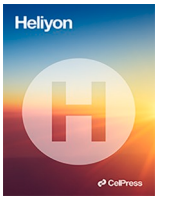
Heliyon 9 (2023) e16953

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon



A fast Markovian method for modeling channel noise in neurons

Norbert Ankri, Dominique Debanne *

UNIS, INSERM UMR_S1072, Aix-Marseille University, Marseille, France

ARTICLE INFO

Keywords:
Ion channel
Neuron
Action potential
Computer simulation
Noise

ABSTRACT

Channel noise results from rapid transitions of protein channels from closed to open state and is generally considered as the most dominant source of electrical noise causing membrane-potential fluctuations even in the absence of synaptic inputs. The simulation of a realistic channel noise remains a source of possible error. Although the Markovian method is considered as the golden standard for appropriate description of channel noise, its computation time increasing exponentially with the number of channels, it is poorly suitable to simulate realistic features. We describe here a novel algorithm at discrete time unit for simulating ion channel noise based on Markov chains (MC). Although this new algorithm refers to a Monte-Carlo process, it only needs few random numbers whatever the number of channels involved. Our fast MC (FMC) model does not exhibit the drawbacks due to approximations based on stochastic differential equations and the values of spike jitter are comparable to those obtained with the true Markovian method. In fact, we show here, that these drawbacks can be highlighted in the approximation based on stochastic differential equation methods even for a high number of channels (standard deviation of the 5th spike is about two-fold larger than that of MCF or true Markovian method for 5000 sodium channels). The FMC model appears therefore as the most accurate method to simulate channel noise with a fast execution time that does not depend on the channel number.

1. Introduction

Channel noise results from stochastic opening and closing of ion channels and is classically responsible for limiting the reliability of neuronal responses to repeated presentations of identical stimuli [1,2,3]. The fluctuations between open and closed states occurring randomly are driven by thermal noise [4]. Under simple assumptions, one can construct models that replicate the behavior of real channels. A crucial assumption of this class of models is that state transitions in ion channels are memoryless and behave like Markov process. The Markovian method at discrete time unit is considered as the gold-standard method for the simulation of ion-channel dynamics in the frame of Hodgkin-Huxley models [5,6,7,8]. The best description of stochastic gating of ion channels is attained with the use of continuous time and discrete states Markov Chain (MC) processes [9].

The Gillespie model developed for chemical reactions [10] presents the advantage to be extremely precise but the computing time required when the channel number increases becomes very large and other methods have been developed. These methods that correspond to the diffusion-based methods (or Langevin methods) are based on stochastic differential equations (SDE) and are designed for limiting the time-cost by simulating all the features of channel noise. In 1994, Fox and Lu [11] proposed two possible approaches: namely, the channel-based and the subunit-based methods. The first method is based on the precise identification of the

* Corresponding author.
E-mail addresses: norbert.ankri@univ-amu.fr (N. Ankri), dominique.debanne@univ-amu.fr (D. Debanne).

https://doi.org/10.1016/j.heliyon.2023.e16953
Received 23 July 2022; Received in revised form 17 April 2023; Accepted 1 June 2023
Available online 7 June 2023

state (1 to 8 for sodium) of each of the channels whereas the second method corresponds only to the state (open or closed) of the subunit.

Nevertheless, the most rigorous of these methods (i.e., the channel-based method) requires a larger dimension space of the channel states than subunit-based methods and the computation of diffusion matrixes becomes extremely time consuming. The other method proposed by Fox & Lu [11] (i.e. the subunit-based approach) was found to be not sufficiently precise to reproduce adequately the channel noise [12,13]. As demonstrated later [14], the approaches that considered the open or closed states of channel subunits were theoretically correct for calculating the mean but could not account for the noise variance found in Markovian chain (MC) models. New models that considered the fraction of channels in a given state (channel-based models) came up [14,15,16,17,18,19]. However, because they were based on the steady-state approximation (this is not the case for the OS model) for the calculation of the stochastic coefficients, these methods were found to be significantly imprecise in two tasks [14,15,16]. The Langevin methods cannot theoretically reproduce the features of channel noise due to Markovian process when the channel number is small.

In fact, it has been noticed that when a very small number of channels (i.e. <2000) is involved, the Gillespie algorithm accumulates all the advantages and is accurate and fast [20]. As stated by Orio & Soudry (2012), the best strategy seems to use the Gillespie method only for a limited number of channels that are estimated by formula $N.\alpha.dt < 1$ where dt is the simulation time step, N the number of channels and α, the typical transition rate of the channel [20]. For larger N, the authors propose another method [16]. Here, we show here that even with a high channel number the Orio & Soudry (OS) method is flawed. In a strict Markovian process, conditional statistical approximations can be used to accelerate the simulation [7]. Based on this, we set up a new simulation method at discrete time unit that we named *fast MC* (or FMC). FMC does not involve Cannon approximations, but the speed efficiency is firstly due to the low number of random variables needed since this number is the same whatever the number of channels and is only depending of the channel types, secondly to the use of fast algorithm according to a procedure described by Stadtlober (1989) to produce random binomial numbers needed for this algorithm [21]. We found that our model is significantly faster than Gillespie model and significantly more precise than all the methods based on stochastic differential equation (SDE), including the OS model.

2. Background

Markovian processes are stochastic processes characterized by the fact that prediction of the future is not changed by knowledge of

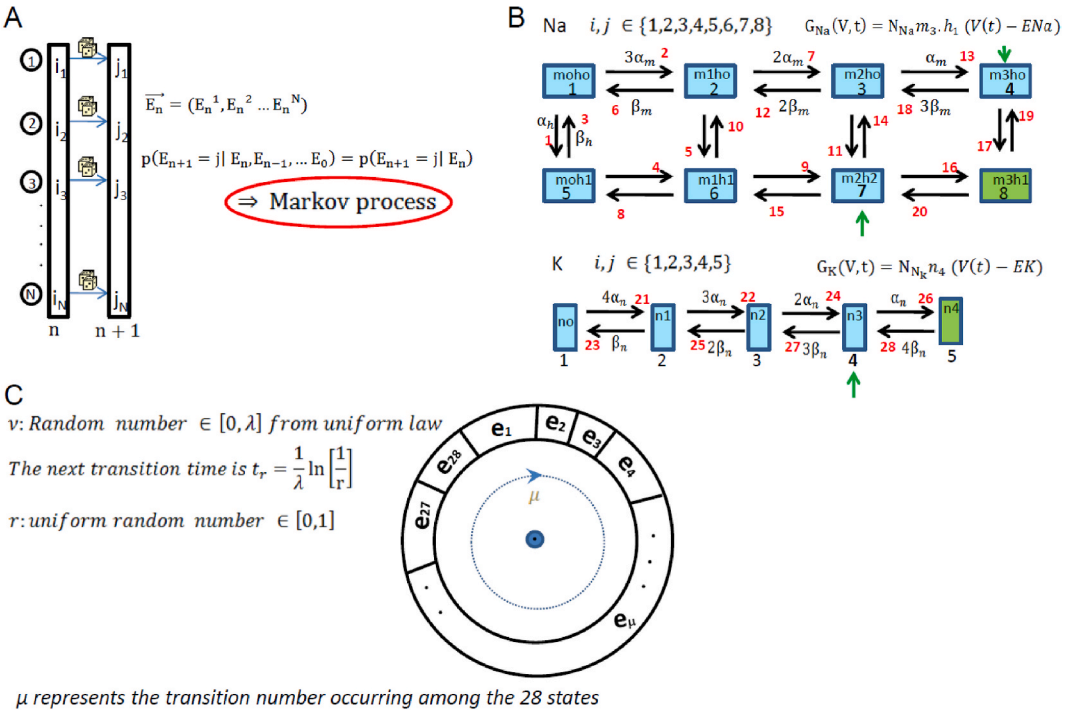


Fig. 1. Markovian Process for the simulation of ion-channel dynamics in the frame of Hodgkin-Huxley models. A. At time n Δt, the state of the N channels is given by the state vector: $\overrightarrow{E_n} = (E_n{}^1, E_n{}^2 \dots E_n{}^N)$. Markov processes are characterized by the fact that the probability of the state $\overrightarrow{E_{n+1}}$ depends only on the state $\overrightarrow{E_n}$ and that for each channel the probability $p(E_{n+1} = j)$ of occupying state j at time (n+1). **B**. We assume 8 possible states for the sodium channel (Na) and 5 possible states for the potassium channel (K). For each type of channel, there is only one open state (represented in green color). Transitions are arbitrarily numbered (red numerals). **C**. Gillespie-algorithm: the next transition is $tr = \frac{1}{N\lambda} \ln\left[\frac{1}{r}\right]$, where **r** is a uniformly distributed random variable between 0 and 1. The transition μ that occurs a time **tr** is randomly selected from a uniform distribution of numbers between 0 and λ. The probability of this transition is $\frac{e\mu}{\lambda}$ is proportional to the area of the corresponding wheel slot.

the past. Let us define *n* as the temporal index which corresponds to time step *n*. Δ*t*, and $\overrightarrow{E_n}$ the channel state at this time step defined by the sequence $E^1, E^2, \ldots E^i \ldots E^N$ in which E^i is the state of the *i*th channel and N the total number of channels. In the case of an ion channel, this means that the probability *p* of the future state $\overrightarrow{E_{n+1}}$ depends only on the present state $\overrightarrow{E_n}$. The passage of a channel from one state to another is called *transition*. Algorithms of the channel state tracking (CST) type take account of the state of every channel at time *n Δt* and their state at time *(n+1) Δt* is randomly determined according to a probability distribution reflecting the rates of all possible transitions In Channel State Tracking (CST) algorithm [12] (Fig. 1A), the state of each channel at each step unit is memorized and selected at random according to the probability law of transition. This way uses huge memory and provides very slow simulations. To overcome this disadvantage, in contrast to the CST-algorithm, the channel number tracking (CNT) algorithm determines only the number of channels in each state rather than the state of each channel, and thus saving both memory and computation time. The Gillespie algorithm which belongs to this last category memorizes only the channel number in a given state and only this number is actualized each time a novel transition occurs. The time of the next transition is selected at random according to the mean number of transitions, λ that occurred by step unit corresponding to the sum of all possible transition rates from a state to the next and for all *N* channels. For the sodium and the potassium channels there are 20 and 8 possible transitions respectively so a total of 28 possible transitions e_i, i = 1 to 28. (Fig. 1B and C).

By using the inversion sampling method, the time of the next transition can be obtained. Let *r* be a uniformly distributed random variable between 0 and 1. The transition μ occurring at time *tr* is randomly selected from a uniform distribution of numbers between 0 and λ. Starting from the number λ and using the inversion method, the moment of the next transition *tr* is given by equation (1):

$$tr = \frac{1}{N\lambda} \ln\left[\frac{1}{r}\right]. \tag{1}$$

Then it is necessary to decide which transition occurs among the 28 possible. Probabilities per time unit are e_i/λ (i = 1 to 28) and:

$$\sum_{i=1}^{28} e_i/\lambda = 1 \tag{2}$$

There are 20 transitions for sodium channel indicated in equation (3):

$$\begin{aligned} e_1 = m_0h_0\alpha_h, e_2 = m_0h_03\alpha_m, e_3 = m_0h_1\beta_h, e_4 = m_0h_13\alpha_m, e_5 = m_1h_0\alpha_h, e_6 = m_1h_0\beta_m, e_7 = m_1h_02\alpha_m, e_8 = m_1h_1\beta_m, e_9 = m_1h_12\alpha_m, e_{10} \\ = m_1h_1\beta_h, e_{11} = m_2h_0\alpha_h \;\; e_{12} = m_2h_02\beta_m, e_{13} = m_2h_0\alpha_m, e_{14} = m_2h_1\beta_h, e_{15} = m_2h_12\beta_m, e_{16} = m_2h_1\alpha_m, e_{17} = m_3h_0\alpha_h, e_{18} \\ = m_3h_03\beta_m, e_{19} = m_1h_1\beta_h, e_{20} = m_3h_13\beta_m \end{aligned} \tag{3}$$

And 8 transitions for potassium channel indicated in equation (4):

$$e_{21} = n_04\alpha_n, e_{22} = n_13\alpha_n, e_{23} = n_1\beta_n, e_{24} = n_22\alpha_n, e_{25} = n_22\beta_n, e_{26} = n_3\alpha_n, e_{27} = n_33\beta_n, e_{28} = n_44\beta_n \tag{4}$$

As every transition *i* comes true in proportion to the quantity e_i which is the product of a transition rate by the occupation density of the initial state, one has to randomly choose, according to the uniform law, a number ν between 0 and λ. The transition μ which then occurs is such as indicated in equation (5):

$$\sum_{i=1}^{\mu} e_i < \nu \leq \sum_{i=1}^{\mu+1} e_i \tag{5}$$

When the number of channels is big the time interval between two successive transitions becomes very small and subsequently the calculation time prohibitive. The tau-leaping method is an approximate of the Gillespie algorithm, performing all transitions for an interval of time tau before updating the probability functions. Reducing the frequency of updates allows more efficient simulations, enabling consideration of larger systems.

