



# Diffusion approximation-based simulation of stochastic ion channels: which method to use?

Danilo Pezo<sup>1</sup>, Daniel Soudry<sup>2</sup> and Patricio Orio<sup>1,3\*</sup>

<sup>1</sup> Centro Interdisciplinario de Neurociencia de Valparaíso, Universidad de Valparaíso, Valparaíso, Chile

<sup>2</sup> Department of Statistics and the Center for Theoretical Neuroscience at Columbia University, New York, NY, USA

<sup>3</sup> Instituto de Neurociencia, Facultad de Ciencias, Universidad de Valparaíso, Valparaíso, Chile

## Edited by:

Joshua H. Goldwyn, New York University, USA

## Reviewed by:

Daniele Lino, University of Antwerp, Belgium

Roberto Fernández Galán, Case Western Reserve University, USA

## \*Correspondence:

Patricio Orio, Centro Interdisciplinario de Neurociencia de Valparaíso, Pasaje Harrington 287, Valparaíso 2360103, Chile  
e-mail: patricio.orio@uv.cl

To study the effects of stochastic ion channel fluctuations on neural dynamics, several numerical implementation methods have been proposed. Gillespie's method for Markov Chains (MC) simulation is highly accurate, yet it becomes computationally intensive in the regime of a high number of channels. Many recent works aim to speed simulation time using the Langevin-based Diffusion Approximation (DA). Under this common theoretical approach, each implementation differs in how it handles various numerical difficulties—such as bounding of state variables to  $[0,1]$ . Here we review and test a set of the most recently published DA implementations (Goldwyn et al., 2011; Lino et al., 2011; Dangerfield et al., 2012; Orio and Soudry, 2012; Schmandt and Galán, 2012; Güler, 2013; Huang et al., 2013a), comparing all of them in a set of numerical simulations that assess numerical accuracy and computational efficiency on three different models: (1) the original Hodgkin and Huxley model, (2) a model with faster sodium channels, and (3) a multi-compartmental model inspired in granular cells. We conclude that for a low number of channels (usually below 1000 per simulated compartment) one should use MC—which is the fastest and most accurate method. For a high number of channels, we recommend using the method by Orio and Soudry (2012), possibly combined with the method by Schmandt and Galán (2012) for increased speed and slightly reduced accuracy. Consequently, MC modeling may be the best method for detailed multicompartment neuron models—in which a model neuron with many thousands of channels is segmented into many compartments with a few hundred channels.

**Keywords:** channel noise, stochastic simulation, ion channel, Langevin, conductance based models

## INTRODUCTION

Understanding the effect of stochastic phenomena on the behavior of the nervous system requires stochastic simulation algorithms that effectively and accurately capture the dynamics of the underlying modeled phenomena. Among the sources for variability, the stochastic opening and closing of ion channels has caught the attention of several works over the past years. The best description of stochastic gating of ion channels is attained with the use of continuous time, discrete states Markov Chain (MC) processes (Neher and Stevens, 1977; Colquhoun and Hawkes, 1981), however this approach can be very slow in simulations with a large number of channels.

As an alternative to the explicit MC simulation, the Diffusion Approximation (DA) calculates the trajectory of a population of independent MCs using a Stochastic Differential Equation (SDE), sometimes called the Chemical Langevin Equation (Gillespie, 2000, 2007). Its application to the simulation of stochastic ion channels was suggested almost 20 years ago (Fox and Lu, 1994; Fox, 1997), but in the beginning there were some errors in the application of the scheme. This led to the belief that the approximation was not good enough (Mino et al., 2002; Bruce, 2007, 2009). Later, revised implementations of the algorithms were

published showing that indeed the DA can reproduce the statistical properties of a population of discrete ion channel fluctuating between open and closed states (Goldwyn et al., 2011; Goldwyn and Shea-Brown, 2011). Other works (Dangerfield et al., 2012; Orio and Soudry, 2012) also offered a simplified description of the algorithm, making it easy to apply to any given kinetic scheme.

What the SDE system approach does is to approximate the trajectory in time of the fraction of channels at every state. For the simulation to remain physically meaningful, none of the fractions can be negative or greater than 1. We call this the “boundary constraint.” This constraint would occasionally break in the numerical simulation of the SDE, if we use its naively discretized form (the Euler–Maruyama method). This is because stochastic fluctuations can make the variables leave the  $[0,1]$  interval. The problem amplifies when the number of channels is low and stochastic fluctuations increase.

If the boundary constraint is not maintained, this can generate additional technical problems in the simulation. Specifically, the calculation of the stochastic terms involves the square root of a term involving the fraction variables, which may yield complex values when the variables are negative. Such complex values must be avoided. Therefore, we get another constraint, which we call

the “Real-valued Square Root” (RSR) constraint. This constraint is automatically fulfilled if the boundary constraint is maintained.

Finally, the sum of the fractions over all states must be equal to 1 at all times. We call this the “normalization constraint.” Although this constraint is supposed to be guaranteed by the continuous-time equations (see below), machine rounding errors of the discretized equations can gradually break it. Additionally, any method that deals with the boundary constraint must also take normalization into account. For example, a naive truncation of any variable that leaves the  $[0,1]$  interval would break the normalization constraint.

To address these issues, a number of improvements have been proposed to the DA schemes.

Orio and Soudry (2012) proposed to allow the variables to freely change, ignoring the boundary constraint. In order to take care of the normalization constraint one of the variables was replaced with one minus the sum of the others. Additionally, an absolute value operation was added in the stochastic terms to maintain the RSR constraint. Previously, Goldwyn et al. (2011) also allowed the variables to freely change, but instead used a steady state approximation on the voltage to maintain the RSR constraint. However, such an approximation can be rather inaccurate (Dangerfield et al., 2012; Orio and Soudry, 2012; Huang et al., 2013a), even when the number of channels is relatively high.

Two methods aim to maintain the boundary constraint. Dangerfield et al. (2012) proposed that if variables break either the boundary or normalization constraint, they are “reflected” back into the valid region, in which all the constraints are kept. This is done using projection into a simplex (Chen and Ye, 2011). Huang et al. (2013a) proposed a different method. When the boundary constraint is broken, the variables should first be truncated. This breaks normalization, so the variables are then renormalized. Finally, in the next time step, the variables are incremented with the remainders from the truncation in the previous steps. As the boundary constraint is almost constantly broken, normalization is continuously corrected in both Dangerfield et al. (2012) and Huang et al. (2013a).

Other methods have been proposed with a different goal in mind. Schmandt and Galán (2012) aimed to reduce computational complexity and speeding up the simulation. They proposed to neglect stochastic noise terms in all state transitions, except those connecting the open state (or states), an approximation they call “Stochastic shielding.” Güler (2013) introduced a stochastic HH model with colored noise in the conductance terms as well as in the current terms in order to capture the non-trivial cross-correlation between the transmembrane voltage fluctuation and the component of open channel fluctuation attributed to multiple number of gates in individual ion channels. Another recently published method (Linaro et al., 2011), also used colored noise in the current terms (but not in the conductance terms). However, Linaro’s method will not be examined here, because it involves a steady-state approximation in the stochastic terms (similarly to Goldwyn et al., 2011), which was shown to introduce inaccuracies (Orio and Soudry, 2012).

Despite the improvement in accuracy or computational efficiency that the new methods represent for the simulation of stochastic ion channel activity, we were concerned about the

comparisons performed between them and the real benefit of implementing the numerical algorithms.

First, there is the computational cost issue. The initial motivation for developing DA methods was to make stochastic simulations faster than MC modeling. Therefore, if the extra computation needed to normalize and bound the variables makes it slower than MC then the purpose is defeated. Moreover, we already noticed that when the number of channels is low (when DA becomes more inaccurate) or with very small integration times, MC modeling can run faster than DA (see Figure 7 in Orio and Soudry, 2012). This, added to the fact that bounding and normalization of the DA requires more coding (and eventually, debugging), can render DA less attractive.

Second, we noted that the standard test employed to prove the accuracy of numerical methods for stochastic ion channels is the original Hodgkin and Huxley (HH) model. This was the only model used for testing in most previous papers, including a recent review (Rowat and Greenwood, 2014). This model, as standard and general as it is, reproduces the kinetics of ion channels of the squid axon at  $6.3^{\circ}\text{C}$ , thus differing greatly from mammalian nerve excitable membranes. This difference can be very significant, as we noted (Orio and Soudry, 2012). There we found that the use of a steady-state approximation in the stochastic terms usually does not introduce severe inaccuracies in the context of the original HH model. However, deviations were detected in common current clamp-based simulations when the steady-state approximation is used in a model inspired in mammalian (therefore faster) ion channels. It is noteworthy that the difference between mammalian inspired and the squid axon model relies only in the parameters that describe the transition rate constants (and thus the time scale of the model), while the equations and the model framework are identical.

Thus, we see a necessity for testing the DA with and without the recently proposed corrections in a wider spectrum of simulation scenarios and taking into account other variables than simulation accuracy, namely:

- To test the algorithms in models with faster kinetics than Hodgkin and Huxley (time scales of mammalian neurons) and models with geometry, where the number of ion channels in different compartments may differ.
- To quantify the real advantage of DA, and specifically its accuracy vs. its computational cost in comparison to MC.

In an attempt to test the real usability of the algorithms in the context of more complex neuronal models, we implemented them in one of the choice tools for biophysically-inspired modeling, the Neuron simulation environment (Hines and Carnevale, 2001; Carnevale and Hines, 2006). We conducted both single-compartment and multi-compartment simulations using MC or DA algorithms and compared their performance as well as the ability of different DA implementations to reproduce the variability introduced by MC modeling.

In our results all DA algorithms deviate to some degree from the MC modeling when the number of channels falls below 1000, regardless of the attempts to deal with normalization and bounding of the variables. However, we see that in this condition MC

modeling runs, in most of the scenarios, faster than DA implementations. Therefore, one of the most common motivations to use DA, which is to achieve faster computation times, is not accomplished when the number of channels is low. However, when the number of channels is high, DA algorithms can accurately reproduce MC, with improved speed. Specifically, in this regime, no inaccuracy was detected in both Orio and Soudry (2012) and Huang et al. (2013a); Schmandt and Galán (2012) was slightly inaccurate; Güler (2013) was somewhat inaccurate; and Dangerfield et al. (2012) was the least accurate. In terms of computational speed, the ranking is as follows (see **Figure 8**): (1) Stochastic Shielding (Schmandt and Galán, 2012) (2) Colored Noise (Güler, 2013) (3) Unbound DA (Orio and Soudry, 2012) (4) Reflected DA (Dangerfield et al., 2012) (5) Truncated and Restored DA (Huang et al., 2013a).

