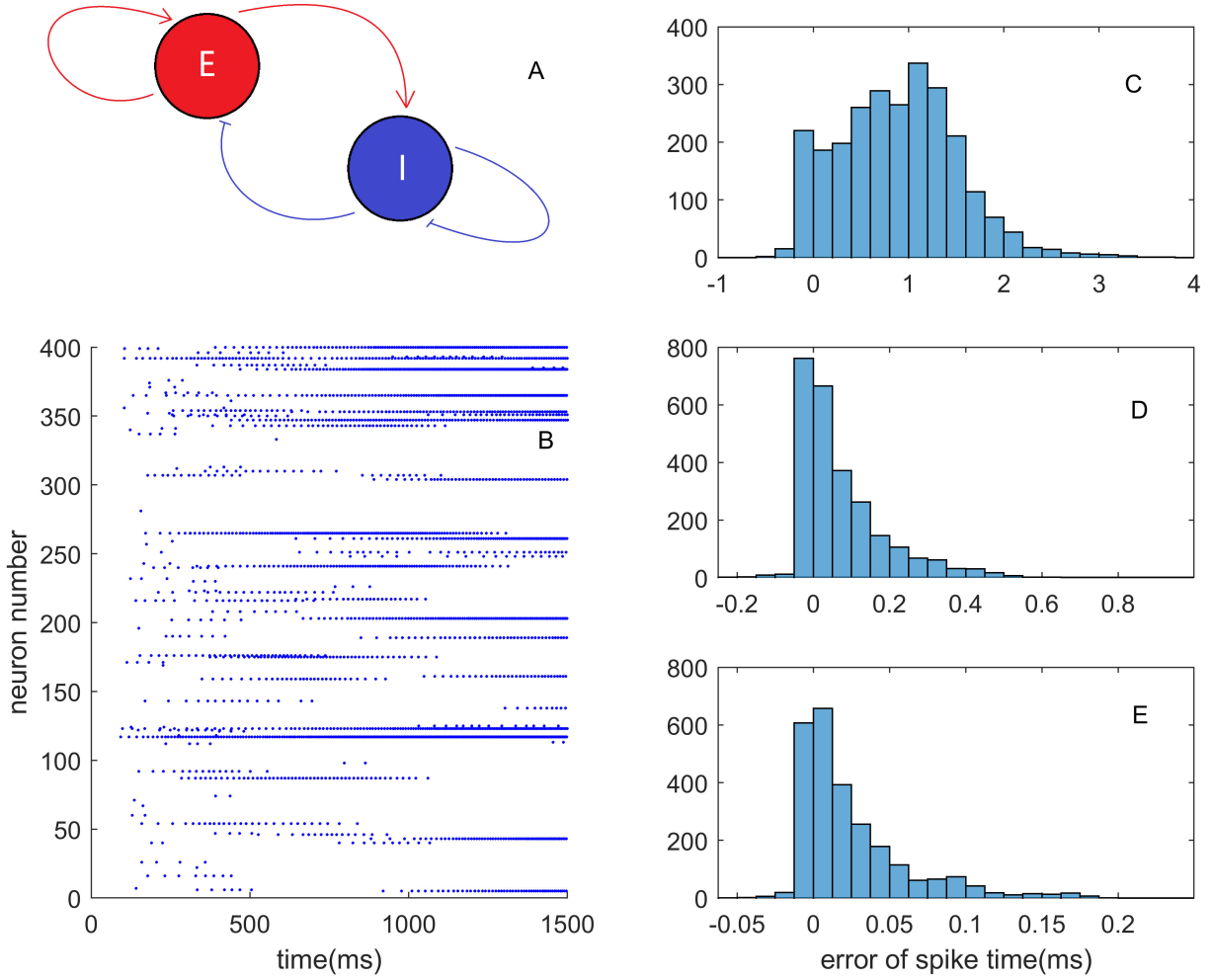


Figure 7. We use the different Poisson-like inputs each neuron to simulate a 400-neuron V1 network with 4 different grids ($\Delta t = 0.1, 0.1 * 4^{-1}, 0.1 * 4^{-2}$ and $0.1 * 4^{-3}$ ms)

Appendix

To restrict our discussion to $(2 + 1)$ dimensional PDE (or numerical PDE), we write the F-P eqns for a homogenized version of 3a, and it's numerical form:

$$\frac{dV_{0,1}}{dt} = -g_L V_{0,1} + \sum_{\mu} f \delta(t - t_{\mu}), \quad (10)$$

$$\frac{V_{0,1}^{(n+1)h} - V_{0,1}^{nh}}{h} = -g_L V_{0,1}^{nh} + \sum_{nh \leq \mu < (n+1)h} f \delta(t - t_{\mu}) \quad (11)$$

where $V_1(0) - V_0(0) = \epsilon$ is satisfied for both 10 and 11. Inspired by I_1 , we now look at probability distribution $\rho(V_0, V_1, t)$ on time t (for continuous or discrete time) and plane $V_0 \times V_1$.