Instead of considering every single transition, we can decide to count those which occur during the time interval *Δt* and to update the resultant conductance with a temporal step of *Δt*. In this manner, the number of updates is much less that the number of transitions

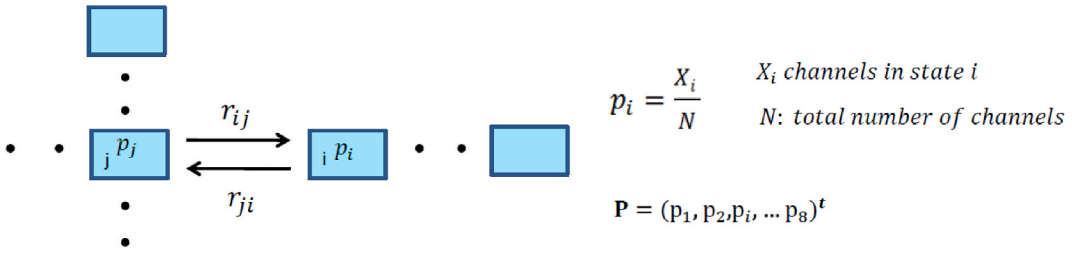


Fig. 2. Tau-leaping method. Tau-leaping is an approximate method for the Gillespie algorithm. At time *t* the proportion s of channels in each state (8 for the sodium channel) are the components of the state vector. These components vary according to the transition rates r_{ij}.

except at very low number of channels. Consider the general case where the transition rate between state i and state j is r_{ij} and let p_i be the proportion of channels in state i (Fig. 2). The derivative of p_i over time is given in equation (6).

$$\frac{dp_i}{dt} = \sum_j r_{ij} p_j - \sum_j r_{ji} p_i \tag{6}$$

We define the matrix M as indicated in equations (7) and (8):

$$M_{ij} = r_{ij} \ for \ i \neq j \tag{7}$$

$$M_{ii} = -\sum_j r_{ji} \tag{8}$$

If the exponent t designates the transposition, the state vector is expressed in equation (9):

$$\mathbf{P} = (p_1, p_2, p_i, \cdots p_8)^t. \tag{9}$$

Its derivative can be written as the master equation (10):

$$\begin{pmatrix} \frac{dp_1}{dt} \\ \vdots \\ \frac{dp_8}{dt} \end{pmatrix} = \begin{pmatrix} M_{1,1} & \cdots & M_{1,8} \\ \vdots & \ddots & \vdots \\ M_{8,1} & \cdots & M_{8,8} \end{pmatrix} \begin{pmatrix} p_1 \\ \vdots \\ p_8 \end{pmatrix} \tag{10}$$

We can integrate the master equation as indicated in equation (11):

$$P(t + \Delta t) = exp(Md\Delta)P(t) \tag{11}$$

Which connects the states vector to its derivative allowing to calculate the transition matrix T given in equation (12):

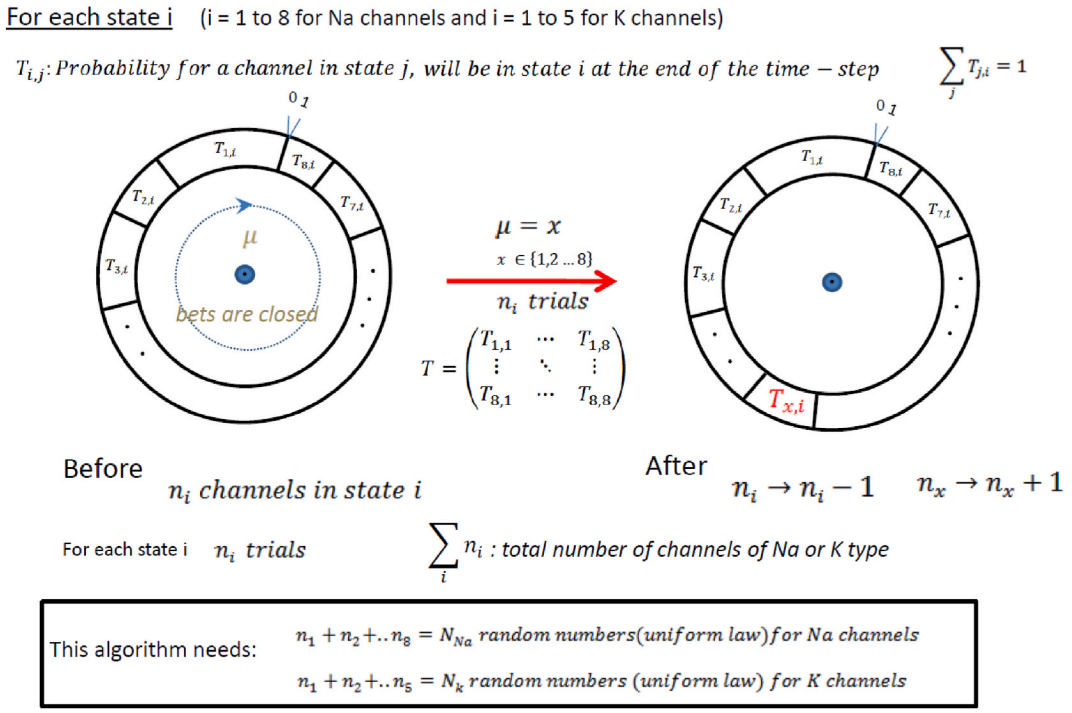


Fig. 3. Markov Chain (MC) algorithm. MC-algorithm – tau-leaping method: the coefficients T_{ij} of the transition matrix T represent the probability that a channel in state j will be in state i at the end of the interval $\triangle$t. From each initial state i the state j is chosen randomly (from $i = 1$ to 8 for the sodium channel and $i = 1$ to 5 for the potassium channel). After each draw the number of channels in the initial state i is diminished by 1 while the number of channels in the randomly selected final state x is increased by 1 (where x can naturally be equal to i). At every simulation step, this MC algorithm thus requires the generation of one random number for each channel considered, i.e. $N_K + N_{Na}$ random numbers of uniform distribution between 0 and 1 for a system of N_{Na} sodium and N_K potassium channels.

$$T = exp(M\Delta t) = \begin{pmatrix} T_{1,1} & \cdots & T_{1,8} \\ \vdots & \ddots & \vdots \\ T_{8,1} & \cdots & T_{8,8} \end{pmatrix} \quad (12)$$

The transition Matrix multiplied by the state vector at time t gives the deterministic state vector at $t+\Delta t$. For the stochastic part, one can use probabilities T_{ij} from the matrix and Monte-Carlo method to draw the arrival state at time $t+\Delta t$. If the interval Δt is sufficiently small to allow not more than a single transition, then the coefficients T_{ij} with indices i,j for which $M_{ij} = 0$ will equally be zero.

If at time t, n_i channels are in the state i, (for example $i = 1$ to 5 possible states for the potassium channel) at the time $t + \Delta t$ each of these n_i channels can be in state j with the probability T_{ji}, no matter what is j, including the same state i. One of the ways of proceeding as described in Fig. 3 is to do the following: from each initial state i the state j is chosen randomly according to the set of probabilities T_{ji}, (from $i = 1$ to 8 for the sodium channel and $i = 1$ to 5 for the potassium channel). After each draw the number of channels in the initial state i is diminished by 1 while the number of channels in the randomly selected final state x is increased by 1 (where x can naturally be equal to i). At every simulation step, this MC algorithm thus requires the generation of one random number for each channel considered, i.e. $N_K + N_{N_a}$ random numbers of uniform distribution between 0 and 1 for a system of N_{N_a} sodium and N_K potassium channels. Unlike those of matrix M, the coefficients T_{ij} of the transition matrix can be all not equal to zero anymore when Δt increases, they represent the probability for a channel to pass from the state j to the state i during the interval Δt. This algorithm modified from Gillespie allows a considerable reduction of the updating frequency of the state vector before integrating the HH equations. This algorithm requires random selection for all the channels in a given initial state j, the final state i among 8 states (Na ion) or 5 states (K ion) possible according to the probability T_{ij}. This operation therefore requires the generation of $N = N_K + N_{N_a}$ random numbers of uniform distribution at each Δt, which makes it very slow when N is large. From this method, Cannon et al. have used some criteria to avoid taking into account the unlikely transition.

3. Materials and methods

3.1. Models

A model of squid giant axon (SGA) was employed with parameters used in the simulations of Orio-Soudry [16] (see Table 1 & Table 2).

Kinetic parameter (in ms^{-1}) of m gates are provided in equations (13) and (14):

$$a_m = 0.1 * (v + 40)/(1 - \exp(-(v + 40)/10)); \quad (13)$$

$$b_m = 4 * \exp(-(v + 65)/18); \quad (14)$$

Kinetic parameters (in ms^{-1}) of h gates are provided in equations (15) and (16):

$$a\_(h) = 0.07 * \exp(-(v + 65)/20); \quad (15)$$

$$b_h = 1/(1 + \exp(-(v + 35)/10)); \quad (16)$$

Kinetic parameters (in ms^{-1}) of n gates are provided in equations (17) and (18):

$$a_n = 0.01 * (v + 55)/(1 - \exp(-(v + 55)/10)); \quad (17)$$

$$b_n = 0.125 * \exp(-(v + 65)/80); \quad (18)$$

If temperature is different from 6.3 °C, a correction is made on the coefficients with a factor Q10 = 3. For a temperature θ, the factor

Table 1
Model parameters used in NEURON.

MODEL parameters used in NEURON	squid giant axon (SGA)	EX
Membrane capacitance **Cm:** (μF/cm^2)	1	0.3
Leak conductance **G_leak:** (mS/cm^2)	0.3	0.11663167716
Leak potential **E_leak:** (mV)	−54.4	−60
Na Channels conductance **G_Na:** (mS/cm^2)	120	100
K channels **G_K:** (mS/cm^2)	36	50
Sodium reversal potential **E_Na:** (mV)	50	60
Potassium reversal potential **E_K:** (mV)	−77	−85
Temperature **θ** (°C)	6.3	25

Table 2
Model parameters used in LabView.

MODEL parameters used in LabVIEW (surface = 0.0001 cm^2)	squid giant axon (SGA)	EX
Membrane capacitance **Cm:** (pF)	100	30
Leak resistance **R_leak:** (MΩ)	0.033333	857.4
Leak potential **E_leak:** (mV)	−54.4	−60
Na Channels conductance **G_Na:** (nS)	120	10000
K channels **G_K:** (nS)	36	5000
Sodium reversal potential **E_Na:** (mV)	50	60
Potassium reversal potential **E_K:** (mV)	−77	−85
Temperature **θ** (°C)	6.3	25

will be $3^{(\theta-6.3)/10}$.