## MATERIALS AND METHODS

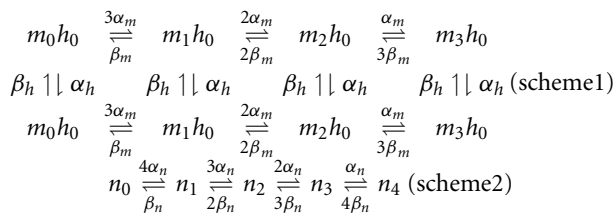
### SIMULATIONS: MODELS EMPLOYED AND TESTS PERFORMED

#### Original Hodgkin and Huxley model

The original Hodgkin and Huxley (HH) model (Hodgkin and Huxley, 1952) was simulated with the equation:

$$C_m \frac{dV(t)}{dt} = -g_{Na}(t)(V(t) - E_{Na}) - g_K(t)(V(t) - E_K) - g_l(V(t) - E_l) + I_{stim}(t) \quad (1)$$

With the exception of Güler's colored noise algorithm, sodium and potassium channels were treated as 8- and 5-state MCs, respectively. The corresponding kinetic schemes are:



Sodium and potassium conductances at time  $t$  ( $g_{Na}(t)$  and  $g_K(t)$ ) were calculated as the fraction of channels in the conducting states  $m_3 h_1$  and  $n_4$  multiplied by the maximum conductances  $\overline{g_{Na}}$  and  $\overline{g_K}$ , respectively. The kinetic rates  $\alpha$  and  $\beta$  are given (in  $ms^{-1}$ ) by:

$$\alpha_m(V) = \frac{0.1(V + 40)}{1 - \exp\left(-\frac{V + 40}{10}\right)}; \quad \beta_m(V) = 4 \exp\left(-\frac{V + 65}{18}\right)$$

$$\alpha_h(V) = 0.07 \exp\left(-\frac{V + 65}{20}\right); \quad \beta_h(V) = \frac{1}{1 + \exp\left(-\frac{V + 35}{10}\right)}$$

$$\alpha_n(V) = \frac{0.01(V + 55)}{1 - \exp\left(-\frac{V + 55}{10}\right)}; \quad \beta_n(V) = 0.125 \exp\left(-\frac{V + 65}{80}\right),$$

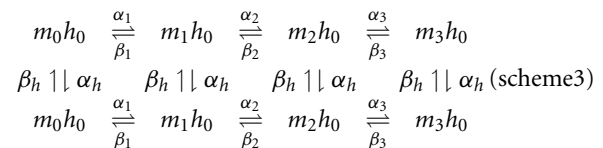
where the terms were corrected to adjust the resting potential to  $-65$  mV. Correspondingly, the rest of parameters are:  $C_m = 1 \mu F/cm^2$ ,  $\overline{g_{Na}} = 120 mS/cm^2$ ,  $\overline{g_K} = 36 mS/cm^2$ ,  $g_l = 0.1 mS/cm^2$ ,  $E_{Na} = 50 mV$ ,  $E_K = -77 mV$ ,  $E_l = -54.3 mV$ .

With this model, the following tests were conducted:

- A 500-s simulation in the absence of any input. When the number of sodium channels is in the order of 20,000 (and lower), spontaneous firing starts to occur. We recorded the spike events and calculated the mean firing rate and the distribution of inter-spike intervals (ISIs).
- 15-s current clamp with 2-ms stimulus. The stimulus current was applied with 1 ms delay. Afterwards, 12 additional ms were simulated and the occurrence and timing of an action potential was recorded. The current amplitude varied from 0 to  $15 \mu A/cm^2$  and 10,000 simulations were performed for each amplitude. Then, the firing efficiency, mean action potential time and variance of action potential time were calculated.
- Voltage clamp with action potential trace. A noisy voltage trajectory of 100 ms (including an action potential) was produced by simulating the HH model with the UA algorithm. Then, this trajectory was used as input to a stochastic model and the number of open channels in time was recorded. 2000 simulations were run and the mean and variance of open channels at each time was calculated. Additionally, the same procedure was performed with a deterministic HH model, thus allowing to obtain the expectation of open sodium and potassium channels,  $E[Na_O](t)$  and  $E[K_O](t)$ . The expected variance was also calculated as  $var[Na_O](t) = E[Na_O](t)(1 - E[Na_O](t))/N_{Na}$  and  $var[K_O](t) = E[K_O](t)(1 - E[K_O](t))/N_K$ . The results of the stochastic simulations were then compared to this exact solution.

#### Schmidt-Hieber and Bischofberger model—single compartment

The Schmidt-Hieber and Bischofberger (SB) model was proposed after the characterization of sodium channels both at the soma and at the axon initial segment of granule cells of the hippocampus (Schmidt-Hieber and Bischofberger, 2010). Sodium channels are described by the following kinetic scheme:



where the kinetic rates are given by

$$\alpha_i(V) = \alpha_{i,0} \exp(\alpha_{i,1} V)$$

$$\beta_i(V) = \beta_{i,0} \exp(-\beta_{i,1} V)$$

$$\alpha_h(V) = \frac{\alpha_{h,0}}{1 + \alpha_{h,1} \exp(\alpha_{h,2} V)}$$

$$\beta_h(V) = \frac{\beta_{h,0}}{1 + \beta_{h,1} \exp(\beta_{h,2} V)}$$

Parameters  $\alpha_{i,j}$  and  $\beta_{i,j}$  are given in **Table 1** for both axonal and somatic channels. In single-compartment simulations, somatic parameters were employed.

**Table 1 | Activation parameters for somatic and axonal sodium channels in the Schmidt-Hieber and Bischofberger model.**

Parameter	Somatic channels	Axonal channels	Parameter	Somatic channels	Axonal channels
$\alpha_{1,0}(\text{ms}^{-1})$	45.850	62.648	$\beta_{1,0}(\text{ms}^{-1})$	0.0144	0.00194
$\alpha_{1,1}(\text{mV}^{-1})$	0.00239	0.0116	$\beta_{1,1}(\text{mV}^{-1})$	0.0885	0.1377
$\alpha_{2,0}(\text{ms}^{-1})$	19.808	34.783	$\beta_{2,0}(\text{ms}^{-1})$	0.5650	0.0957
$\alpha_{2,1}(\text{mV}^{-1})$	0.02218	0.0299	$\beta_{2,1}(\text{mV}^{-1})$	0.06108	0.0928
$\alpha_{3,0}(\text{ms}^{-1})$	71.812	76.698	$\beta_{3,0}(\text{ms}^{-1})$	0.7531	1.2488
$\alpha_{3,1}(\text{mV}^{-1})$	0.0659	0.0537	$\beta_{3,1}(\text{mV}^{-1})$	0.0365	0.0311
$\alpha_{h,0}(\text{ms}^{-1})$	0.5757	6.882	$\beta_{h,0}(\text{ms}^{-1})$	2.8301	3.573
$\alpha_{h,1}$	162.84	4654.0	$\beta_{h,1}$	0.289	01933
$\alpha_{h,2}(\text{mV}^{-1})$	0.0268	0.0296	$\beta_{h,2}(\text{mV}^{-1})$	0.0696	0.07496

Potassium channels in SB model are simulated by the same kinetic scheme as HH model (scheme 2) with the following voltage dependent transition rates:

$$\alpha_n(V) = sc \times 0.01 \frac{V + 55}{1 - \exp\left(-\frac{V+55}{10}\right)}$$

$$\beta_n(V) = sc \times 0.125 \exp\left(-\frac{V + 65}{80}\right)$$

where  $sc$  is a scale parameter that adjusts the kinetic constants according to the compartments in which the channels are being simulated. For single compartment simulations, equation (1) was used with the following parameters:  $C_m = 1\mu\text{F}/\text{cm}^2$ ,  $\overline{g_{Na}} = 20\text{ mS}/\text{cm}^2$ ,  $\overline{g_K} = 4\text{ mS}/\text{cm}^2$ ,  $g_l = 0.1\text{ mS}/\text{cm}^2$ ,  $E_{Na} = 75\text{ mV}$ ,  $E_K = -95\text{ mV}$ ,  $E_l = -70\text{ mV}$ . With this model, the number of channels was controlled by the membrane area, given a unitary conductance of  $20\text{ pS}/\text{cm}^2$ . We tested areas ranging from  $15.7$  to  $628\mu\text{m}^2$ , resulting in 157 to 6283 sodium channels and 31 to 1257 potassium channels.

With this model, the following tests were conducted at different values of membrane area:

- (a) 20-s Iclamp with 1-ms stimulus: the stimulus current was applied with 1 ms delay. Afterwards, an additional 18 ms were simulated and the occurrence and timing of an action potential was recorded. The current amplitude varied from 0 to  $8\mu\text{A}/\text{cm}^2$  and 10,000 simulations were performed for each amplitude. Then, the firing efficiency, mean action potential time and variance of action potential time were calculated.
- (b) Voltage clamp with action potential trace: the same procedure described for the HH model.

**Schmidt-Hieber and Bischofberger model—idealized multicompartment model**

An idealized model similar to the one described in Schmidt-Hieber and Bischofberger (2010, see **Figure 5A**) was simulated using stochastic algorithms for the ion channels. The parameters for ion channel densities and kinetic constants were used as described in the article with minor modifications, such as the absence of an axonal bleb and a longer axon for some simulations.

**SIMULATION ALGORITHMS**

The methods described and tested here are designed to simulate a number of independent and identical Markov Chains (MCs) with a discrete number of states, keeping track of the number of channels in any state at any given time. For the description of the algorithms, we denote  $S$  as the total number of states in a MC,  $i \in \{1, \dots, S\}$  are the individual states, and  $N_i$  is the number of MCs in state  $i$ .

**Markov chain simulations (MC)**

Markov chains were simulated using the Stochastic Simulation Algorithm (SSA) (Gillespie, 1976) with some modifications. Briefly, the method consists in:

At time  $t$ , calculate the effective transition rate  $\lambda(t)$  as

$$\lambda(t) = \sum_i^S N_i(t)\zeta_i(t)$$

where  $\zeta_i(t)$  is the sum of transition rates for transitions escaping from state  $i$ .

Calculate the time for the next transition  $t_n$  as

$$t_n = t_p - \frac{\log \xi_1}{\lambda(t)}$$

where  $t_p$  is the time of the previous transition (0 at the beginning of the simulation) and  $\xi_1$  is a random number uniformly distributed within  $[0,1]$ , drawn after the previous transition.

If  $t_n > t$ , continue integrating the time and the membrane voltage equation.

If  $t_n < t$ , perform a transition:

Calculate the probability of all  $j$  transitions:

$$P_j(t) = \frac{N_i(t)\alpha_j(t)}{\sum_j N_i(t)\alpha_j(t)}$$

where  $i$  is the state originating transition  $j$  and  $\alpha_j$  its rate.

Build a cumulative sum of all transition probabilities. Draw a random number  $\xi_2$  uniformly distributed in  $[0,1]$  and find the first term in the cumulative probability that is greater than  $\xi_2$ .

Execute the transition indicated by the term found in the previous step, and draw a new random number  $\xi_1$  to be used for the time of the next transition.