For 10, the F-P eqn is

$$\partial_t \rho(V_0, V_1, t) = \partial_{V_0}(g_L V_0 \rho) + \partial_{V_1}(g_L V_1 \rho) + \lambda[\rho(V_0 - f, V_1 - f, t) - \rho(V_0, V_1, t)] \quad (12)$$

For small f , with Taylor expansion we could rewrite 10 into

$$\begin{aligned} \partial_t \rho &= \partial_{V_0}(g_L V_0 \rho) + \partial_{V_1}(g_L V_1 \rho) + \lambda[-f(\partial_{V_0} + \partial_{V_1})\rho + \frac{f^2}{2}(\partial_{V_0}^2 + 2\partial_{V_0}\partial_{V_1} + \partial_{V_1}^2)\rho] \\ &= \partial_{V_0}[(g_L V_0 - f\lambda)\rho + \frac{f^2\lambda}{2}(\partial_{V_0} + \partial_{V_1})\rho] + \partial_{V_1}[(g_L V_1 - f\lambda)\rho + \frac{f^2\lambda}{2}(\partial_{V_0} + \partial_{V_1})\rho] \end{aligned}$$

where we often use μ to substitute $f\lambda$ and σ^2 for $f^2\lambda$. According to the equation of continuity $\partial_t \rho + \partial_{V_0} J_{V_0} + \partial_{V_1} J_{V_1} = 0$, we have:

$$\begin{aligned} J_{V_0} &= -(g_L V_0 - \mu)\rho - \frac{\sigma^2}{2}(\partial_{V_0} + \partial_{V_1})\rho, \\ J_{V_1} &= -(g_L V_1 - \mu)\rho - \frac{\sigma^2}{2}(\partial_{V_0} + \partial_{V_1})\rho, \end{aligned} \quad (14)$$

with natural boundary conditions (similar to (4.25) in [2]):

$$\begin{aligned} J_{V_0}(0, v_1, t) &= J_{V_0}(1, v_1, t), & \rho(0, v_1, t) &= \rho(1, v_1, t) \\ J_{V_1}(v_0, 0, t) &= J_{V_1}(v_0, 1, t), & \rho(v_0, 0, t) &= \rho(v_0, 1, t) \end{aligned} \quad (15)$$

For 11, we consider the condition probability for $\rho(V_0, V_1, t + \Delta t)$ giving conditions for $V_0(t), V_1(t)$. Assuming Δt is too small for more than 1 input events occur in $[t, t + \Delta t)$, then there are only 2 choices for the system in this time period: receiving one input spike with probability $\lambda \Delta t$, and receiving nothing with probability $1 - \lambda \Delta t$. Thus, we have:

$$\begin{aligned} & \mathbb{E}[\delta(v - V_0(t + \Delta t))\delta(v - V_1(t + \Delta t))|V_0(t), V_1(t)] \\ &= \lambda \Delta t \delta(v - V_0(t) - f)\delta(v - V_1(t) - f) \\ &+ (1 - \lambda \Delta t)\delta(v - V_0(t) + g_L V_0(t)\Delta t)\delta(v - V_1(t) + g_L V_1(t)\Delta t) \end{aligned}$$

Using Taylor expansion for δ -function:

$$\delta(v - V_i(t) + g_L V_i(t)\Delta t) = \delta(v - V_i(t)) + \partial_v [\delta(v - V_i(t))g_L V_i(t)]\Delta t + O(\Delta t^2)$$

we can rewrite 16 into probability form:

$$\begin{aligned} \rho_{\Delta t}(V_0, V_1, t + \Delta t) &= \lambda \Delta t \rho_{\Delta t}(V_0 - f, V_1 - f, t) \\ &+ (1 - \lambda \Delta t)[\rho_{\Delta t} + \partial_{V_0}(g_L V_0 \rho_{\Delta t})\Delta t + \partial_{V_1}(g_L V_1 \rho_{\Delta t})\Delta t] + O(\Delta t^2) \end{aligned}$$

where $\rho_{\Delta t}$ is the probability distribution function for discrete-time Fokker-Planck equation in 17. One can easily check that 17 is consistent with 12 at order $O(\Delta t)$, thus the "numerical" solution converges to the "actual" solution for 12 when $\Delta t \rightarrow 0$. Thus, the global error for $\Delta t - \Delta t$ grid converge to the global error for actual solution in probability.