The SGA model was implemented with LabVIEW 2010, both for Gillespie, HMC and Orio-Soudry (OS) and compared with the result by Pezo et al. in a NEURON environment: Comparison of DA-based Stochastic Algorithms [20]. Gillespie, DA with truncation and restoration (hhTR), Stochastic Shielding (hhssDA) and DA with reflection (HHRef), sample codes and.mod files can be found in ModelDB http://senselab.med.yale.edu/ModelDB/Accession 167772.

In order to test whether the simulation of channel-noise according to OS functioned in other types of neurons, we used a second model (EX) with parameters in Table 1 & Table 2.

Kinetic parameters of *m* gates are provided in equation (19) and 20:

$$a_m = 0.1 * (v+40) / ((1 - exp(-(v+40)/10)));\tag{19}$$

$$b_m = 4 * \exp(-(v+65)/18);\tag{20}$$

Kinetic parameters of *h* gates are provided in equations (21) and (22):

$$a_h = 0.07 * \exp(-(v+65)/20);\tag{21}$$

$$b_h = 1/(1+\exp(-(v+35)/10));\tag{22}$$

Kinetic parameters of *n* gates are provided in equation (23) and 24:

$$a_n = 0.01 * (v+34) / (1 - \exp(-(v+34)/10));\tag{23}$$

$$b_n = 0.125 * \exp(-(v+44)/80);\tag{24}$$

Note that, in the EX-model, the only changes from the HH model is in the a_n rate for the potassium ion which makes it more excitable (a train of action potential is produced without current input), the capacity that is threefold smaller, reversal potentials for Na, K and Leak, and the temperature (25 °C instead 6.3 °C).

3.2. Generator of binomial noise with the algorithm BIN

Algorithm BIN ($n \geq 1$, $p \leq 0.5$)

1. (Set-up) Constant B = 7 (for 9 decimal digits' precision).

Pre-set $q = 1-p, r = 1-\frac{p}{q}, \mu = np, t = (n+1)r, f0 = q^n, b = \min(n, \lfloor \mu + B\sqrt[2]{\mu q} \rfloor)$..

2. Generate U from U(0,1). Set $K = 0, f = f0$.
3. If $U \leq f$ return to *K*.
4. Set $K = K+1$. *If* $K > b$ *goto* 1..

Otherwise set $U = U - f, f = f\left(\frac{t}{K} - r\right)$ and goto 2.

Execution time is proportional to *np*.

This choice of the *BIN* procedure was mainly dictated by the efficiency and the simplicity of this procedure. It should be noted that *BIN* is comparable to what is used in *Python* to write a multinomial function.

For *np* greater than 10, this product being calculated at each time step, the binomial law was approximated by the normal law *N*(n, sqrt(npq))* with variance *np(1-p)*.

4. Results

4.1. Fast Markov Chain (FMC) algorithm

Rather than randomly choosing for each of all channels being in the same initial state *i*, it is possible to reduce the number of random values by considering in one shot, the number of all channels arriving in a certain state (Fig. 4). As usual, we call $B(N, \mathrm{p})$ the binomial law where *n* is here the number of trials and *p* the probability of success. For each state initially containing N_i channels, 1) a number randomly selected from a binomial distribution $B(N_i, p_1 = T_{1i})$ determines the number of channels arriving at state 1, the first of the possible destinations. 2) Following the first draw, N_i is diminished by the number n_1 of channels who have left state i to enter state 1, whose occupancy increases by the same amount $n_1 = \Delta_{1i}$. A new draw $B[(N_i - n_1), p_2]$ is performed with a probability of transition towards state 2 equal to $p_2 = \frac{T_{2i}}{1-T_{1i}}$, which is a conditional probability taking account of the remaining states. We continue the operation until the last of these destinations (5 in the case of the potassium current, as illustrated here) is reached. For the destination number *k,* we have equation (25):

$$B\left[\left(N_i - \sum_{u=1}^{u=k-1} n_k\right), p_k\right] \tag{25}$$

and equation 26

$$p_k = \frac{T_{ki}}{1 - \sum_{u=1}^{u=k-1} T_{ui}} \tag{26}$$

that gives the probability of transition from state *k* to state *j*, based on the outcomes of previous draws.

The last destination, to which a probability of 1 is assigned, collects all the remaining channels. Calls to the random number generator are reduced to the number of states minus one (e.g. 4 in the case of potassium channels) vs. the number of channels in the classical MC algorithm. However, the distribution to be generated is not uniform but binomial, which requires more computation time. The benefit of the present method will, therefore, strongly depend on the availability of an efficient and rapid subroutine to generate random numbers from a binomial distribution B(n,p), for which we chose the BIN algorithm [21].

4.2. Multinomial law

The procedure described above is equivalent to random drawing according to a multinomial law. Indeed, if n_i is the number of channels in state *i* at the time *t*, the new repartition at the next time step of these n_i channels is n'1, n'2 … …n'8 with for sodium

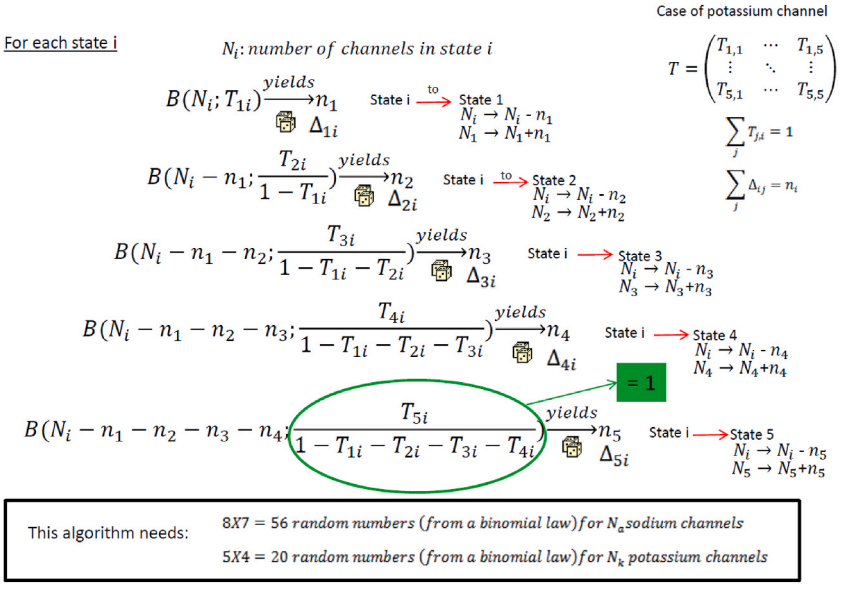


Fig. 4. Fast Markov Chain (FMC) method. FMC algorithm applied to the *Ni* channels in state *i* at time *t*. This algorithm is applied for the other possible states of the considered ion channel (here potassium channel).

channels equation (27) gives:

$$\sum_{j=1}^{8} n_j^{'} = n_i(t) \tag{27}$$

We draw at random, according to probability T_{ji}, for each of these channels in state i, its destination state j among the 8 possible states. The product ${T_{1i}}^{n_1^{'}}$ x ${T_{2i}}^{n_2^{'}} \ldots {T_{8i}}^{n_8^{'}}$ gives the probability that the first *n'1* trial among the n_i total trials give the state 1, then the n'2 following trials give the state 2 and so on in this precise order. There are $n_i!$ possible permutations that give the same final vector (n'1, n'2 ….n'5) and among these $n_i!$ permutations, n'1!n'2!n'3!n'4!n'5! are indiscernible. Finally, we see that the probability law to simulate is a multinomial law (see methods) given in equation (28):

$$\mathrm{Mul}\left(n_{1,}^{'} n_2^{'} \ldots n_8^{'};\ T_{1i}, T_{2i} \ldots T_{8i}\right) = \frac{n_i!}{n_1^{'}! \ldots n_8^{'}!} {T_{1i}}^{n_1^{'}} . {T_{2i}}^{n_2^{'}} \ldots . {T_{8i}}^{n_8^{'}} \tag{28}$$

Since the number of possible states is generally low compared to the number of channels, the number of draws is considerably reduced. But in contrast with previous methods, draws have to follow binomial law. Therefore, this process would take advantage of using a fast binomial noise generator. For this reason, to produce these binomial random numbers according to the law $B(N,p)$, we used the inversion method of Ernst Stadlober. This algorithm called BIN (for Binomial) is used each time the product *np* was less than 10 and is very simple, its execution time is proportional to the product *np*.

At each time step, the product *np* is calculated with equations (29) and (30).

$$n = N_i - \sum_{k=1}^{k=i-1} n_k \tag{29}$$

$$p = \frac{T_{k,i}}{1 - \sum_{u=1}^{u=k-1} T_{u,i}} \tag{30}$$

When n is greater than 10, the binomial law was approximated by the normal law *N*(n, sqrt(npq))* with variance *np(1-p)* and the

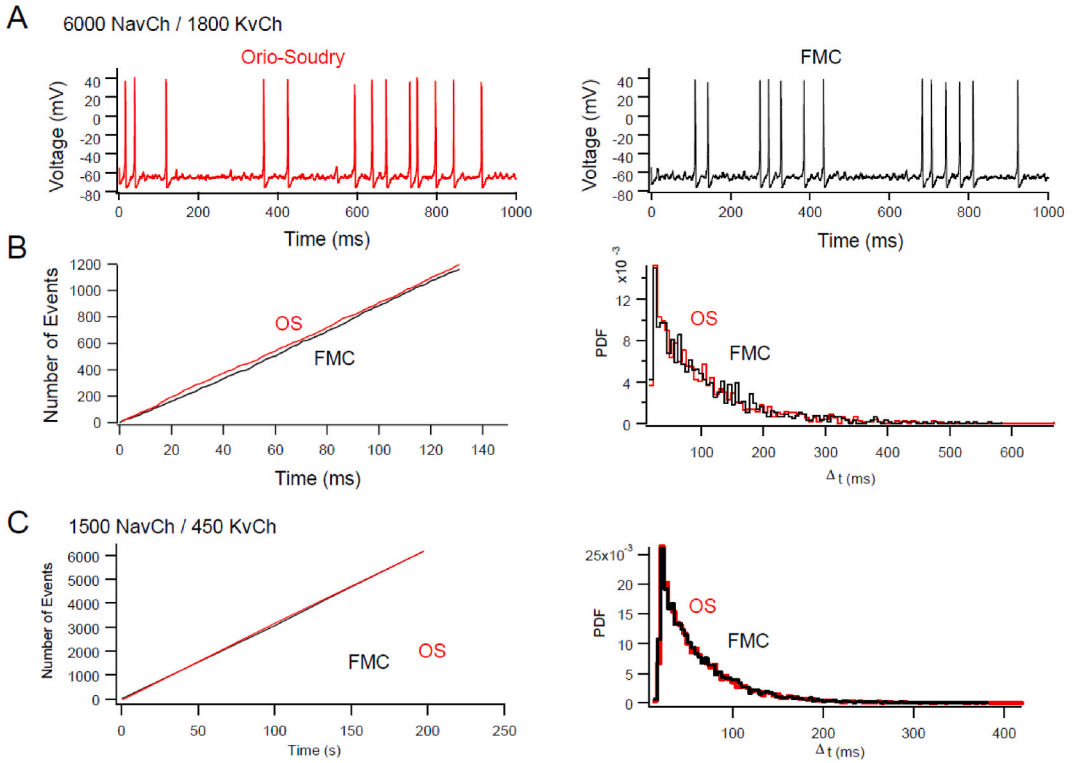


Fig. 5. Equivalence of the OS and FMC models in the squid axon. Comparison of results for the example of the squid giant axon simulated by Orio & Soudry (2012) using their model (in red) and the FMC algorithm (in black). Simulation results for 6000 Na- and 1800 K-channels (**A**, **B**) or 1500 Na- and 450 K-channels (**C**). All parameters are identical to Orio & Soudry (2012). A. Representative voltage traces (1 s). **B**/**C**, left: Evolution of AP number as a function of simulation time in ms. **B**/**C**, right: Histograms of Inter-AP intervals.