**Unbounded diffusion approximation (UA) (Orio and Soudry, 2012)**

The DA algorithm was implemented with SDEs described previously (Orio and Soudry, 2012; see also Mélykúti et al., 2010). In matrix form, the equations for the sodium and potassium channels are, respectively:

$$\begin{aligned} \frac{dX_{Na}}{dt} &= A_{Na}X_{Na} + \frac{1}{\sqrt{N_{Na}}}S_{Na}(X_{Na})\xi(t)_{Na} \\ \frac{dX_K}{dt} &= A_KX_K + \frac{1}{\sqrt{N_K}}S_K(X_K)\xi(t)_K \end{aligned} \quad (2)$$

where  $X_{Na} = [m_0h_0 \ m_1h_0 \ m_2h_0 \ m_3h_0 \ m_0h_1 \ m_1h_1 \ m_2h_1 \ m_3h_1]^T$  and  $X_K = [n_0 \ n_1 \ n_2 \ n_3 \ n_4]^T$  are column vectors with the fraction of channels at any given state, and  $\xi(t)_{Na}$  and  $\xi(t)_K$  are column vectors of independent normally distributed random variables (mean 0, variance 1) with length 10 and 4, respectively.  $N_{Na}$  and  $N_K$  are the number of sodium and potassium channels, respectively. The rate matrices  $A_{Na}$  and  $A_K$  and square root matrices  $S_{Na}(X_{Na})$  and  $S_K(X_K)$  can be directly found from the state diagram of the corresponding ion channel type. This is explained in detail around equations 1 and 13 and the Supplemental Material in Orio and Soudry (2012). For example, in the case of Potassium channels we have

$$A_K = \begin{bmatrix} -4\alpha_n & \beta_n & 0 & 0 & 0 \\ 4\alpha_n & -3\alpha_n - \beta_n & 2\beta_n & 0 & 0 \\ 0 & 3\alpha_n & -2\alpha_n - 2\beta_n & 3\beta_n & 0 \\ 0 & 0 & 2\alpha_n & -\alpha_n - 3\beta_n & 4\beta_n \\ 0 & 0 & 0 & \alpha_n & -4\beta_n \end{bmatrix}$$

and

$$S_K(X_K) = \begin{bmatrix} \sqrt{4\alpha_n n_0 + \beta_n n_1} & 0 \\ -\sqrt{4\alpha_n n_0 + \beta_n n_1} & \sqrt{3\alpha_n n_1 + 2\beta_n n_2} \\ 0 & -\sqrt{3\alpha_n n_1 + 2\beta_n n_2} \\ 0 & 0 \\ 0 & 0 \\ & 0 & 0 \\ & 0 & 0 \\ & \sqrt{2\alpha_n n_2 + 3\beta_n n_3} & 0 \\ -\sqrt{2\alpha_n n_2 + 3\beta_n n_3} & \sqrt{\alpha_n n_3 + 4\beta_n n_4} \\ 0 & -\sqrt{\alpha_n n_3 + 4\beta_n n_4} \end{bmatrix}$$

Note that in the case of the Schmidt-Hieber and Bischofberger model, the only difference from the HH model is that the rate constants for the sodium channel equations are different.

To take care of normalization, variables  $m_1h_0 \dots m_3h_1$  and  $n_1 \dots n_4$  were advanced by an Euler-Maruyama scheme and the remaining two were calculated as  $m_0h_0 = 1 - m_1h_0 - m_2h_0 - m_3h_0 - m_0h_1 - m_1h_1 - m_2h_1 - m_3h_1$  and  $n_0 = 1 - n_1 - n_2 - n_3 - n_4$ . As we do not control the bounding of the variables between 0 and 1, in order to ensure real-valued square roots we calculated the stochastic terms  $S(X)$  (and only those terms) taking

the absolute value of the variables. Thus, we refer to this algorithm as UA – Unbounded with Absolute values in stochastic terms.

**Reflected SDEs (Ref) (Dangerfield et al., 2012)**

This method aims to normalize the variables  $m_1h_0 \dots m_3h_1$  and  $n_1 \dots n_4$  and to keep them bounded in the interval [0, 1] using the reflected stochastic equation approach, described in section V of Dangerfield et al. (2012). We have used this method together with the DA equation system (Equations 2).

**Truncated and restored DA (TR) (Huang et al., 2013a)**

We used the DA equations (Equations 2) with an additional residual term. This residual term was introduced to ensure the boundary and normalization constraints, as explained in section II.F in Huang et al. (2013a).

**Stochastic shielding approximation (SSmc) (Schmandt and Galan, 2012)**

In this method, transitions not connecting to the conducting states are approximated to be deterministic and solved as ODEs. Transitions connecting conducting with non-conducting states are solved as Markov Chains with the already mentioned algorithm. As there is a mixture of continuous (ODEs) and discrete (MC) treatment of variables, violations of the constraints occur. In our implementation, and inspired by the code by Schmandt and Galan, variables going off the [0,N] boundary are manually corrected and normalization was performed as in the UA algorithm.

**Stochastic shielding approximation with DA (SSda)**

We modified the Schmandt and Galan (2012) approach by calculating the stochastic transitions with a DA approach (Equations 2) rather than using MCs. Therefore, SSda similarly uses the same DA equations, but the stochastic terms related to transitions not connecting to the conducting states were neglected.

For example, for potassium channels, we now used

$$S_K(X_K) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sqrt{\alpha_n n_3 + 4\beta_n n_4} \\ 0 & 0 & 0 & -\sqrt{\alpha_n n_3 + 4\beta_n n_4} \end{bmatrix},$$

Thus, we needed only 2 Brownian terms for sodium channels and 1 for potassium channels. Boundary and normalization constraints were again treated as in the UA algorithm, that is: there was no bounding of the variables; the absolute value of the variables was used in the square roots of the stochastic terms; and normalization was applied by calculating  $m_0h_0 = 1 - m_1h_0 - m_2h_0 - m_3h_0 - m_0h_1 - m_1h_1 - m_2h_1 - m_3h_1$  and  $n_0 = 1 - n_1 - n_2 - n_3 - n_4$ .

**HH with colored noise terms (CN) (Güler, 2013)**

We simulated the Güler’s Brownian harmonic oscillator using the system of stochastic differential equations 6.1, 6.2 in Güler (2013), using the constant parameters given in Table 2 in Güler (2013). Note that these equations are very different from the standard DA equations.

To take care of the normalization constraint at each time step, if any of the variables  $m$ ,  $h$  or  $n$ , left the  $[0,1]$  interval, then the stochastic term  $\eta$  was redrawn until the variable fulfilled the boundary constraint.

## SOFTWARE

All the models and simulations algorithms presented here were implemented and run in the Neuron simulation environment (Hines and Carnevale, 2001; Carnevale and Hines, 2006). The different algorithms were written inside the MOD files for each channel. With the exception of TR, all algorithms run fine regardless of the numeric integrator specified (cnexp or euler). They also produce the same results in Python using the Euler-Maruyama integration method (tested in some selected cases). The TR algorithm required the specification of the Euler integrator within the MOD file to produce the results presented here, otherwise a much lower firing rate was obtained. Simulation control and the recording of variables were specified with Python scripts (Hines et al., 2009). Sample codes and .mod files can be found in ModelDB (<http://senselab.med.yale.edu/ModelDB/Accession167772>).

Data analysis and plotting was performed using the Python libraries numpy, scipy, and matplotlib.

## RESULTS

To test the accuracy of the methods we performed a series of simulations, comparing the variability of the results to that obtained with explicit MCs solved by the exact Gillespie algorithm (Gillespie, 1976, 2007). We employed the original Hodgkin and Huxley model (Hodgkin and Huxley, 1952), in order to reproduce previously published comparisons of the algorithms. In addition we performed some tests with a faster model, based on sodium channels from granular cells in the hippocampus (Schmidt-Hieber and Bischofberger, 2010). Finally, we simulated a model neuron with multiple compartments (Schmidt-Hieber and Bischofberger, 2010) and measured the variability in the generation and conduction of action potentials.

### SIMULATIONS WITH THE ORIGINAL HODGKIN AND HUXLEY MODEL

#### Firing variability—15 ms simulation with stimulus

A widely used test to compare stochastic simulation algorithms (Mino et al., 2002; Bruce, 2007; Orío and Soudry, 2012 and others) consists of a short simulation (15 ms) in which a 2-ms current stimulus is given after a 1-ms delay (Figure 1A). Depending on the amplitude of the stimulus the probability of eliciting an action potential increases, and this relationship depends on the number of channels. Figure 1B shows the probability of firing an action potential in 10,000 trials at different stimulus amplitudes, for the algorithms tested with  $N_{Na} = 5000$ . Figure 1C plots the variance of action potential timing, a measure of jitter. The Reflection method produces a higher firing probability at all stimulus amplitudes. This entails a lower variability in action potential timing. Additionally, Güler's CN method produces a higher variability than MC and other methods. This result is repeated with higher number of channels, however at a lower number of channels the spontaneous firing of action potentials makes the comparison unreliable. To compare the behavior of the models with  $N_{Na} \leq 1600$ , we modified the protocol so the stimulus is sustained during

the simulation (Figure 1D, inset) and explored negative values of current amplitude. As can be seen in Figures 1D,E, the behavior of all the DA algorithms (as well as of SSmc) deviates from MC considerably for  $N_{Na} = 50$  and to a minor degree for  $N_{Na} = 500$ . Again, the Reflection method produces a higher firing probability than the other methods.