To use numerical results from different grids, we also need to consider the case that V_0 is evolved by Δt -grids and V_1 is evolved by $\frac{\Delta t}{2}$ -grids, or $\Delta t - \frac{\Delta t}{2}$ grids, to see if the global error $V_0 - V_1$ diminish in time. Similar to 16, we could write the conditional expectation as:

$$\begin{aligned} & \mathbb{E}[\delta(v - V_0(t + \Delta t))\delta(v - V_1(t + \Delta t))|V_0(t), V_1(t)] \\ &= \frac{\lambda \Delta t}{2} \delta[v - V_0(t) - f]\delta[v - V_1(t) - f + g_L(V_1(t) + f)\frac{\Delta t}{2}] \\ &+ \frac{\lambda \Delta t}{2} \delta[v - V_0(t) - f]\delta[v - V_1(t) - f + g_L V_1(t)\frac{\Delta t}{2}] \\ &+ (1 - \lambda \Delta t)\delta[v - V_0(t) + g_L V_0(t)\Delta t]\delta[v - V_1(t) + g_L V_1(t)\frac{\Delta t}{2}] \end{aligned}$$

Using Taylor expansion for δ -function, we know the corresponding F-P eqn is exactly the same as 17 if we discard terms with order higher than $O(\Delta t)$. This is suggested by Fig.3, as if we fix a finite time point t both 2 series of discrete-time distributions ($\Delta t - \Delta t$ and $\Delta t - \frac{\Delta t}{2}$) seem to have the same limit when $\Delta t \rightarrow 0$, i.e., $\rho_{\Delta t}(V_0, V_1, t) \rightarrow \rho(V_0, V_1, t)$. Also, it is implied by the figure that the density for $\rho(V_0, V_1, t)$ should aggregate along the diagonal $V_0 = V_1$ when $t \rightarrow \infty$.

We have proved that both $\Delta t - \Delta t$ and $\Delta t - \frac{\Delta t}{2}$ grids yield discrete-time F-P eqns that are consistent with 12. The solutions to them thus converge to 13 when $f \rightarrow 0$. The last part of proof for convergence in probability is to show that the solution to 13 converge to the diagonal $V_0 = V_1$ when $t \rightarrow \infty$.

Steady Solution on the Diagonal

We now start to show that the steady distribution $\rho_{\infty}(V_0, V_1)$ can only be non-zero on the diagonal $V_0 = V_1$. Before that, we need several lemmas for preparation.

Lemma 1 For the steady solution $p_{\infty}(v)$ of the F-P eqn for only 1 neuron:

$$\partial_t p(v, t) = \partial_v \left\{ [g_L v - \mu]p + \frac{\sigma^2}{2} \partial_v p \right\}, \quad (19)$$

$$m(t) = J(V_T, t) = -[g_L v - \mu]p - \frac{\sigma^2}{2} \partial_v p, \quad p(0, t) = p(1, t) = 0$$

we have $M_1 = \mu - m/g_L$ and $M_2 = \mu M_1 + [\sigma^2(1 - p_{\infty}(1)) - m]/(2g_L)$ for M_1 and M_2 are the first 2 order moments of $p_{\infty}(v)$.

Proof.

$$\begin{aligned} \frac{dM_j}{dt} &= \int_0^1 v^j \partial_t p(v, t) dt \\ &= \int_0^1 v^j \partial_v \left\{ [g_L v - \mu]p + \frac{\sigma^2}{2} \partial_v p \right\} dt \\ &= -m - j[g_L M_j - g_L \mu M_{j-1} + \frac{\sigma^2}{2} (V_T^{j-1} p(1, t) - V_r^{j-1} p(0, t))] \end{aligned}$$

Especially, for $j = 1, 2$, as $M_0 = \int_0^1 p(v, t) dt = 1$, we have:

$$\begin{aligned} \frac{dM_1}{dt} &= -m - g_L (M_1 - \mu), \\ \frac{dM_2}{dt} &= -m - 2[g_L M_2 - g_L \mu M_1 + \frac{\sigma^2}{2} (p(1, t) - 1)]. \end{aligned}$$

So we could find the steady states for M_1 and M_2 .

Lemma 2 Consider $\mathbb{E}(V_0 V_1)$ for $\rho_{\infty}(V_0, V_1)$, we have

$$\mathbb{E}(V_0 V_1) = \mu^2 - \frac{\mu m}{g_L} + \frac{\sigma^2(1 - p_{\infty}(1))}{2g_L} - \frac{1}{g_L} \int_0^1 m(V) V dV. \quad (21)$$

Proof.

$$\begin{aligned} \frac{d\mathbb{E}(V_0 V_1)}{dt} &= \int_0^1 \int_0^1 V_0 V_1 \partial_t \rho dV_0 dV_1 \\ &= - \int_0^1 V_0 dV_0 \int_0^1 V_1 dJ_{V_1} - \int_0^1 V_1 dV_1 \int_0^1 V_0 dJ_{V_0} \end{aligned}$$