Gaussian number is rounded to the nearest in the order to keep the discretization of the noise. But it is known that there is an additional noise due to the roundness, the absolute value of this noise is given by the following integral (equation (31)):

$$\int_{-1/2}^{1/2} \varepsilon^2 f(\varepsilon) d\varepsilon \approx \frac{1}{12} \tag{31}$$

When np is greater than 10, we can consider that the density probability function $f(\varepsilon) \approx 1$ and is uniform on the interval. So, we have to subtract the value $\frac{1}{12}$ from the theoretical calculate variance. This correction was invisible in all cases of this study, except a slight but significant effect in the last example considered.

Compared to the Gillespie algorithm, in the case of potassium channels, the number of random values (*RV*) is reduced from N_K the number of potassium channels, to 20 RV and for Na, the number of sodium channels, to 56 RV for sodium channels. Indeed, in the case of a number of states N_s, only the quantity $N_s*(N_s-1)$ random numbers is to be produced. When there is no more than a single state to be filled, all the remaining channels are assigned to this value and the probability of the last binomial drawing is equal to 1 (Fig. 4).

In the LabView environment, implementation of the OS method allowed to obtain the same results as in the NEURON environment. No significant statistical difference between the FMC and the Orio-Soudry algorithm (OS) can be observed regarding the evolution of the number of action potentials (APs) in time as the inter-spike intervals (Fig. 5). Indeed, for the model of squid axon, the spike number was identical for OS and FMC algorithms for 6000 Na and 1800 K channels (Fig. 5A and B) and for 1500 Na and 450 K channels (Fig. 5C).

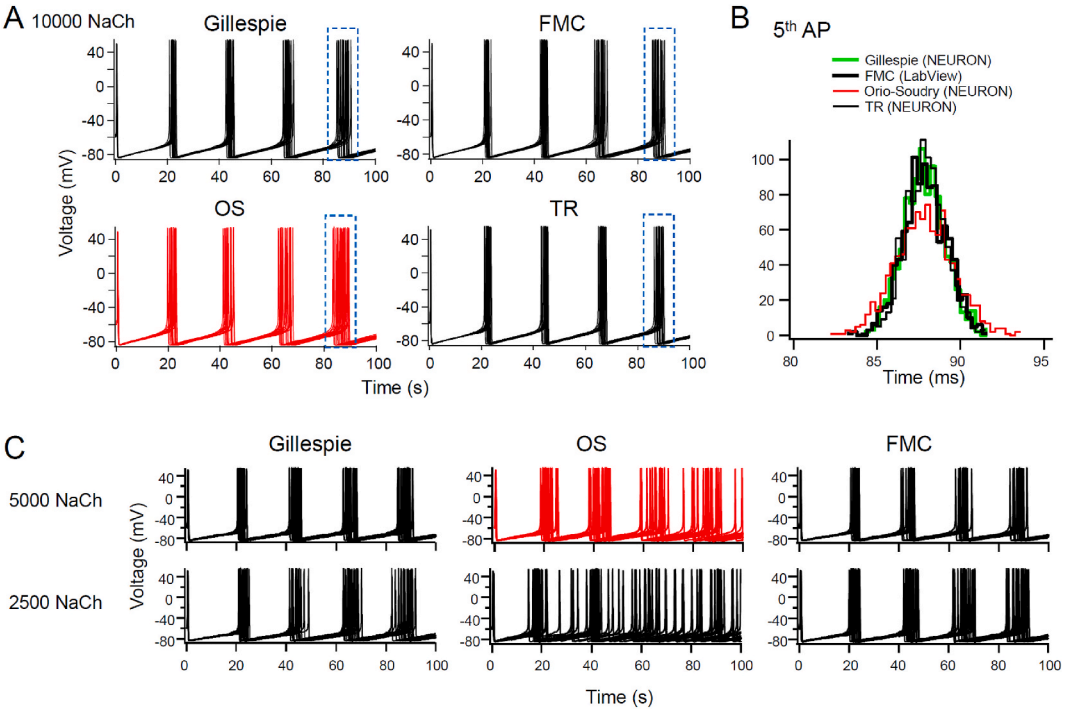


Fig. 6. Discrepancy in spike-timing between OS and Gillespie/FMC models. Even with a large number of channels, one cannot be sure that SDE-based methods produce noise with the same effect as that of noise produced by the standard method. An example is shown here (this particular example, where the neuronal parameters yield diverging results between EDS and MC methods, will be designated example EX). $\Delta t = 10$ μs for all simulations. **A.** Channel numbers are: $N_{Na} = 10000$, $N_K = 5000$ (see other parameters). Upper-right, original Gillespie simulation, upper-left, FMC simulation, Lower-left, Orio-Soudry, Lower-right, TR Truncated-Restored (Huang et al., 2013). **B.** Superimposed histograms of jitter of 5th AP (n = 1000 runs) for Orio-Soudry (red) Gillespie (dotted line), FMC (black thick line) and Truncated-Restored method (TR) (black line). While as expected, means of all distributions are equivalent (87.68 for Gillespie, 87.75 for HMC, 87.69 for Orio-Soudry, 87.75 for Truncation-Restoration). Jitter distributions are identical (not significantly different) for Gillespie, FMC and TR respectively SD = 1.25 and SD = 1.29, SD = 1.20 (F-Test: P = 0.94) but (significantly) larger for Orio-Soudry SD = 1.77 (F-Test: $P = 7 \times 10^{-12}$). It should be noted that, with respect to the standard model, the model of Orio-Soudry engenders a larger variance in AP-jitter, irrespective of the time step used ($\Delta t = 10$ μs or $\Delta t = 1$ μs, not showed). **C.** Comparison between the FMC (right) Gillespie (left) algorithms and that of Orio-Soudry (middle) for gradually diminishing channel numbers (5000, 2500 Na-channels; K-channel numbers were (Na-channel)/2 numbers throughout; note increase in noise). The model of Orio-Soudry, but not the FMC model, diverges from that of Gillespie.

4.3. Divergence of SDE from Markovian algorithm

Even with a large number of channels, one cannot be sure that SDE-based and standard (Markovian) methods would produce noise with the very same effect on spike jitter. In order to evaluate the robustness of the SDE-based method, we used another model of neuron (Model Ex) that spontaneously fire action potentials and that contained 10000 Na channels (see Methods). We should note that in the deterministic case (no channel noise) the EX-model is very stable showing always for the parameters used, 5 action potentials separated by the same time interval and this for a temporal resolution which extends over a range of 10 to 100 μs. We analyzed the variance of the fifth action potential over 1000 runs (Fig. 6A). Four methods of channel noise simulation were used: Gillespie (Markovian), OS, FMC and TR (Truncation-Restoration) [18]. The Gillespie, FMC and TR methods provided similar results but not OS that clearly diverged (Fig. 6B). While as expected, means of all distributions are equivalent (87.68 for Gillespie, 87.75 for FMC, 87.69 for OS, and 87.75 for TR), jitter distributions are identical (not significantly different) for Gillespie FMC and TR (respectively SD = 1.25 and SD = 1.29, SD = 1.20; F-Test: P = 0.94) but significantly larger for OS (SD = 1.77; F-Test: $P = 7 \times 10^{-12}$). If we increase the temporal resolution, these results remain the same.

In a second step, the number of Na channel was reduced to 5000, 2500 Na channel to probe the robustness of FMC model. As illustrated in Fig. 6C, Gillespie and FMC provided similar behavior but OS clearly diverged (mean timing of the 5th action potential,

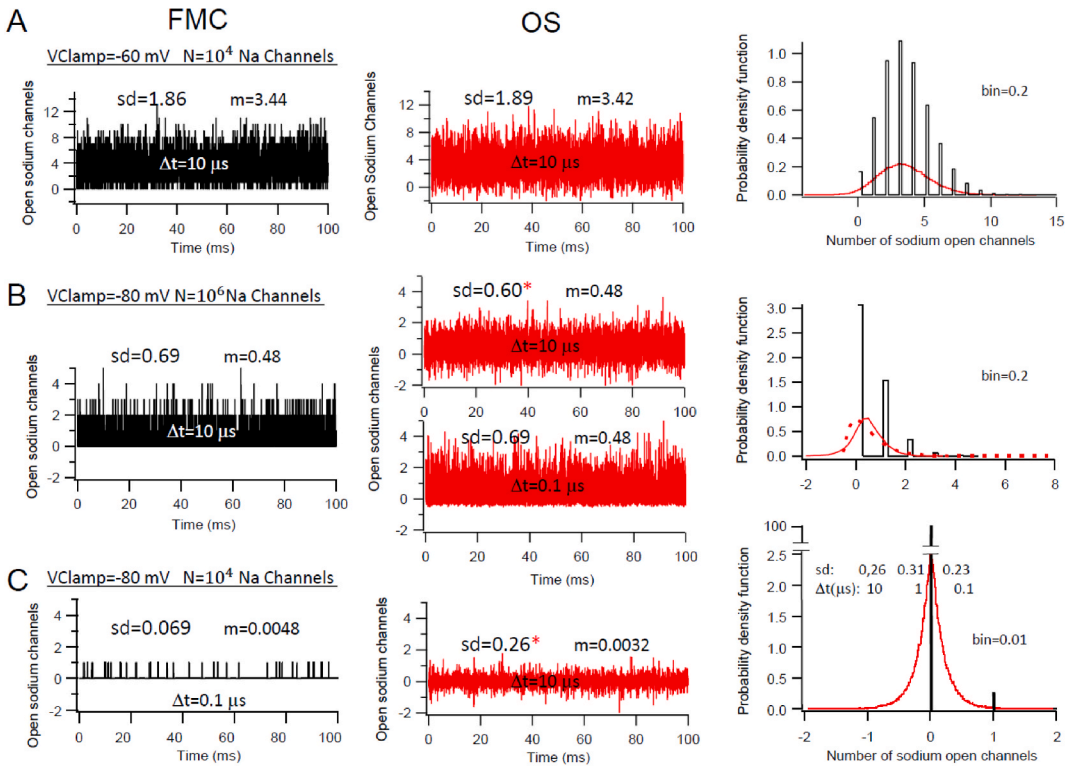


Fig. 7. Voltage clamp in FMC and OS models. The samples of noise in voltage clamp represent in all cases the number of sodium channels open as a function of time. The noise histograms of the different cases are on the right column, those corresponding to the Gillespie method being indistinguishable from those of the FMC algorithm, have not been represented in the figure. Bin used are in number of channels. The time of computation for the time resolution $\Delta t = 0.01$ ms for the 100 ms of signal is less than 0.5 and 1 s for respectively OS and FMC. But this time for the Gillespie method is about 10 s when $N = 1000$ and reach almost 900 s when $N = 1{,}000{,}000$. The statistics (mean and sd) were calculated with 1 s of signal. **A.** With $N = 10^4$ sodium channels, the voltage clamped at $V_c = -60$ mV, and for a simulation performed at a time step $\Delta t = 10$ μs, we see that the mean (m) and the standard deviation (sd) of the noise generated by the OS algorithm (red color in the center) are identical to those produced by the FMC or Gillespie method (black color on the left). Note that the noise has a negative part in the OS case. This is normal since OS does not use boundaries constraints. On the right, the histogram of the number of open sodium channels clearly shows the continuous nature of the Gaussian noise OS compared to the binomial models (Gillespie or FMC). **B.** The number of sodium channels is $N = 10^6$ and the holding potential is −80 mV. When the temporal sampling step is 10 μs, the standard deviation of the OS noise marked by an asterisk sd = 0.60 is less than those produced in the case of FMC or Gillespie sd = 0.69. Note that OS noise, unlike binomial noise, has a symmetrical appearance confirmed by its histogram on the right (solid red line) which, however, departs from Gaussian law. For a sampling frequency of 10 μs, the OS noise finds a correct standard deviation value and the noise has an asymmetry comparable to that of binomial noise, asymmetry of the OS noise can be seen on its histogram (red dotted line color). **C.** At VClamp = −80 mV but for $N = 10^4$, there is a divergence between the OS algorithm and binomial algorithms (Gillespie, FMC) materialized by the great difference between the standard deviations (sd = 0.07 for FMC or Gillespie and sd = 0.26 for OS). The symmetric histogram corresponding to the OS method is no longer Gaussian at all and its standard deviation difference from Gillespie varies with the sampling frequency: sd = 0.26, sd = 0.31 and sd = 0.23 respectively for $\Delta t = 10$ μs, $\Delta t = 1$ μs and $\Delta t = 0.1$ μs.