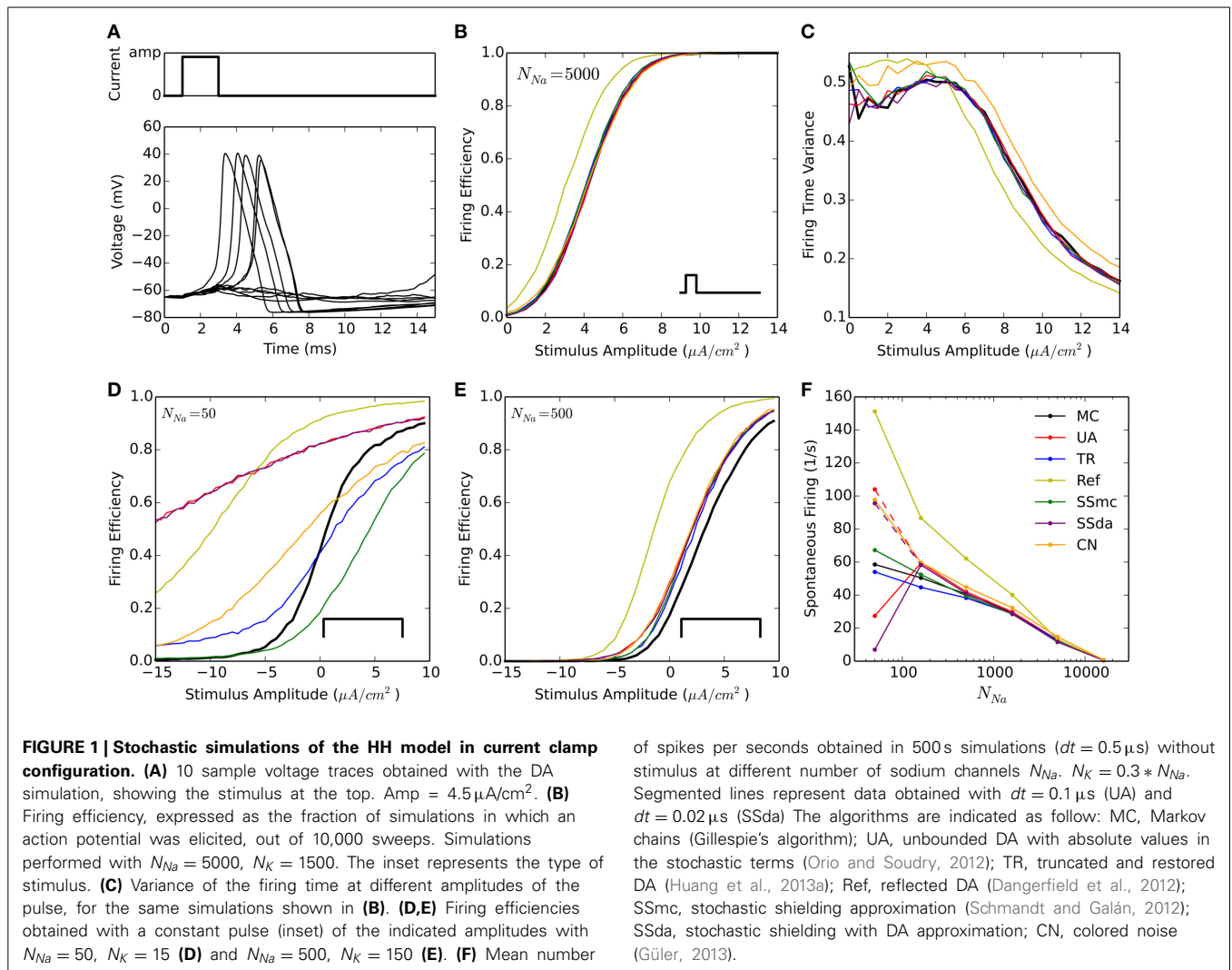
#### Spontaneous firing rate

The original HH model with stochastic ion channel produces spontaneous firing activity that increases as the number of channels is decreased. With each simulation algorithm and with sodium channel number ( $N_{Na}$ ) ranging from 50 to 50,000, we simulated 500 s and recorded the occurrence of action potentials. Figure 1F shows the mean frequency of spikes that were detected in the simulations. With  $N_{Na} \geq 1600$ , almost all the methods reproduce the behavior of MC modeling. The sole exception is the Reflection method, that showed a higher firing probability at  $N_{Na} = 1600$  and below. To discard some incompatibility of the Neuron simulation environment with the reflection procedure, we repeated this simulation using an Euler-Maruyama integration procedure written in Python and obtained the same result. Huang's truncated and restored DA method seems to be the one that more closely follows MC modeling at extremely low number of channels, only slightly underestimating the firing rate. At  $N_{Na} = 160$ , the Unbound DA and Stochastic Shielding methods overestimate the firing rate, dropping abruptly when  $N_{Na} = 50$ . This latter behavior is actually due to numeric overflows that made the simulations run without producing action potentials. This was corrected using a smaller time step (Figure 1F, segmented lines). Thus, the Unbound DA (note that SSda is also unbound) becomes numerically unstable when the number of channels is too low.

#### Voltage clamp—noisy voltage trace with action potential

A third test, to check how the different DA methods can reproduce the variability of channel openings obtained with MC modeling, consists on recording the response of the model channels to a fixed voltage trajectory obtained from a stochastic simulation. The voltage trace is shown in Figure 2A and it contains an action potential as well as a noisy background (zoomed in Figure 2B). With each model and condition, 2000 independent simulations were run and the time evolution of open channels was recorded. At each point in time, the mean and variance of the open channels was calculated. In addition to the comparison with the behavior of MC simulation, we compared to the expected mean of open channels which is calculated by applying the same voltage clamp simulation to deterministic HH channels. Moreover, we can compute the expected variance as explained in Methods.

The results are shown in Figure 2. During the subthreshold regime, Reflection method overestimates the mean of open channels, both for sodium and potassium (Figures 2C,D, left). Huang's TR algorithm also overestimates the mean of open sodium channels (Figure 2C, left) to a minor extent. However, during the action potential any difference between the DA methods and the MC modeling or the exact solution appears to be negligible (Figures 2C,D, right). Regarding the variance of



the open channels (Figures 2E,F), the main deviation seems to occur with Güler's CN algorithm, which overestimates the variance of both open sodium and open potassium channels, during the subthreshold regime (left) and the action potential (right). Schmandt's stochastic shielding approximation (both SSmc and SSda) underestimates the variance of open channels during the action potential, when the voltage changes more rapidly. The results shown here are for  $N_{Na} = 500$ ,  $N_K = 160$ ; with higher number of channels ( $N_{Na} = 5000$ ,  $N_K = 1600$ ) we naturally observed less fluctuations but the results maintained: Reflected DA overestimates the mean of open channels in subthreshold regime; Colored Noise overestimates the variance of open channels in subthreshold regime; and Stochastic Shielding underestimates the variance of open channels during the action potential (not shown).

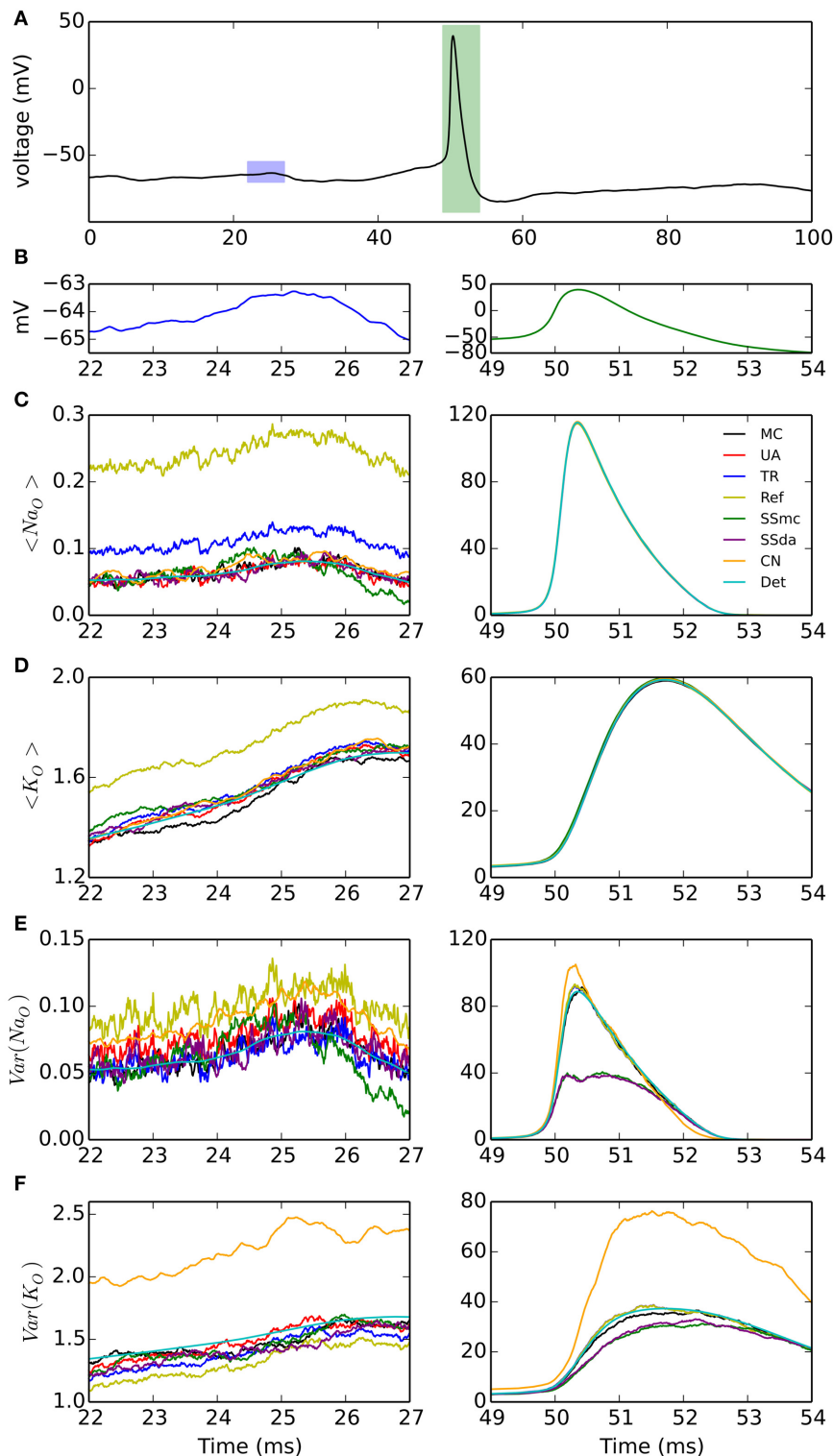
### SCHMIDT-HIEBER AND BISCHOFBERGER MODEL—SINGLE COMPARTMENT

We decided to use a model with faster sodium channels, resembling mammalian ion channels, to test the accuracy of the DA

methods when the transitions between states occur at faster rates. We chose a recently published model that focuses on the fast opening of sodium channel in the axon initial segment of granule cells from the hippocampus (Schmidt-Hieber and Bischofberger, 2010). We will refer to this model as the "SB" model. We noted that the SB model does not show spontaneous firing when simulated stochastically. Therefore, the 500-ms simulation test was not performed.