As V_0 and V_1 are commutable in 13, i.e., they have the same status, the stationary distribution function $\rho_{\infty}(V_0, V_1)$ should be a symmetrical function along the diagonal $V_0 = V_1$, thus $\int_0^1 V_0 dV_0 \int_0^1 V_1 dJ_{V_1} = \int_0^1 V_1 dV_1 \int_0^1 V_0 dJ_{V_0}$, giving us that

$$\begin{aligned} \frac{d\mathbb{E}(V_0 V_1)}{dt} = 0 &\implies \int_0^1 V_1 dV_1 \int_0^1 V_0 dJ_{V_0} = 0 \\ &\implies \int_0^1 V_1 dV_1 \left\{ -m(V_1) - \int_0^1 [(g_L V_0 - \mu)\rho + \frac{\sigma^2}{2} \partial_v \rho] dV_0 \right\} \\ &\implies - \int_0^1 m(V_1) V_1 dV_1 - g_L \mathbb{E}(V_0 V_1) + \mu M_1 - \frac{\sigma^2}{2} \int_0^1 \partial_v \rho dV_0 dV_1 = 0 \end{aligned}$$

With **Lemma 1**, we know that 21 is true.

Lemma 3 ρ_∞ attaches 0 on the boundary, except the 4 corners.

Proof.

Again, as V_0 and V_1 are commutable in 13, we know that $\rho_\infty(V_0, V_1) = \rho_\infty(V_1, V_0)$. So

$$\begin{aligned} \partial_{V_0} J_{V_0} + \partial_{V_1} J_{V_1} = 0 &\implies \partial_{V_0} J_{V_0}(x, y) + \partial_{V_0} J_{V_0}(y, x) = 0 \\ &\implies J_{V_0}(x, y) + J_{V_0}(y, x) = f(y). \end{aligned}$$

As $J_{V_0}(0, y) = J_{V_0}(1, y)$, we then have $J_{V_0}(y, 0) = J_{V_0}(y, 1)$ for arbitrary $y \neq 0, 1$. Thus,

$$(y - \lambda f)\rho(y, 0) + (\partial_{V_0} + \partial_{V_1})\rho(y, 0) = (y - \lambda f)\rho(y, 1) + (\partial_{V_0} + \partial_{V_1})\rho(y, 1).$$

Combining this with the boundary condition that $\rho(y, 0) = \rho(y, 1)$, we can see that $\partial_{V_0}\rho(y, 0) = \partial_{V_0}\rho(y, 1) \implies \partial_{V_1}\rho(0, y) = \partial_{V_1}\rho(1, y)$. Use $J_{V_0}(0, y) = J_{V_0}(1, y)$ and $\rho(0, y) = \rho(1, y)$ again, now we have

$$(0 - \lambda f)\rho(0, y) + (\partial_{V_0} + \partial_{V_1})\rho(0, y) = (1 - \lambda f)\rho(1, y) + (\partial_{V_0} + \partial_{V_1})\rho(1, y).$$

So $\rho(0, y) = 0$ for arbitrary $y \neq 0, 1$.

Now we can go to prove the main theorem for the steady solution.

Theorem 1 The steady distribution for 13 has a support on the diagonal line $V_0 = V_1$.

Proof.

From **Lemma 3** we can see that $\int_0^1 m(V)VdV = m(0) \times 0 + m(1) \times 1 = m/2$ as $m(0) = m(1)$. Thus combining **Lemma 1** with **Lemma 2** we get

$$\begin{aligned} \mathbb{E}(V_0 V_1) &= \mu^2 - \frac{\mu m}{g_L} + \frac{\sigma^2(1 - p_\infty(1))}{2g_L} - \frac{m}{2g_L} \\ &= M_1^2 + M_2 = \mathbb{E}(V_0)\mathbb{E}(V_1) + \sqrt{\text{var}(V_0)\text{var}(V_1)} \end{aligned}$$

which exactly means that $\text{cor}(V_0, V_1) = [\mathbb{E}(V_0 V_1) - \mathbb{E}(V_0)\mathbb{E}(V_1)]/\sqrt{\text{var}(V_0)\text{var}(V_1)} = 1$. Thus, stochastic variables V_0 and V_1 are fully synchronized when $t \rightarrow \infty$. Convergence in probability is then proved.

References

1. Giles, Michael B. "Multilevel monte carlo path simulation." Operations Research 56.3 (2008): 607-617.
2. Cai, David, et al. "Kinetic theory for neuronal network dynamics." Communications in Mathematical Sciences 4.1 (2006): 97-127.
3. Rangan, Aaditya V, et al. "Dynamics of spiking neurons: between homogeneity and synchrony." J Comput Neurosci (2013) 34:433–460.