OS: 89.20 ms, FMC: 90.64 ms & Gillespie: 90.51 ms; SD of the 5th action potential timing, OS: 3.49, FMC: 1.93 & Gillespie: 2.022, n = 500 trials).

For certain values of parameter, we can notice a significant difference in the results obtained with the Markovian type algorithms as Gillespie and FMC and those of the SDE methods. In the simulation of Fig. S1 the number of sodium and potassium channels is respectively 10000 and 5000. By superimposing 19 trials, we can notice that almost all the methods present a spreading of their AP centered around those of the determinist one which possesses 5 APs (Figs. S1A and B), except both Stochastic shielding (hhss) and Reflection (hhref) methods (Fig. S2) for who single sweeps are represented and which are ipso-facto disqualified by showing as already noticed by OS, an excitability that is clearly larger than the reference model. OS exaggerates the jitter variance of the AP. This larger variance can be easily seen on the histogram and is completely confirmed by the analysis of variance realized by the Fisher test of equality of two variances (F-test).

4.4. Channel noise in voltage-clamp

Next, we tried to understand the reasons for the divergence between OS and FMC. With the same parameters of neuron (evoked spikes of EX simulation) we then compared the variation in sodium channel noise obtained in voltage-clamp at a holding −60 mV with OS and FMC methods. The sodium noise was found to display similar means and standard deviations using OS or FMC methods for both squid axon model (**not shown)**) and for our second neuron model (Fig. 7A). With a low number of channels, the value discretization of the channels can be seen in current traces. This discretization present in the Markovian method is absent in the OS method that uses a continuous noise instead discrete one (Fig. 7A). As notified by Huang et al. (2015), the discretization procedure is feasible only when the boundary condition is satisfied that is not the case for the OS algorithm [19]. In addition, when the values of the noise are rounded to the nearest integer, although the jitter of the 5th peak for the EX-model is improved as we can see in the traces in red color (Figure suppl. 1A), the algorithm OS give a wrong number of spontaneous PA (Figure suppl. 1B). Indeed, there is an additional noise due to the roundness operation. So, one cannot use the round function with OS algorithm because it wrongly increases excitability.

At low voltage (Fig. 7B and C) the OS method produces a wrong standard deviation of channel noise compared to that produced by the FMC one which is always conform to Gillespie model. With OS algorithm the number of channels in some states sometime reaches negative value, and the calculation of standard deviations causes wrong values of these latest. At a sample period of $\Delta t = 10$ μs, depending on the number of channels, the standard deviation can be higher than the true one (Fig. 7C, $N = 10{,}000$) or lower (value marked by an asterisk Fig. 7B, $N = 1000{,}000$). In this case, the use of a high temporal precision ($\Delta t = 0.1$ μs) corrects the underestimation present at the lower sampling frequencies (Fig. 7B).

To obtain the correct result, the OS algorithm must use a sampling frequency 100 times faster. This has the major consequence to make it slower in practice than the FMC algorithm. When the number of channels is only 10,000 as in the simulation illustrated in Fig. 6A, even at $\Delta t = 0.1$ μs, the variance remains with the OS algorithm much above the value given by the binomial algorithm (0.26 versus 0.07) (Fig. 7C).

Note that although using a Gaussian generator, the noise produced by the OS algorithm may not be Gaussian. At low potentials, the

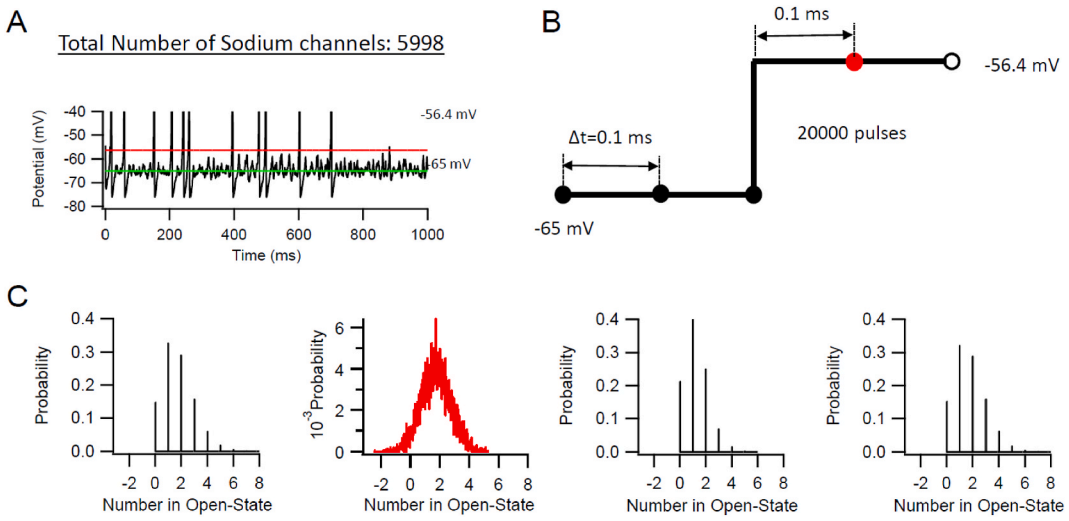


Fig. 8. Model dynamics during voltage transition. A. Current clamp simulation of a duration 1 s in current clamp for the example of the Squid Giant Axon with 6000 Na-Channels and 1800 K-channels. The noise oscillates around an average level of −65 mV and the maximal depolarization in the absence of PA does not exceed a potential of −56.4 mV (i.e. the maximal empirical value observed). **B.** Depolarizing pulse from −65 to −56.4 mV in Voltage Clamp used for the four algorithms: Gillespie, OS, OS-Bino and FMC. Points represent the instants when the distribution of channels in the various states is calculated using the four programs; the temporal step is here 100 μs. **C.** Histograms of open-state number of channels for the four considered algorithms. The used bin is 0.01. We notice that in the case of the OS algorithm, the use of a not discrete variable as well the permission of the negative values. Note that Gillespie and FMC Histograms are indistinguishable.

Table 3
Comparison of the data obtained with different models at the equilibrium (−65 mV). DETERMINIST case: SD = 0 and <N ≥ 0.533. Channels noise: n = 20,000 trials, **s** is the square root of the estimated variance.

Algorithm	Gillespie	OS	OS-Bino	FMC
<N>	0.533	0.528	0.543	0.533
$s/\sqrt{n}$	±0.005	±0.005	±0.005	±0.005
SD	0.728	0.730	0.734	0.759
$s^2/\sqrt{2/(n-1)}$	±0.004	±0.004	±0.004	±0.004

Table 4
Comparison of the data obtained with different models 0.1 ms after jump to −56.4 mV. DETERMINIST case: SD = 0 and <N ≥ 1.721. Channels noise: n = 20,000 trials, s is the square root of the estimated variance. **, F-test p < 0.00001.

Algorithm	Gillespie	OS	OS-Bino	FMC
<N>	1.727	1.724	**1.214** **	1.7222
$s/\sqrt{n}$	±0.007	±0.007	±0.007	±0.007
SD	1.217	**0.994** **	**0.909** **	1.215
$s^2/\sqrt{2/(n-1)}$	±0.015	±0.01	±0.01	±0.015

noise can have negative values, and to remedy this problem, the algorithm uses an absolute value to calculate the standard deviation of the noise. When the absolute value acts on negative values, the noise is no longer Gaussian.

Fig. 8 examines the Orio-Soudry model for the squid giant axon with 5998 sodium channels and 1800 potassium channels. We first look with a temporal step of 10 μs, and then 100 μs, at the distribution of channels in 8 possible states of the sodium, in voltage clamp at the equilibrium at −65 mV or during one depolarizing pulse (Fig. 8B). The values of the depolarizing pulse were chosen so that the resting membrane potential corresponds to the average level of the noise in current clamp (Fig. 8A) and the highest potential to the maximal noise level which does not activate an action potential.

These voltage steps were realized by starting from an initial distribution of the number of channels sodium: *N1, N2 ⋯ N8* corresponding at the equilibrium to a voltage of −65 mV and by rounding each of these values to the closest integer so as to guarantee a same initialization for all the four following algorithms: Gillespie, HMC, OS and OS-bino (see below), because that of OS treats continuous values with Gaussian approximation contrary to three other algorithms which use only integers of the binomial law. So, the initial distribution was exactly:

$N1 = 2058, N2 = 345, N3 = 19, N4 = 0, N5 = 3038, N6 = 509, N7 = 28, N8 = 1$ for a total number of sodium channels equal to 5998 instead 6000.

In the case of the equilibrium at −65 mV, the average number of channels in the open state of the sodium channel (state 8) was estimated from a simulation of 20,000 trials corresponding to a total duration of 20,000 X $\Delta t = 2$ s for $\Delta t = 10$ μs or 20 s in the case $\Delta t =$ 100 μs. For each step Δt, thus we have an initial distribution which is the result of the previous one and is not consequently strictly equal to the theoretical initial one. On the other hand, in the case of depolarizing pulse, 20,000 trials are realized by starting every time from the initial theoretical distribution and by looking at the new distribution at the end of the time step Δt.

To objectively estimate the possible differences between the algorithms we calculate the standard error on the average and the standard deviation estimated respectively by: $\frac{s}{\sqrt{n}}$ et $s^2\sqrt{2/n-1}$ where S is the square root of the variance estimated on the size sample of 20,000.

When $\Delta t = 10$ μs we see no difference among four tested algorithms (data not shown). On the other hand, when $\Delta t = 100$ μs four algorithms continue to agree concerning the number of open state channels at an equilibrium potential of −65 mV (Tables 3 and 4), there is this time differences for OS and OS-Bino with regard to the reference Gillespie algorithm in the case of depolarizing pulse. The algorithm FMC is equivalent in all cases to that of Gillespie (Fig. 8C). By focusing on the open state of sodium channels we notice from the reference algorithm, that during depolarizing pulses, when that the time step is 10 μs, then, less than 1% of these open states results from transitions not directly connected. On the other hand, for $\Delta t =$ 100 μs, this proportion increase to 37% with the following repartition: 78% for state 1, 21% for state 2, 0.2% for state 3, 0.2% for state 4 and less than 1% for state 6.

4.5. Variant of the OS algorithm: OS-Bino

As suggested by Orio and Soudry (2012), there is a connection between the DA approach and another simulation method belonging to the family of "binomial population" [16]. Each channel transition Δ_{ij} from state j to state j is binomially distributed. So, instead to approximate Δ_{ij} by a Gaussian random variable (RV) as OS do in order to increase simulation speed, one could directly generate this RV by using the fast binomial generator of Stradlober [21]. We have therefore implemented this algorithm and called it OS-Bino.

For larger values of the sampling step Δt, the binomial method OS-Bino led to a bias in both the variance and the mean value of the

noise (Fig. 8, Table 3 & Table 4). Indeed, when Δt is not small enough, transitions not directly linked in the underlying kinetic model occur during this interval, in particular towards the open state. In the case of OS algorithm only the variance is affected by this bias since the mean is identical to that obtained from the deterministic calculation.