### Firing variability—20 ms with 1 ms stimulus

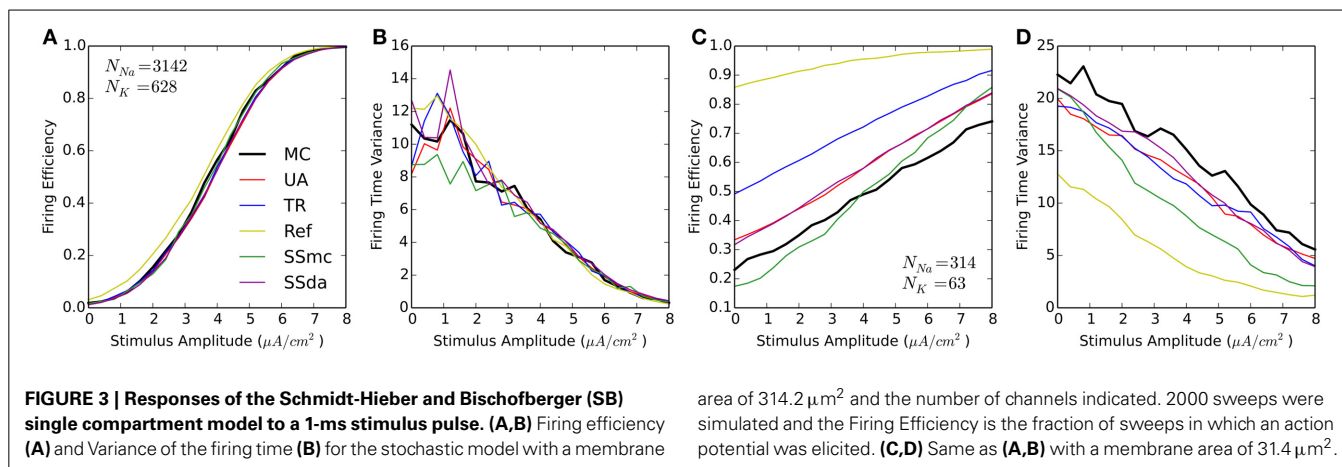
We performed the test in which 20-ms were simulated with a 1-ms current stimulus (similar to Figure 1A). For each stimulus amplitude, 2000 simulations were run and the Firing Efficiency, mean firing time, and firing time variance were calculated. Figure 3 shows that the results are similar to that obtained with HH. At high number of channels, most methods perform reasonably similar to MC with a higher excitability of the Reflection method (Figures 3A,B). However, when the number of channels is low all DA methods fail to approximate the results of MC, showing a much higher probability of firing at all amplitudes of the stimulus



**FIGURE 2 | Simulations of the stochastic HH model under voltage clamp. (A)** Voltage trace applied to simulated channels in a 100-ms simulation, repeated 2000 times. Blue and green rectangles represent the 5-ms intervals that are expanded in the left and right columns of the figure, respectively. **(B)** Detail of the voltage traces corresponding to the time windows analyzed in **(C–F)**. Note the different vertical

scales. **(C)** Mean of open sodium channels during the subthreshold (left) and action potential (right) regimes, for the simulation algorithms tested. **(D)** Mean of open potassium channels. **(E)** Variance of open sodium channels. **(F)** Variance of open potassium channels.  $N_{Na} = 500$ ,  $N_K = 160$ . Very similar results were obtained with  $N_{Na} = 5000$ ,  $N_K = 1600$  (see text).





(Figure 3C). Also, the Reflection method shows a lower firing time variability (Figure 3D).

### Voltage clamp

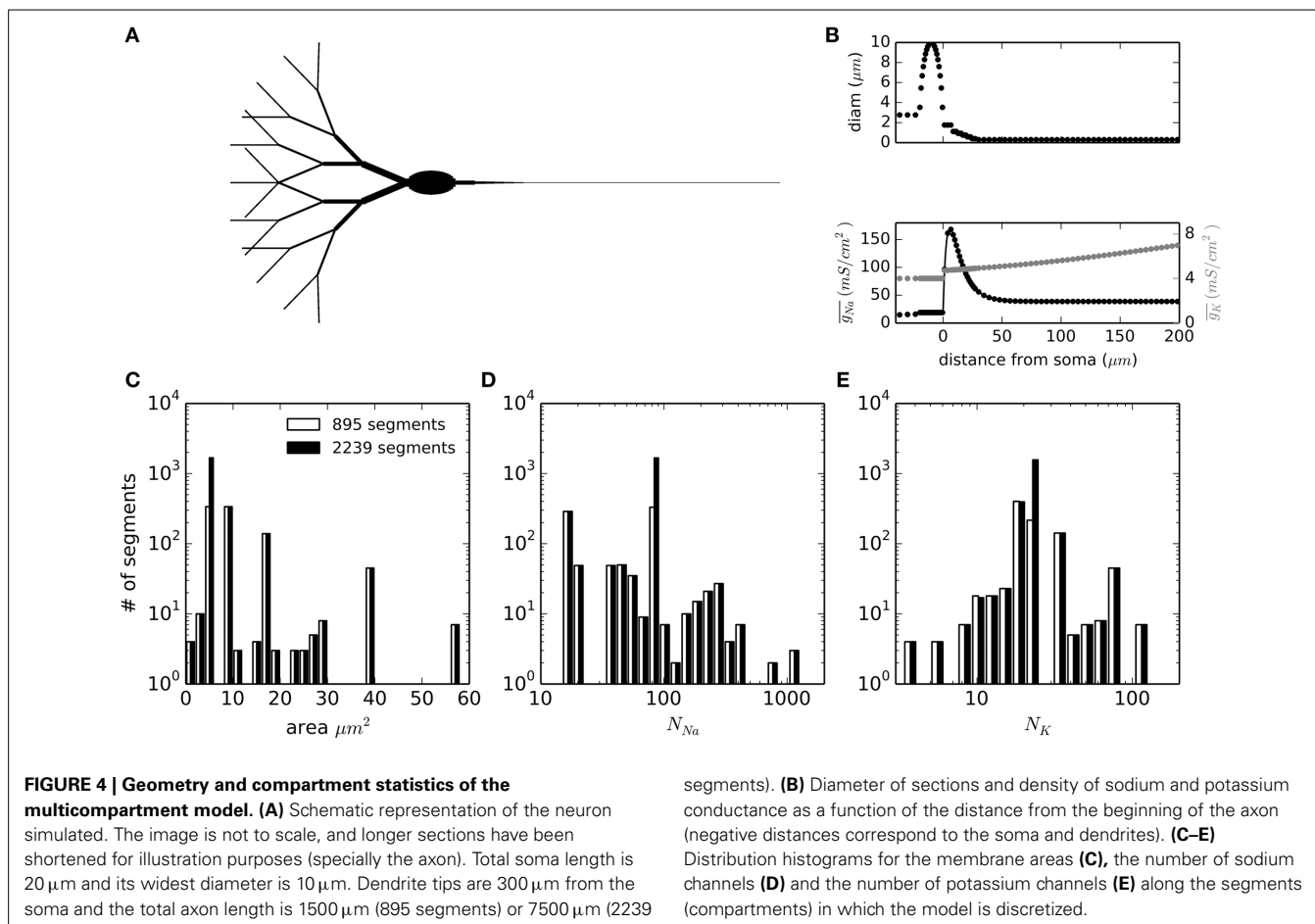
We applied the voltage clamp test with the same voltage trace as the HH model to the SB model stochastic channels. Similar to what we observed for HH model, the Reflection method overestimates the mean of open channels (both sodium and potassium) during the subthreshold regime. Huang’s truncated and restored method also overestimates it to a minor degree. During the action potential, the greatest deviation occurs with the stochastic shielding approximation, which underestimates the variance for both channels. With a higher number of channels, we observed similar results, with the exception of Huang’s TR method performing better in the mean of open channels (not shown).

### SCHMIDT-HIEBER AND BISCHOFBERGER MODEL—MULTI-COMPARTMENT SIMULATIONS

To test the applicability of DA methods in more complex simulations of physiological relevance, we set up a multi-compartmental model of a neuron. We chose the idealized neuron described in Schmidt-Hieber and Bischofberger (2010) and shown in Figure 4A. Moreover, we kept the particular inhomogeneous sodium channel density for the axon that causes the action potentials to be initiated in the axon initial segment (AIS), about 10  $\mu\text{m}$  from the soma (Figure 4B, bottom). The neuronal sections were spatially discretized according to their spatial constant  $\lambda$ , with a further increase in the number of segments in the AIS area. In total, the number of segments simulated were 895 with a 1500  $\mu\text{m}$  axon and 2239 when the axon was extended to 7500  $\mu\text{m}$ . The distributions of segment areas is shown in Figure 4C. Together with the different ion channel densities and a unit conductance of 20 pS, the resulting distributions of number of channels per segment are shown in Figures 4D,E. It is noteworthy that the number of channels to be simulated in any given segment is rarely higher than 500 for sodium channels and never higher than 120 for potassium channels. In the model with the long axon, this adds up to 171189 sodium channels and 50670 potassium channels.

The model neuron with different stochastic channels was subject to a current clamp stimulus applied to the soma. The stimulus consisted in a 2 s noisy stimulus (Figure 5A) which in a deterministic simulation elicited 8 action potentials (represented by stars in the Figure). 400 independent simulations were performed with the same stimulus and Figures 5B,C show the raster plots (100 simulations) of spikes detected at the soma and at the tip of the axon, respectively. Figure 5D depicts a normalized firing probability calculated for the spikes at the tip of the axon. Both raster and firing probability plots show that the simulations with the Reflection method displayed a greater excitability, as several action potentials were only elicited with this algorithm and were not seen with the other DA methods or were seen with a much lower probability (i.e., around  $t = 1300$  ms,  $t = 1700$  ms and near the end of the trace). On the other hand, Stochastic Shielding with MC produced a lower excitability, firing near half of the action potentials per sweep than the other methods. The mean of spikes per sweep (Figure 6A) was significantly different to MC for all the algorithms, not only the Reflection and SSmc methods. With the Reflection method, however, the deviations from the other methods go beyond a higher excitability. Some spikes fired with high probability with all DA methods except for Reflection (see for instance around 800 ms), and some spikes had a slightly different timing with Reflection (700, 1500 ms). Therefore, the Reflection method in this test actually introduced a bias, producing spikes with different timings than the other DA method. The variability of the number of spikes elicited per trial varied with some DA algorithms compared to MC (Figure 6B) but only in the case of UA and SSmc a significant difference was observed.

We took the raster plots and searched for spikes that were repeated in at least 50% of the sweeps with the same timing  $\pm 5$  ms. These were called “common spikes” and for most algorithms 10–11 common spikes were found (Figure 6C), with the exception of Reflection and SSmc methods. Then, we measured how variable the timing of these spikes was at two axon locations, one near the initiation site and the other at the tip (Figure 6D). Although the TR method reproduced more closely the variability obtained with MCs, none of the observed differences was significant.



As a measure of variability with functional consequences, we examined how the duration of the action potential (duration measured at the detection threshold level of  $0\ \text{mV}$ ) evolved as it propagates along the axon. The relevance of this measure is that the duration of action potentials at the release zone of a synapse will impact the amount of neurotransmitter released. **Figure 7A** plots the duration of all the action potentials recorded in a MC simulation at several sites of the axon, plotted against the duration at a site near the initiation. As a first observation, in the reference site there is a wide distribution of action potential durations, which gets narrower as the measurement site moves along the axon. Also, action potentials are shorter in the distal axon than in the initiation site and the Reflection method produces the shorter action potentials of all the methods (**Figure 7B**).

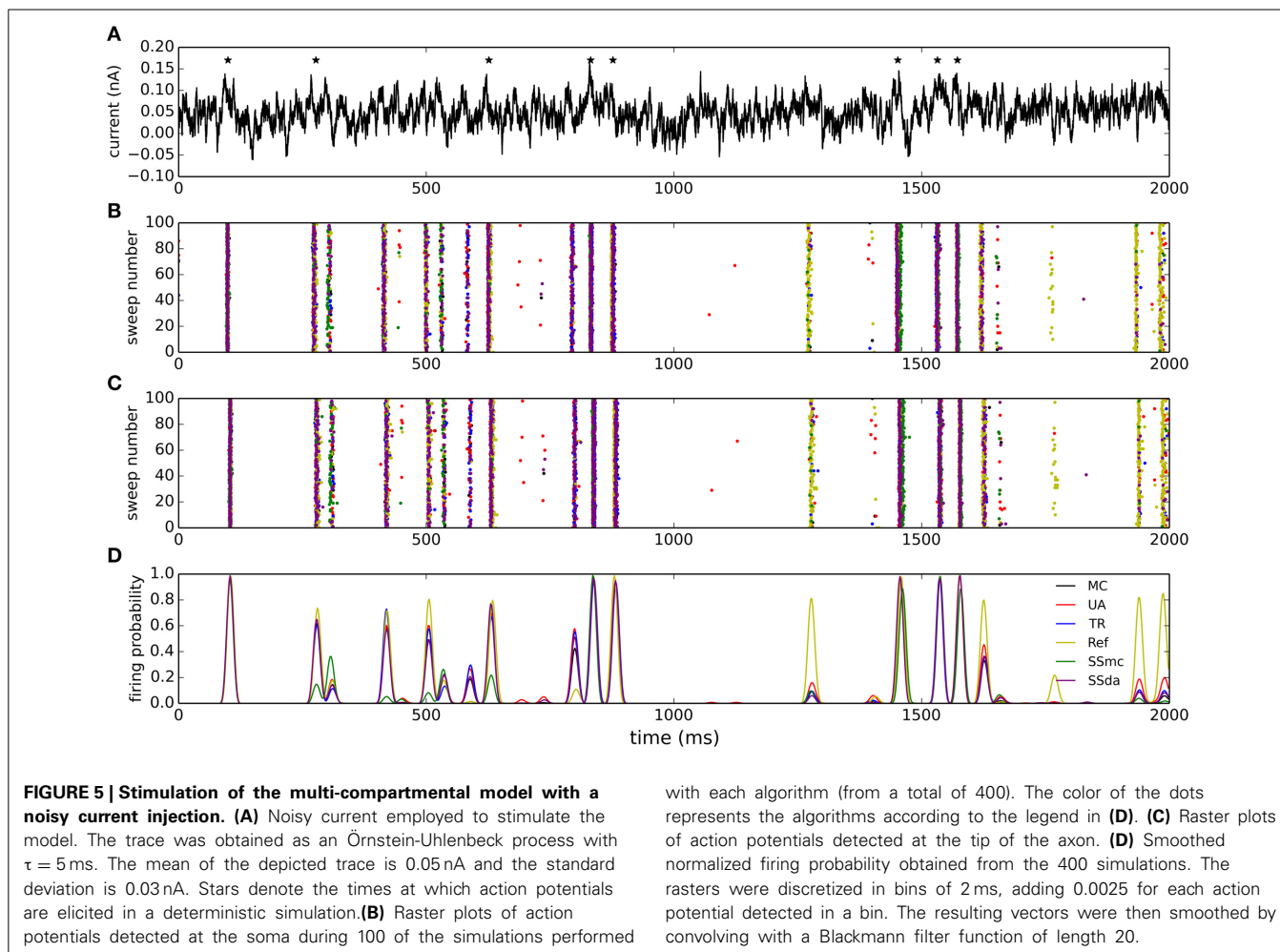
Besides getting shorter, action potential duration at the distal axon is completely uncorrelated to the duration at the initiation site (**Figure 7A**, right). We looked at how the correlation of action potential duration decays along the axon with the different stochastic simulation algorithms. Results are shown in **Figure 7C**. It is apparent that Reflection method produces a faster decay in the correlation, while the Unbound DA produces a longer propagation of correlation. To test for similarity, we fitted an exponential decay to the data points, obtaining a space constant  $\lambda$ . An extra sum of squares  $F$  test was performed to test the null hypothesis that the data points of each set could be fitted with the

same  $\lambda$  as the MC data, showing that Unbound DA, Reflection and SSda methods produced a behavior significantly different to that obtained with MC (**Figure 7C**, inset). When the test was repeated with a fit to a double exponential decay, the same result was obtained.

#### COMPUTATION TIME

To account for the usefulness of the simulation algorithms, we found important to compare the computational cost of each of them. **Figures 8A,B** show the time required to simulate  $500\ \text{ms}$  of the HH model with an integration time step ( $dt$ ) of  $5$  and  $0.5\ \mu\text{s}$ , respectively, for each algorithm used. As we reported previously (Orio and Soudry, 2012), DA methods are highly sensitive to  $dt$  but mostly insensitive to the number of channels. On the other hand, MC simulations are sensitive to both, but its sensitivity to the number of channels approaches a linear relationship as the number of channel increases. Importantly, as both  $dt$  and/or number of channel decrease, MC outperforms all of the DA methods, giving the best simulation times precisely in the condition where the DA methods showed to be more problematic.

In the case of the multi-compartment simulation, MC is faster than all DA methods except for Schmandt's stochastic shielding approximation (**Figure 8C**). This result is not surprising, given that the number of channels per simulated compartment was almost always below  $1000$  (**Figures 6C,D**).



## DISCUSSION

In this work we numerically tested five Diffusion Approximation algorithms proposed to reproduce the behavior of a number of simultaneous Markov Chains, in the context of stochastic ion channel modeling. Most of these algorithms (Dangerfield et al., 2012; Orío and Soudry, 2012; Schmandt and Galán, 2012; Huang et al., 2013a) are based on a Langevin Equation proposed for stochastic modeling of the Hodgkin and Huxley model (Fox and Lu, 1994; Fox, 1997). However, they deal in different ways with numerical issues that appear in the simulation of stochastic trajectories: the requirement that the variables be bounded in  $[0,1]$  (“boundary constraint”) and the requirement that at any given time the sum of variables must be equal to 1 (“normalization constraint”). The boundary constraint breaks more frequently when the number of channels being simulated is low or when the integration time step increases, because the stochastic terms scale with  $\sqrt{dt/N}$ . Therefore, special attention should be put to the performance of these simulation algorithms—and their capacity of reproducing MC behavior—with a low number of channels.

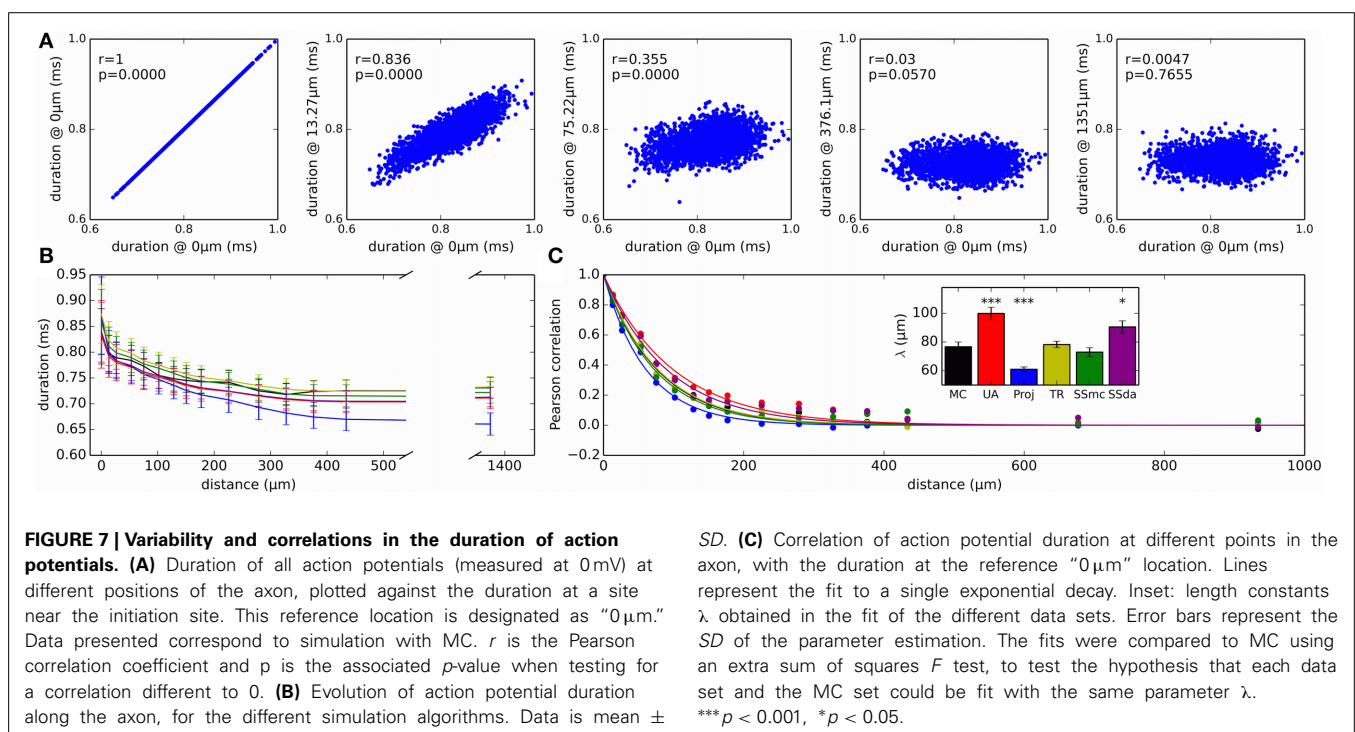
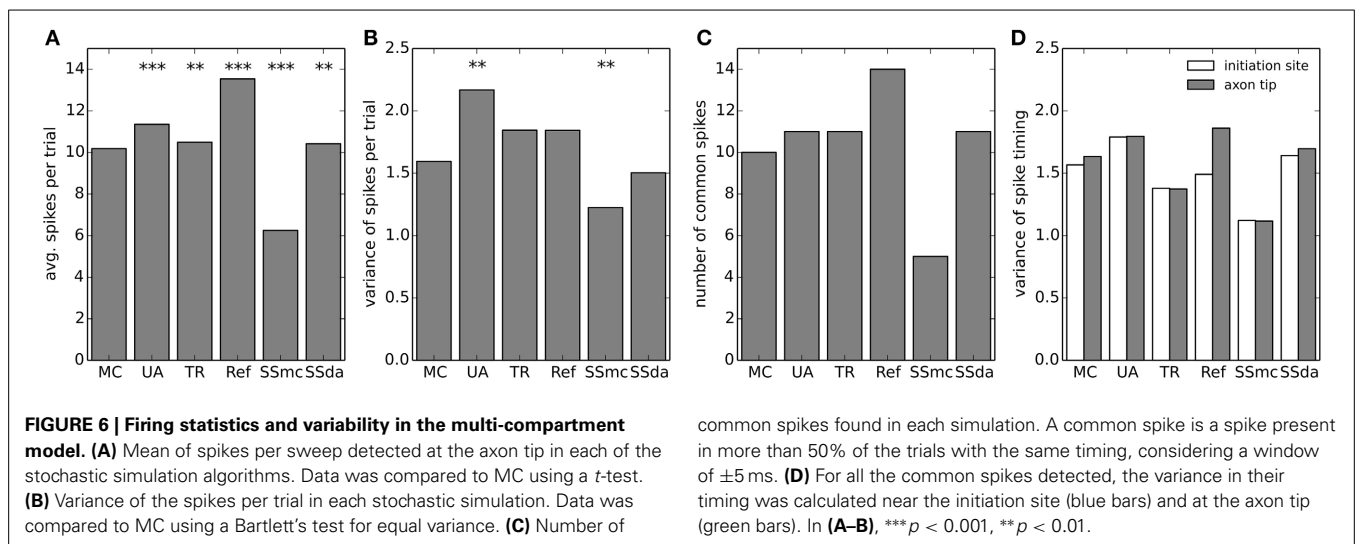
Our tests were primarily aimed with a practical question in mind: what method should be used to study the contribution of channel stochasticity to neural excitability in any given specific context? To answer this question, the simple test of the

numerical accuracy is not enough. Modeling algorithms should also be examined for applicability and simplicity of implementation in different contexts and for the computational efficiency for the intended model to be solved.