This is illustrated in Table 4 where we can see that OS-bino leads to the underestimate of both the average number of channels in the open state (1.21 instead of 1.72) and the standard deviation of the noise (SD = 0.90 instead of 1.21; F-test $p < 0.0001$). In contrast, the OS algorithm gives a true value of the mean because the symmetric Gaussian noise is added to the theoretical average value (value DET <N > Fig. 9) which comes directly from the integration of the determinist part of the stochastic differential equation. On the other hand, even for OS, the calculation of the variance of the Gaussian noise being calculated without taking into account not connected states, is underestimated too (SD = 0.99 instead of 1.21) and has to grow up with Δt. Indeed, when we start again the simulation of Fig. 6 with $\Delta t = 100$ μs, with $N = 10{,}000$ or $N = 1{,}000{,}000$ being the number of sodium channels, we found (data not shown) that the "jitter" of the fifth peak for the OS algorithm is much smaller than binomial algorithms in agreement with the fact that the variance of the noise in "voltage clamp" at −80 mV smaller too.

4.6. High number of channels: $N = 10^7$

We show in this voltage-clamp simulation of Fig. 9 that a statistical difference can be highlighted even when the number of channels involved is large and that the temporal resolution used plays an important role. Again, with the parameters of the evoked spikes (EX model), Fig. 9A shows a voltage clamp simulation with a holding potential VC = −80 mV and a very high number of sodium channels ($N = 10^7$). This simulation was done both for the FMC and OS algorithm with 3 different time resolutions: $\Delta t = 10$ μs, $\Delta t = 1$ μs, $\Delta t = 0.1$ μs? The simulation was not done with the Gillespie algorithm for obvious reasons of computation time with this channel

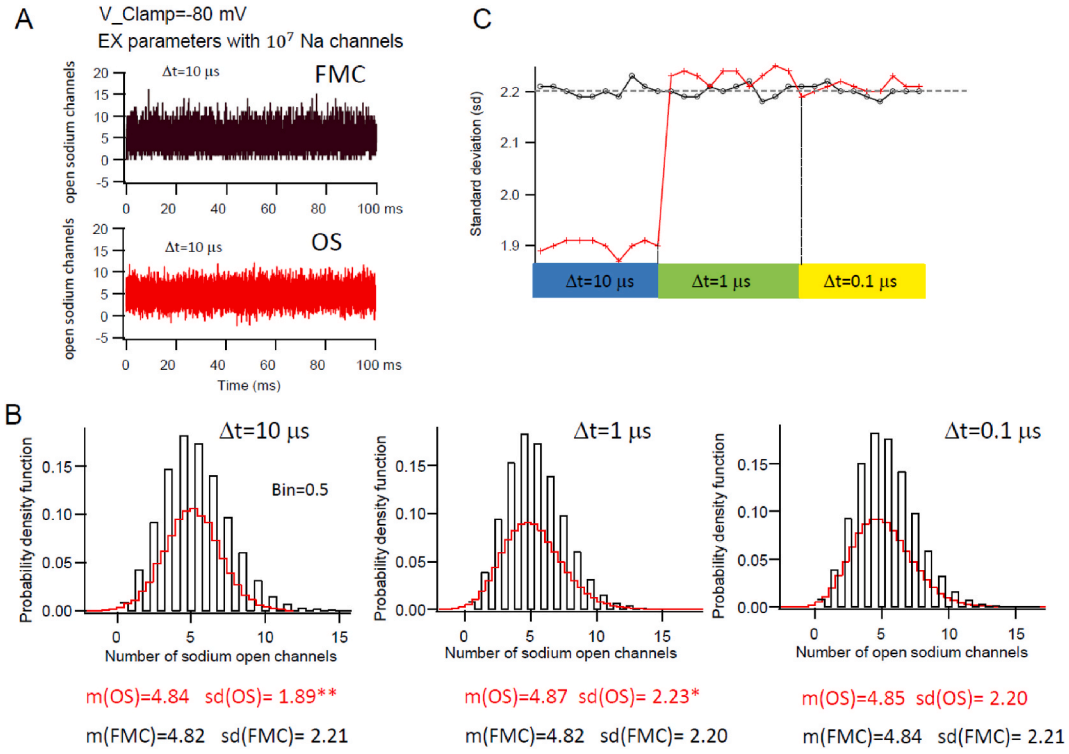


Fig. 9. Incidence of time resolution on modelled channel noise. The parameters of the model are the same as those in Fig. 8 for the evoked action potentials but the number of sodium channels is $N = 10^7$. **A.** Signal sample of duration 100 ms representing the number of sodium channels open as a function of time in voltage clamp (VC = −80 mV) simulated with a time step of 100 μs with the FMC algorithm (in high black color) and OS (below red color). **B.** Histograms of the noise showed in A (OS color red, FMC color black) made with a bin of 0.5 mV and a time resolution of 10 μs (left), 1 μs (middle) and 0.1 μs (right). The standard deviation values produced by the OS algorithm (red color) are marked with a single (small difference with FMC – 1.5%) or a double asterisk (large difference with FMC – 15%). For $\Delta t = 10$ μs, the OS algorithm greatly underestimates the noise variance (SD = 1.89 versus 2.21), slightly overestimates it for $\Delta t = 1$ μs (2.23 versus 2.21) and becomes correct for $\Delta t = 0.1$ μs (see C). **C.** Ten successive tests of the OS and FMC algorithms on a simulation corresponding to a duration of 100 ms for each of the three time resolutions: 10, 1 and 0.1 μs. These tests confirm the result illustrated in B. for $\Delta t = 10$ μs the variance produced by the OS method is each time very underestimated, for $\Delta t = 1$ μs, this variance is 8 times out of 10 slightly greater than that produced by the binomial algorithm FMC. For $\Delta t = 0.1$ μs the values are each time comparable for the two algorithms. Note the very high stability of the FMC algorithm whatever the temporal resolution. For reasons of prohibitive computation time we did not use the Gillespie algorithm of which no difference from the FMC algorithm could be noted for N = 10000 sodium channels.

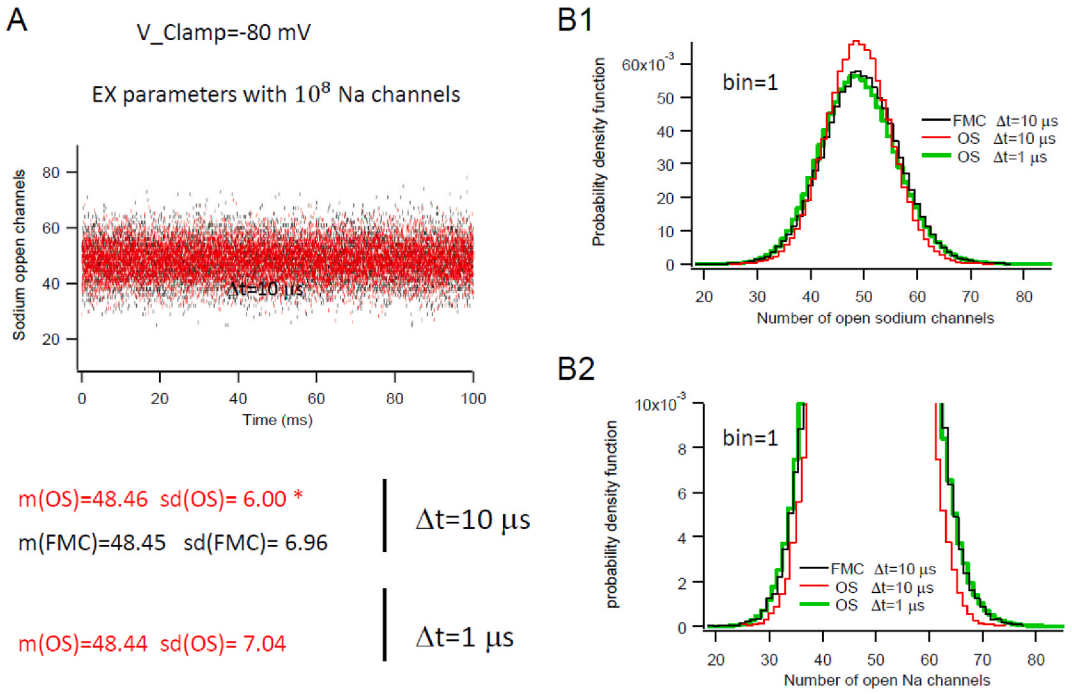


Fig. 10. Voltage clamp at hyperpolarized voltage to −80 mV. A. Parameters are those of the Ex simulation with 10^8 sodium channels. When the time step is $\Delta t = 1$ µs, the variance of the noise (number of sodium channels in the open state) produced by the OS algorithm (red trace) is underestimated SD = 6.00 (marked by an asterisk) compared to the FMC algorithm SD = 6.96 (black trace). **B1**. Histogram of the number of the number of sodium channels open for FMC algorithm (black) and the OS algorithm at $\Delta t = 10$ µs (in red) or at $\Delta t = 1$ µs (in green). The bin used is 1. **B2**: Enlargement of B1. For the lowest temporal resolution, the OS algorithm underestimate the variance. For $\Delta t = 1$ µs the two algorithms give similar variances (SD = 6.94 versus SD = 7.00 for OS). The means, variances and histograms were calculated on a signal lasting 1 s.

number, however for a number of channels of 10,000 no difference between FMC and Gillespie could be noted whatever the time resolution. We have represented the number of sodium channels open during the time and the statistics calculated relate to the mean and standard deviation of this number for 100 ms of signal.

An example corresponding to each temporal resolution and for the two algorithms is shown in Fig. 9B. All the algorithms arrive at the same result concerning the means of open sodium channels but the noise variance (standard deviation SD marked by a double asterisk) appears very underestimated at $\Delta t = 10$ µs in the OS case (SD = 1.89 vs 2.21), very slightly overestimated when $\Delta t = 1$ µs (SD = 2.23) and right for the highest resolution of $\Delta t = 0.1$ µs. The significance of these results is confirmed by ten successive tests at each resolution for the two algorithms OS and FMC showed on Fig. 9C. The FMC algorithm give exactly the same result for the three temporal resolutions. We must use a sampling frequency at least a hundred times faster to have identical noise with the OS algorithm.

Even when the noise is far above zero (i.e., when the Gaussian noise of the OS algorithm is supposed to be well outside the zone where it can encounter problems), the standard deviation is underestimated when the sampling frequency is 10 µs and becomes correct at 1 µs (Fig. 10). This confirms our previous analysis illustrated in Fig. 8, Tables 3 & 4 where the calculation of the variance of the Gaussian noise is underestimated when it is calculated without taking into account unconnected states. This problem occurs even when the noise (expressed in number of open channels) is never negative.

4.7. *Role of potassium channels*

The dynamics of potassium channels is slower but as shown in Fig. 11, at a hyperpolarized potential of −80 mV we find again the problems encountered with sodium channels. As expected, by examining again the example of Orio-Soudry of Fig. 5 for the squid giant axon with 6000 channels sodium and 1800 channels potassium in current clamp, we find that the OS-bino algorithm produces the same excitability (estimated in number of AP by unit of time) as the FMC (or Gillespie) method when $\Delta t = 10$ µs (Figure suppl. 3A). However, OS-bino does not match anymore with FMC method when Δt increases to 100 µs (Figure suppl. 3B). Nevertheless, the original OS method which uses a Gaussian approximation for the noise, continues to produce a suitable result (Fig. suppl. 3C).

The algorithm FMC differs essentially from the OS-Bino algorithm by the fact that all the transitions occurring during the interval of time Δt and not only those ensuing from the transition scheme which consequently exclude those corresponding to not directly connected states. This difference is fundamental and confers to our method a good stability of the results for a very vast range of values of temporal steps. Also, one may note that the OS algorithm realizes an approximation at the order 1 and neglects the variance of the noise with terms in dt^2. This could be harmful only in simulations demanding a precision at the order 2.