## ACCURACY

Regarding accuracy, in brief we found that all DA algorithms fail in the reproduction of MC behavior when the number of channels is low (generally speaking, below 1000), with no clear “winner.” For example, the TR procedure improved the results in some current clamp simulations (Figures 1D–F), but introduced higher deviations than the UA in others (Figure 3). The Reflection method in our simulations performed the worse, introducing a higher firing probability in current clamp simulations. Most likely, this is related to the higher mean of open sodium channels observed in voltage clamp (Figure 2C).

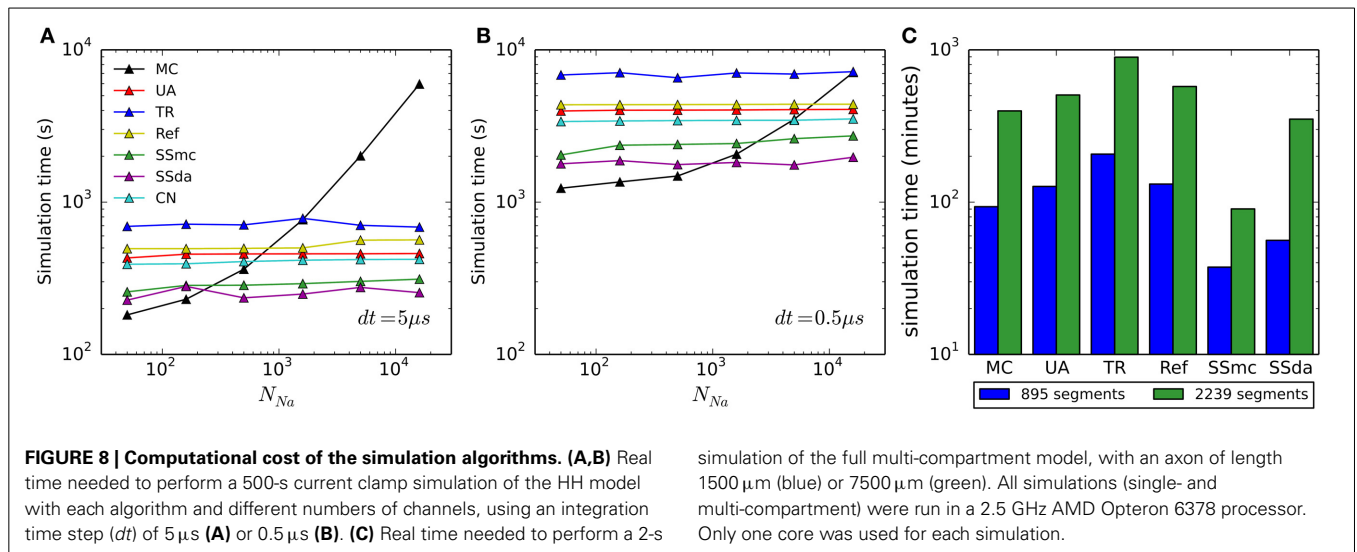
When the number of channels is 5000 or higher, the TR, UA and SS methods perform well in reproducing MC behavior in current clamp tests. This means that in the high number of channels regime, bounding the variables to  $[0,1]$  is not essential, as the UA and SSda implementations (both unbounded) give the same results than TR. Inaccuracies were observed (Figure 1) in the Reflection method, and, to a lesser extent, the CN method.



Additionally, both methods showed inaccuracy in the voltage clamp tests. Also on these tests, the SS methods showed some inaccuracy in the variance of open channels during the action potential, but the results in current clamp simulations suggest that this may not be relevant for the neural excitability.

The results with the multicompartment model deserve special attention, because some results were different to what was observed in a single compartment. Although the SSmc method introduced only minor inaccuracies in the single compartment test, this method severely altered the excitability of the multicompartment model. We could not identify the reason for this effect, and tested several alternatives of bounding and normalization

which did not improve the results. Nevertheless, the overall effect is a reduced excitability, which is consistent with the deviation observed in single compartment (Figure 1D). We confirmed that this effect does not arise from our implementation of the algorithm, by repeating the single compartment test with the Matlab code published by Schmandt and Galán (ModelDB acc. 144468) and obtaining similar results (not shown). So, although it is an attractive method for increasing simulation speed in multicompartmental models, its use is not recommended until further testing is performed. The Reflection method altered the excitability of the neuron but the results were in agreement with the single compartment results. Curiously, the use of Stochastic Shielding in



a DA framework (SSda) improves the behavior of the simulations, bringing it closer to the behavior of MCs, like the TR does.

**APPLICABILITY AND SIMPLICITY**

In terms of applicability, a first observation is that the colored noise approach (Güler, 2013) can be used only if the ion channel is composed of independent subunits. This is sometimes true (e.g., the original HH model), but not always (e.g., the SB model). For example, channels with non-identical voltage sensors (Vandenberg and Bezanilla, 1991; Horn et al., 2000), cooperativity in the movement of voltage sensors (Bezanilla et al., 1994; Schoppa and Sigworth, 1998) or complex allosteric gating mechanisms (Horrigan and Aldrich, 2002) do not have independent subunits. Therefore, they cannot be modeled with the colored noise approach. Besides, Güler’s equations have a number of constant parameters ( $\gamma_K, \gamma_{Na}, \omega_K^2, \omega_{Na}^2, T_K, T_{Na}$ ) that were estimated empirically to obtain an adequate level of channel noise (Güler, 2013). It is not clear how these parameters can be derived for other ion channels, even if they are composed of identical and independent gating subunits.

The other DA-based algorithms can be applied to any given kinetic scheme but first require to obtain the corresponding system of SDEs. Until recently, it seemed quite complicated to implement, since the original descriptions involved the calculation of a matrix root square (Fox and Lu, 1994; Goldwyn et al., 2011). However, alternative derivations of the Langevin equation (Mélykúti et al., 2010; Orio and Soudry, 2012) yield an explicit form that does not use complex matrix operations. This method can be derived for any given kinetic scheme using simple and intuitive rules without using matrix notation (see Supplemental Material in Orio and Soudry, 2012). Importantly, any kinetic scheme can be translated to an SDE system and the equations can be written explicitly. This makes it simpler to employ low-level or limited languages such as C or Neuron’s NMODL, which are compiled prior to execution code and therefore run faster.

The DA algorithms differ in the treatment of boundary and normalization constraints. In this regard, the Unbound DA (UA) is the simplest, not taking care of the boundaries issue and doing

a simple normalization by making one variable to depend on the others (alternative normalization procedures can be implemented, for instance dividing all the variable values by their sum). Finally, to avoid non-real square roots in the stochastic terms, a simple absolute value operation is performed. The Stochastic Shielding approximation, when used with DA equations (SSda), can simplify the code even further because it uses less stochastic terms. Huang’s truncation and restoration is also rather simple to follow and implement. However, it requires several lines of code and a series of nested *if* and *for* blocks when written in simple languages. As a side note, we found that the restoration procedure requires the specification of the Euler integration method within Neuron’s NMODL files. Failing to do so, using instead the default Crank-Nicholson integrator, results in severely distorted results such as a much lower firing rate. Dangerfield’s reflection method takes similarly amount of lines of code as TR but it was more complicated to follow and implement.

**COMPUTATIONAL EFFICIENCY**

Our comparison of simulation time shows that DA-based methods are not the best choice for all the situations. It has already been noted that when the number of channels is low, MC modeling runs faster than DA (Orio and Soudry, 2012). The limit (number of channels) at which this happens is variable, depending not only on the time step used for numerical integration (as shown in Figure 8A vs. Figure 8B) but also on the kinetics of the channels being simulated, as this determines the number of transitions occurring in MCs (see below). Interestingly, the Stochastic Shielding in the context of MC (SSmc) behaves like the DA methods: it is faster than MC only with large numbers of channels and its performance depends mainly on the integration time step. This results from the increase in the number of (non-stochastic) ODEs that have to be solved at each integration step, regardless of the actual transitions that occur.

At first glance, it seems that a similar comparison carried out in another work (see Figure 8 in Huang et al., 2013b) produced very different results. In Huang et al. the MC method is faster than DA only if there are less than 20 ion channels. In contrast, we found

that MC is more efficient if there are less than  $\sim 500$  channels ( $dt = 5\mu\text{s}$ ) or  $\sim 5000$  channels ( $dt = 0.5\mu\text{s}$ ). The main reason for this large difference is that in the MC simulations of Huang et al. the state dynamics of each gating particle were updated individually. This method is highly inefficient and should be avoided in stochastic simulations of neurons. In contrast, we used the standard efficient MC method by Gillespie (1976), which tracks only the total number of channels in each state.

Interestingly, in another work (Rowat and Greenwood, 2014) it was found that Güler's CN method is much faster than Unbound DA. However, we have found that it is only a little faster (Figures 8A,B). The difference might be attributed to the language (Neuron vs. Python), or some other difference in the implementation.

Within all DA methods, Stochastic Shielding is the fastest in all circumstances. Next comes the colored noise approach (as noted, not applicable to all kinetic schemes) closely followed by Unbounded DA and Reflection. Recall that Stochastic Shielding (SSda) produces a minor loss in accuracy when used with a large number of channels, being in almost all cases indistinguishable from MC modeling. Finally, in our simulations the less efficient (slowest) algorithm was the Truncated and Restored DA (Huang's).

The numerical stability issue also deserves to be considered. Although we did not perform a systematical assessment of numerical stability in our simulations, we noted that simulations with the UA and SSda algorithms produced unreliable results with  $N_{Na} = 50$ , even at the lowest  $dt$  of  $0.5\mu\text{s}$ . As the other algorithms did not show this problem, it is most likely due to the lack of variable bounding. We did not pursue in finding a fix for these fails as the low channel number is a condition where MC modeling becomes the fastest and most accurate method.

### THEORETICAL ESTIMATES OF ACCURACY AND SPEED

For a given model, when is it better to use DA instead of MC? Specifically, we would like to know in advance when a DA approach will be accurate, and also faster than MC. A definite answer usually requires some preliminary simulations. However, as we explain next, a rough estimate could be obtained based on the following numbers: the simulation timestep, the number of channels to be simulated, and their typical constant rates.

Suppose we have  $N$  ion channels (of some particular type), with  $X$  of these ion channels in some state A, and  $\alpha$  being a kinetic rate from state A to another B. In each simulation timestep  $dt$ , let  $\Delta$  be the number of channels switching from state A to state B. As different channels are independent,  $\Delta$  is distributed according to a binomial distribution with  $n = X$ , and  $p = \alpha dt$ . Therefore, the average number of channels switching from A to B in that timestep is  $np = X\alpha dt$ .

These quantities can be used to estimate the expected accuracy of DA. As explained in Orio and Soudry (2012), the key idea in DA is to use the central limit theorem and approximate the distribution of  $\Delta$  to be Gaussian. This approximation becomes accurate when  $np = X\alpha dt \gg 1$ . This also means that  $N\alpha dt \gg 1$  since  $X < N$ . For example, in the HH model the slowest kinetic rates (in the relevant voltage range), are about  $\alpha \sim 0.1\text{ms}^{-1}$ . Then

if  $dt = 5\mu\text{s}$  DA should be expected to be accurate only when  $N > 2000$ , which is comparable to what we found in our simulations.

Using the same quantities we can also estimate the relative speed of the MC and DA algorithms. Simulation time is roughly proportional to the number of times the simulation variables (the fraction of channels in each) are being updated at each  $dt$  timestep (in which the voltage is updated). On the one hand, the MC algorithm performs updates each time a single channel switches between states. The number of these updates in each timestep, for each type of switch, is proportional to  $\Delta$ . Recall that the mean of  $\Delta$  is equal to  $np = X\alpha dt$ . Therefore, in total, about  $N\bar{\alpha}dt$  updates are performed on each timestep (where  $\bar{\alpha}$  is the appropriate average over all the kinetic rates). On the other hand, the DA algorithms perform a single update at each timestep. Therefore, DA should become more efficient than MC only when  $N\bar{\alpha}dt > 1$ . For example, in the HH model most state transitions occur near rest voltage in the (fast)  $m$  kinetics, and so we get approximately  $\bar{\alpha} \sim 2\text{ms}^{-1}$ . This condition yields results comparable to what we found in our simulations (Figure 8): if  $dt = 5\mu\text{s}$  then  $N > 100$ , and if  $dt = 0.5\mu\text{s}$  then  $N > 1000$ .

### CONCLUSIONS

Suppose  $dt$  is the simulation timestep,  $N$  is the number of ion channels, and  $\alpha$  is the "typical" transition rate of the channel. Our results suggest that, as a rule of a thumb,

- If  $N\alpha dt < 1$ , then MC simulation should be used—since it is both the fastest and most accurate method. Note that this is usually relevant to neuron models with less than 500 channels in a compartment—which is the common case in large multi-compartmental neuron models.
- If  $N\alpha dt > 1$ , DA should be used. In this case, one should use the method by Orio and Soudry (2012) which allows the simulated variables to remain unbounded (with an absolute value used to keep the stochastic terms real-valued). Additionally, the stochastic shield method by Schmandt and Galán (2012) method can be used with the DA equations to further speed up simulation, while remaining reasonably accurate.

### ACKNOWLEDGMENTS

This work was supported by grants Fondecyt 1130862, ACT-1104, and ACT-1113 to Patricio Orio (Consejo Nacional de Investigación Científica y Tecnológica CONICYT, Chile). The Centro Interdisciplinario de Neurociencia de Valparaíso is a Millennium Science Institute supported by funds P09-022-F, Ministerio de Economía, Chile.

### REFERENCES

- Bezanilla, F., Perozo, E., and Stefani, E. (1994). Gating of Shaker K<sup>+</sup> channels: II. The components of gating currents and a model of channel activation. *Biophys. J.* 66, 1011–1021. doi: 10.1016/S0006-3495(94)80882-3
- Bruce, I. C. (2007). Implementation issues in approximate methods for stochastic Hodgkin-Huxley models. *Ann. Biomed. Eng.* 35, 315–318. author reply: 319. doi: 10.1007/s10439-006-9174-9
- Bruce, I. C. (2009). Evaluation of stochastic differential equation approximation of ion channel gating models. *Ann. Biomed. Eng.* 37, 824–838. doi: 10.1007/s10439-009-9635-z
- Carnevale, N. T., and Hines, M. L. (2006). *The NEURON Book*. Cambridge: Cambridge University Press. doi: 10.1017/CBO9780511541612

- Chen, Y., and Ye, X. (2011). *Projection onto a Simplex*. Available online at: <http://arxiv.org/abs/1101.6081> [Accessed October 20, 2014].
- Colquhoun, D., and Hawkes, A. G. (1981). On the stochastic properties of single ion channels. *Proc. R. Soc. Lond. B Biol. Sci.* 211, 205–235. doi: 10.1098/rspb.1981.0003
- Dangerfield, C. E., Kay, D., and Burrage, K. (2012). Modeling ion channel dynamics through reflected stochastic differential equations. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.* 85:051907. doi: 10.1103/PhysRevE.85.051907
- Fox, R. F. (1997). Stochastic versions of the Hodgkin-Huxley equations. *Biophys. J.* 72, 2068–2074. doi: 10.1016/S0006-3495(97)78850-7
- Fox, R., and Lu, Y. (1994). Emergent collective behavior in large numbers of globally coupled independently stochastic ion channels. *Phys. Rev. E Stat. Phys. Plasmas Fluids Relat. Interdiscip. Topics* 49, 3421–3431. doi: 10.1103/PhysRevE.49.3421
- Gillespie, D. T. (1976). A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Comput. Phys.* 22, 403–434. doi: 10.1016/0021-9991(76)90041-3
- Gillespie, D. T. (2000). The chemical Langevin equation. *J. Chem. Phys.* 113, 297. doi: 10.1063/1.481811
- Gillespie, D. T. (2007). Stochastic simulation of chemical kinetics. *Annu. Rev. Phys. Chem.* 58, 35–55. doi: 10.1146/annurev.physchem.58.032806.104637
- Goldwyn, J. H., Imennov, N. S., Famulare, M., and Shea-Brown, E. (2011). Stochastic differential equation models for ion channel noise in Hodgkin-Huxley neurons. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.* 83:041908. doi: 10.1103/PhysRevE.83.041908
- Goldwyn, J. H., and Shea-Brown, E. (2011). The what and where of adding channel noise to the Hodgkin-Huxley equations. *PLoS Comput. Biol.* 7:e1002247. doi: 10.1371/journal.pcbi.1002247
- Güler, M. (2013). Stochastic Hodgkin-Huxley equations with colored noise terms in the conductances. *Neural Comput.* 25, 46–74. doi: 10.1162/NECO\_a\_00384
- Hines, M. L., and Carnevale, N. T. (2001). NEURON: a tool for neuroscientists. *Neuroscientist* 7, 123–135. doi: 10.1177/107385840100700207
- Hines, M. L., Davison, A. P., and Müller, E. (2009). NEURON and Python. *Front. Neuroinform.* 3:1. doi: 10.3389/neuro.11.001.2009
- Hodgkin, A. L., and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol.* 117, 500–544.
- Horn, R., Ding, S., and Gruber, H. J. (2000). Immobilizing the moving parts of voltage-gated ion channels. *J. Gen. Physiol.* 116, 461–476. doi: 10.1085/jgp.116.3.461
- Horrigan, F. T., and Aldrich, R. W. (2002). Coupling between voltage sensor activation, Ca<sup>2+</sup> binding and channel opening in large conductance (BK) potassium channels. *J. Gen. Physiol.* 120, 267–305. doi: 10.1085/jgp.20028605
- Huang, Y., Rüdiger, S., and Shuai, J. (2013a). Channel-based Langevin approach for the stochastic Hodgkin-Huxley neuron. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.* 87:012716. doi: 10.1103/PhysRevE.87.012716
- Huang, Y., Rüdiger, S., and Shuai, J. (2013b). Langevin approach for stochastic Hodgkin-Huxley dynamics with discretization of channel open fraction. *Phys. Lett. A* 377, 3223–3227. doi: 10.1016/j.physleta.2013.10.008
- Linaro, D., Storace, M., and Giugliano, M. (2011). Accurate and fast simulation of channel noise in conductance-based model neurons by diffusion approximation. *PLoS Comput. Biol.* 7:e1001102. doi: 10.1371/journal.pcbi.1001102
- Mélykúti, B., Burrage, K., and Zygalakis, K. C. (2010). Fast stochastic simulation of biochemical reaction systems by alternative formulations of the chemical Langevin equation. *J. Chem. Phys.* 132, 164109. doi: 10.1063/1.3380661
- Mino, H., Rubinstein, J. T., and White, J. A. (2002). Comparison of algorithms for the simulation of action potentials with stochastic sodium channels. *Ann. Biomed. Eng.* 30, 578–587. doi: 10.1114/1.1475343
- Neher, E., and Stevens, C. F. (1977). Conductance fluctuations and ionic pores in membranes. *Annu. Rev. Biophys. Bioeng.* 6, 345–381. doi: 10.1146/annurev.bb.06.060177.002021
- Orio, P., and Soudry, D. (2012). Simple, fast and accurate implementation of the diffusion approximation algorithm for stochastic ion channels with multiple states. *PLoS ONE* 7:e36670. doi: 10.1371/journal.pone.0036670
- Rowat, P. F., and Greenwood, P. E. (2014). The ISI distribution of the stochastic Hodgkin-Huxley neuron. *Front. Comput. Neurosci.* 8:111. doi: 10.3389/fncom.2014.00111
- Schmandt, N. T., and Galán, R. F. (2012). Stochastic-Shielding approximation of Markov chains and its application to efficiently simulate random ion-channel gating. *Phys. Rev. Lett.* 109:118101. doi: 10.1103/PhysRevLett.109.118101
- Schmidt-Hieber, C., and Bischofberger, J. (2010). Fast sodium channel gating supports localized and efficient axonal action potential initiation. *J. Neurosci.* 30, 10233–10242. doi: 10.1523/JNEUROSCI.6335-09.2010
- Schoppa, N. E., and Sigworth, F. J. (1998). Activation of Shaker potassium channels. III. An activation gating model for wild-type and V2 mutant channels. *J. Gen. Physiol.* 111, 313–342. doi: 10.1085/jgp.111.2.313
- Vandenberg, C. A., and Bezanilla, F. (1991). A sodium channel gating model based on single channel, macroscopic ionic, and gating currents in the squid giant axon. *Biophys. J.* 60, 1511–1533. doi: 10.1016/S0006-3495(91)82186-5

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

Received: 26 May 2014; accepted: 13 October 2014; published online: 03 November 2014.

Citation: Pezo D, Soudry D and Orio P (2014) Diffusion approximation-based simulation of stochastic ion channels: which method to use? *Front. Comput. Neurosci.* 8:139. doi: 10.3389/fncom.2014.00139

This article was submitted to the journal *Frontiers in Computational Neuroscience*. Copyright © 2014 Pezo, Soudry and Orio. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.