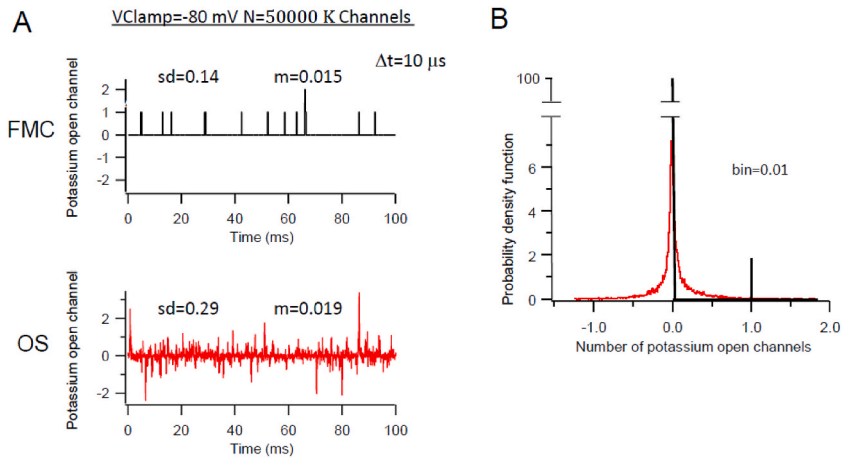


Fig. 11. Voltage clamp data with K channels. A. Voltage clamp at −80 mV for 50,000 potassium channels. The traces represent the number of open channels for the FMC algorithm (black) and the OS algorithm (red) at $\Delta t = 10$ µs. The standard deviation of the noise for OS algorithm (SD = 0.29) is almost twice the FMC one (SD = 0.15). **B**. Histogram of the number of open potassium channels for the two algorithms. The bin is 10 µs. We see that the problem encountered with sodium channels remains to a lesser extent in the case of potassium channels.

4.8. *Excitability in a « realistic » neuron model*

As we will see, in order to exhibit a correct excitability, OS algorithm requires greater temporal resolution than FMC especially at hyperpolarized voltages. To model a realistic neuron, the respective Na and K conductance are now 300 and 100 nS. The specific conductance, identical for the two types of channels (20 pS/Ch), brings the number of Na and K channels to 15,000 and 5000 respectively. The membrane capacity is 4 pF. We measure excitability with the rate of evoked AP (action potential) after applying 1000 successive step current stimulations (Fig. 12A) with a short pulse of 1 ms. We made this protocol for three different membrane potentials of −80 mV, −75 and −70 mV. The amplitude of the current pulse was calibrated a little below the threshold which triggers a PA in the deterministic case. We used a time step of $\Delta t = 10$ µs or $\Delta t = 1$ µs. We see that when the temporal resolution is $\Delta t = 10$ µs and at a membrane potential of −80 or −70 mv the OS algorithm (Fig. 12B and C) clearly underestimates the excitability compared to the Gillespie algorithms or FMC for which no difference is detected. At $\Delta t = 5$ µs (not shown) we get the same result than for $\Delta t = 10$ µs and we need to decrease Δt from 10 to 1 µs to get the true excitability with the OS algorithm.

At $V_m = -70$ mV the excitability appears as a function of the time resolution whatever the algorithm in used, this is due to the sample frequency too law compared to the band-pass of the channel noise and acting as a low pass filter. In all cases, the excitability appears higher when the time resolution decrease to 1 µs.

For $\Delta t = 10$ µs the Gillespie algorithm and FMC show no significant difference in excitability, on the other hand, the OS algorithm underestimates the excitability as compared to Gillespie or as compared to FMC. At $\Delta t = 1$ µs the OS algorithm still produces slightly but significantly less excitability than the noise produced by the FMC algorithm. (See statistical test). We can see that the excitability is more important for the FMC algorithm than for the OS algorithm.

4.9. *Excitability with semi-stochastic models*

Here simulations (Fig. 13) are done with only stochastic Na channels and K channels are deterministic. We show that, in this case, whatever the membrane potential −80 or −75 mV, the excitability for the OS model and FMC is the same as for Gillespie. But the important point is that for OS, the excitability is the same that when the two types of channels was stochastic and $\Delta t = 10$ µs (see Fig. 12). In contrast, for FMC with only Na stochastic channels the excitability is less than with the two types of stochastic channels work. Obviously, when only the stochastic K channels are active and Na deterministic, no Action potentials are elicited in all cases. We have to note that results are time resolution dependent and the excitability is higher when the time resolution is smaller. This result holds too for Gillespie simulation. For $\Delta t = 1$ µs the probability of spike is around 0.21 and fall around 0.12 when $\Delta t = 10$ µs?

4.10. *Computation time*

In order to compare the speed of the different algorithms only the core of the noise production was tested. The computation time for solving HH equations is not taken into account here. The simulations were made for a constant potential of −65 mV and with the parameters of the evoked PA simulations. The total time is equivalent to 100 s of simulation and the time step is 10 µs. The simulations begin for 5 potassium channels and this number increases two-fold in every iteration. In the determinist case, the calculation time corresponds mainly to the multiplication of the matrices of transition with the vectors of state of the number of channels for the potassium and the sodium.

With the use of a given $\Delta t = 10$ µs the execution speed of our procedure using the BIN subroutine is about twice as slow as the OS

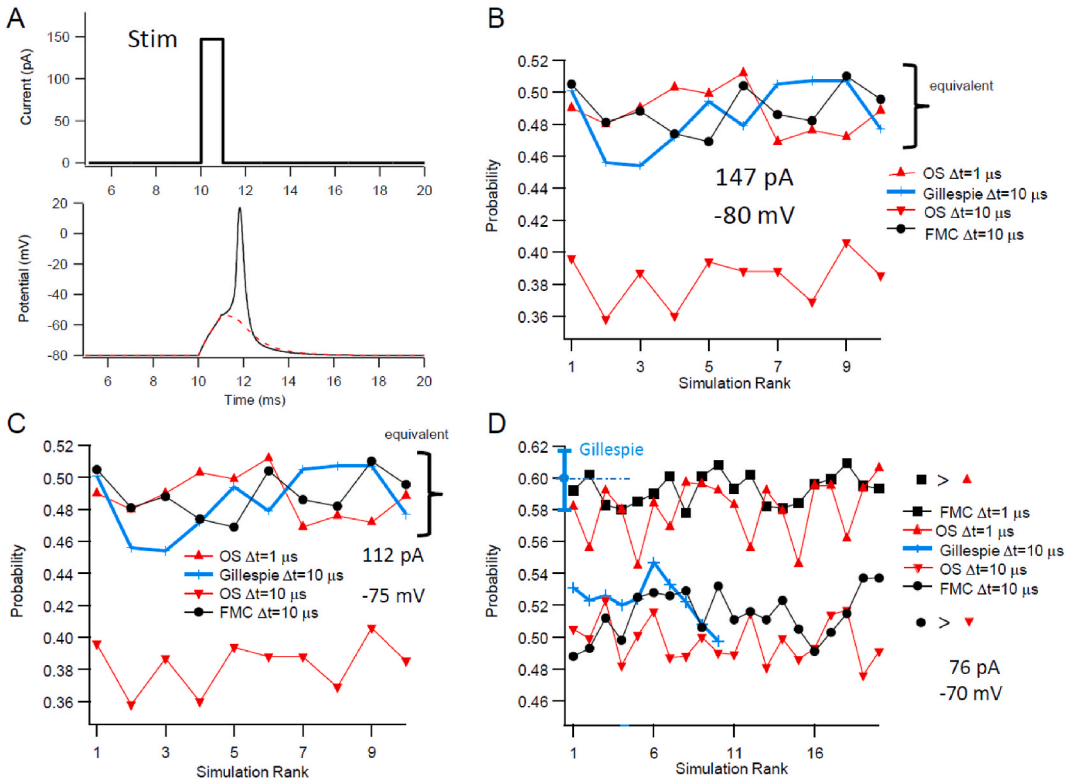


Fig. 12. Differences in excitability between models in a realistic neuron model. The Ex model has been modified into a realistic model of brain neuron with Na and K conductance set to 300 and 100 nS. The specific conductance, identical for the two types of channels (20 pS/Ch), brings the number of Na and K channels to 15,000 and 5000, respectively. The membrane capacity is 4 pF. **A**. Successive stimulations are carried out with a short pulse (1 ms) of current and the number of trials where an AP is triggered is counted (red: failure, black: triggered). Whenever the membrane potential exceeds −20 mV an action potential is counted. The simulation is made for 3 different membrane potentials (−80 mV, −75 and −70 mV). The amplitude of the current pulse is calibrated a little below the threshold which triggers an AP in the deterministic case. For the three cases, it is 147, 112 and 76 pA, respectively. The probabilities of triggering a PA for the three situations are calculated and each point in panels B, C, D represents 1000 trials. A time step of $\Delta t = 10$ μs or $\Delta t = 1$ μs is used. **B**. $V_m = -80$ mV and I = 147 pA; when the temporal resolution is $\Delta t = 10$ μs, the OS algorithm (red line + triangle) clearly underestimates the excitability compared to the Gillespie algorithms (blue line + cross) or FMC (black line + round) for which no difference is detected. However, when Δt is decreased to 1 μs the OS algorithm (red line + square) has a excitability. **C**. Same as B for $V_m = -75$ mV and I = 112 pA. The same conclusions as in B apply. **D**. Same as B, C with $V_m = -70$ mV and I = 76 pA. For $\Delta t = 10$ μs the Gillespie algorithm (blue line + cross, calculated on the first 10 points because of low speed of the Gillespie algorithm) and FMC (black line + triangle) show no significant difference in excitability. The OS algorithm (red triangle line) underestimates the excitability as compared to Gillespie as compared to FMC (20 points). For $\Delta t = 1$ μs, the excitability conditions of the simulation change and the latter increases. For reasons of prohibitive computation time, only one point (blue point with error bar equal to ± standard deviation) is represented for the Gillespie algorithm at this temporal resolution. The OS algorithm still produces slightly but significantly less excitability at this resolution than the noise produced by the FMC algorithm. (See statistical test). We can see that the excitability is more important for the FMC algorithm than for the OS algorithm. The difference between the two algorithms increases with the hyperpolarization of the membrane and reach a factor of 2 when the membrane potential equal to −80 mV.

method (Fig. 14). But our method has a speed comparable to the Diffusion Approximation (DA) methods and of the order of the Truncated-restored (TR) methods [18]. However, it is important to note that our FMC algorithm can be used with a much larger Δt and consequently can be in practice faster than the OS one.

5. Discussion

5.1. Accuracy of the FMC method over existing methods

We have shown that the OS algorithm designed by its authors to respond to the remaining problems in the "SDE" methods did not always give the right variance for channel noise (Fig. 6). This error was also present when the number of channels is very large (i.e., when this method was supposed to be right; Fig. 6). More precisely, we have shown that compared to the FMC or the Gillespie methods, the OS algorithm has two independent problems which bias the variance of the simulated noise: i) Neglecting the directly unconnected transitions when the sampling frequency is not high enough compared to the dynamics of the channel and ii) The possibility of negative

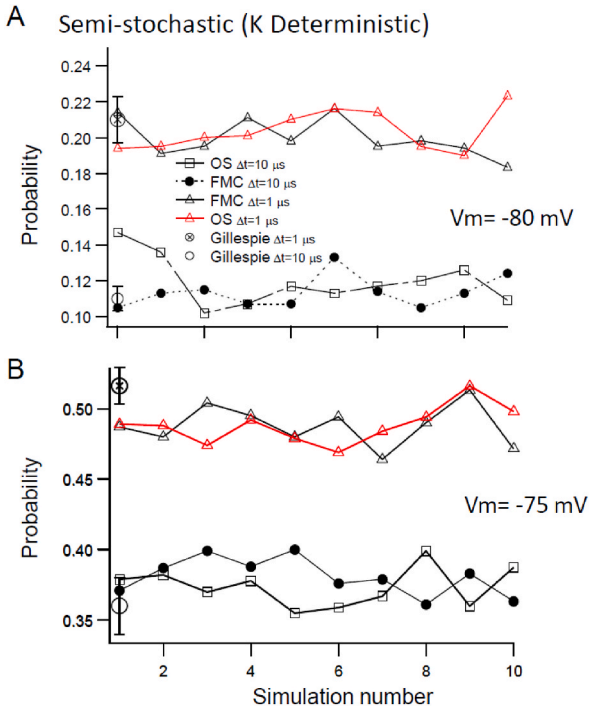


Fig. 13. Excitability with semi-stochastic models. The simulation is the same as Fig. 12 but with only stochastic Na channels (K are deterministic). **A.** At a membrane potential of −80 mV, the excitability for the OS model (mixt dotted black line and square) and FMC (dotted black line and round point) is the same as for Gillespie. In the latest case of Gillespie only one point representing 1200 trials is represented with the error bar equal±standard deviation. For OS this excitability is the same that when the two types of channels are stochastic and $\Delta t = 10$ μs. In contrast, for FMC with only Na stochastic channels the excitability is less than with the two types of stochastic channels work. Obviously, when only the stochastic K channels are active and Na deterministic, no Action potentials are elicited in all cases. **B.** Same results as in A with a membrane potential of −75 mV.

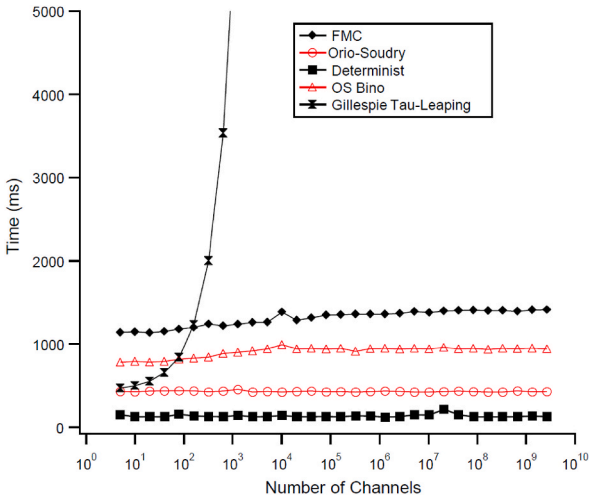


Fig. 14. Computation time of OS, FMC and Gillespie's algorithms, in log-X scale. The computation time for solving HH equations is not taken in account here. The total time is equivalent to 100 s of simulation and the time step is 10 μs. The FMC algorithm is about 2 or 3 times slower than the OS one for the same sample frequency however it should be noticed that the OS algorithm sometimes needs a sampling frequency ten times faster which makes it slower than FMC in practice.

values of the number of channels in the open state (Fig. 8C), especially for the hyperpolarized values of the potential. This last point is the most problematic. We have also presented current-clamp simulations using the EX-model which show that the deviation from binomial noise has important consequences on neuronal excitability and spike timing (Fig. 6). The EX-model differs from HH model only by a few parameters such as the a_n rate for the potassium channel which makes it more excitable, a smaller capacity, small changes

in reversal potentials for Na, K and Leak, and a higher temperature (25 °C instead 6.3 °C).

5.2. Calculation time

For reasons of prohibitive calculation time, the original Gillespie method cannot not be used when the "SDE" based methods fail. In addition, we have shown that, in all conditions, our FMC algorithm successfully reproduces the noise with the same characteristics as that of the Gillespie method (Fig. 6). This success is due to the fact that the FMC algorithm essentially produces a real binomial noise and not an imitation of the latter. Because of its slowness, the Gillespie method cannot be used when the number of channels is more than a few tens (Fig. 14). Consequently, a lot of fast methods based on stochastic differential equations "SDE" have been proposed [14,15,16,17]. These approaches have proven to be either exact but hard to use for reasons of computation difficulties, or to be convenient and rapid but at least sometimes inaccurate. Calculation time in Markovian processes can be reduced by simulating only visible transitions [22]. This method running with continuous time unit and applied to ligand-gated ion channels is also promising as it simplifies the Markovian process. The continuous-time Gillespie algorithm is briefly shown in Fig. 1, but it is the faster, discrete-time (tau-leaping) version of this algorithm that has been used as the gold standard. Gillespie's continuous-time algorithm would be even slower.

5.3. Limitations of the FMC method

The FMC method is faster than the Gillespie method, but less than the OS method at the same sample frequency (Fig. 14). However, in contrast to the OS method, it can be used with a great stability when larger time resolutions are chosen. In practice, because a slower sample frequency than OS method can be used, it can be faster which can be essential for experiments using the technique of "Dynamic Clamp" [23,24,25]. Another possible limitation of the present study is that we do not consider the precise location of the simulated ion channels. In addition, in a segmentation of a neuronal compartment, each segment is localized by its axial abscissa without precisely knowing the exact position of each channel in the segment. Biologically, it is known that channel clustering affects the opening properties [26,27]. For instance, disruption of Nav channel clustering at the axon initial segment leads to delayed AP initiation and a reduced maximal firing rate [28]. In addition, disruption of Kv2.1 channel clustering shifts the activation towards negative potential, thus enhancing the current evoked by a constant depolarization [29]. The underlying mechanism could be due to the modulation of inactivation properties [30,31].

5.4. Biological relevance

Hodgkin Huxley's model is based on the simplest interpretation of experimental data rather than direct evidence, and we cannot be sure that the resulting stochastic noise model is the best. However, if we consider the principle of parsimony and the fact that we know for example that potassium channels are effectively composed of four identical subunits, each of which contains a single voltage-sensor domain [32,33,34], it is therefore reasonable to consider the Gillespie method as the reference method for faithfully reproducing the noise of ion channels as observed experimentally. However, we must keep in mind that if we want to emulate reality as faithfully as possible, we should know the state of each channel at all times, we should know the precise moment and location of any changes in channels state. We have seen (see Fig. 12) that even Gillespie's algorithm with a constant time step can lead to different probability of evoked spikes if the temporal resolution is below a certain limit so excitability could be time resolution dependent. We have to admit that, to some extent, only channel states tracking (CST) methods are potentially capable of reproducing reality under any conditions. The MarkoLAB algorithm belongs to this category [35]. Compared to the FMC algorithm, the MarkoLAB is slow (3.8 s vs. 1.2 s for 1000 channels) and does not claim to be efficient in terms of speed. However, it offers the advantage to be a didactic tool, well suited for students who want to familiar themselves with ion channel stochastic behavior.

If the segmentation is such that the calculations must be carried out on a few dozen channels only and if we are looking to reproduce the reality as faithfully as possible from a quantitative point of view, then this last strategy (i.e., CST) is a good choice. If we are dealing with an intermediate or large number of channels and we are interested in the qualitative effects, then the FMC algorithm is recommended.

6. Conclusion

In conclusion, the FMC method described in this manuscript appears as the more robust and accurate method for simulating channel noise. The novelty of the method resides essentially in the reduction of the number of random values needed for applying Markovian models by considering in one shot the number of all channels arriving in a certain state. This method has also the advantage to be relatively faster than classical Markovian methods.

Production notes

Author contribution statement

Norbert Ankri: Conceived and designed the study; Performed the computation; Analyzed and interpreted the data; Wrote the paper.
Dominique Debanne: Analyzed and interpreted the data; Wrote the paper.

Data availability statement

No data was used for the research described in the article.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Supported by Institut National de la Santé et de la Recherche Médicale (INSERM), Centre National de la Recherche Scientifique (CNRS), Agence Nationale de la Recherche (ANR) (ANR-14-CE13-003 and ANR-21-CE16-013 to DD) and FRM (DEQ2018-0839583 to DD). We thank Drs J Berhens, R Brette and P Marcaggi for useful comments on the manuscript.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.heliyon.2023.e16953.

References

[1] A.A. Faisal, L.P.J. Selen, D.M. Wolpert, Noise in the nervous system, Nat. Rev. Neurosci. 9 (2008) 292–303, https://doi.org/10.1038/nrn2258.
[2] J.A. White, J.T. Rubinstein, A.R. Kay, Channel noise in neurons, Trends Neurosci. 23 (2000) 131–137, https://doi.org/10.1016/s0166-2236(99)01521-0.
[3] G.A. Jacobson, K. Diba, A. Yaron-Jakoubovitch, Y. Oz, C. Koch, I. Segev, Y. Yarom, Subthreshold voltage noise of rat neocortical pyramidal neurones, J. Physiol. 564 (2005) 145–160, https://doi.org/10.1113/jphysiol.2004.080903.
[4] T.F. Weiss, Cellular Biophysics, MIT Press, 1996.
[5] L.J. DeFelice, A. Isaac, Chaotic states in a random world: relationship between the nonlinear differential equations of excitability and the stochastic properties of ion channels, J. Stat. Phys. 70 (1993) 339–354, https://doi.org/10.1007/BF01053972.
[6] A.F. Strassberg, L.J. DeFelice, Limitations of the hodgkin-huxley formalism: effects of single channel kinetics on transmembrane voltage dynamics, Neural Comput. 5 (1993) 843–855, https://doi.org/10.1162/neco.1993.5.6.843.
[7] R.C. Cannon, C. O'Donnell, M.F. Nolan, Stochastic ion channel gating in dendritic neurons: morphology dependence and probabilistic synaptic activation of dendritic spikes, PLoS Comput. Biol. 6 (2010), https://doi.org/10.1371/journal.pcbi.1000886.
[8] S. Zeng, Y. Tang, P. Jung, Spiking synchronization of ion channel clusters on an axon, Phy. Rev. E Stat. Nonlin. Soft Matt. Phys. 76 (2007), 011905, https://doi.org/10.1103/PhysRevE.76.011905.
[9] E. Neher, C.F. Stevens, Conductance fluctuations and ionic pores in membranes, Annu. Rev. Biophys. Bioeng. 6 (1977) 345–381, https://doi.org/10.1146/annurev.bb.06.060177.002021.
[10] D.T. Gillespie, Exact stochastic simulation of coupled chemical reactions, J. Phys. Chem. 81 (1977) 2340–2361, https://doi.org/10.1021/j100540a008.
[11] R.F. Fox, Y. Lu, Emergent collective behavior in large numbers of globally coupled independently stochastic ion channels, Phys. Rev. E. 49 (1994) 3421–3431, https://doi.org/10.1103/PhysRevE.49.3421.
[12] H. Mino, J.T. Rubinstein, J.A. White, Comparison of algorithms for the simulation of action potentials with stochastic sodium channels, Ann. Biomed. Eng. 30 (2002) 578–587, https://doi.org/10.1114/1.1475343.
[13] I.C. Bruce, Implementation issues in approximate methods for stochastic Hodgkin-Huxley models, Ann. Biomed. Eng. 35 (2007) 315–318, https://doi.org/10.1007/s10439-006-9174-9. ; author reply 319.
[14] J.H. Goldwyn, N.S. Imennov, M. Famulare, E. Shea-Brown, Stochastic differential equation models for ion channel noise in Hodgkin-Huxley neurons, Phy. Rev. E Stat. Nonlin. Soft Matt. Phys. 83 (2011), 041908, https://doi.org/10.1103/PhysRevE.83.041908.
[15] D. Linaro, M. Storace, M. Giugliano, Accurate and fast simulation of channel noise in conductance-based model neurons by diffusion approximation, PLoS Comput. Biol. 7 (2011), e1001102, https://doi.org/10.1371/journal.pcbi.1001102.
[16] P. Orio, D. Soudry, Simple, fast and accurate implementation of the diffusion approximation algorithm for stochastic ion channels with multiple states, PLoS One 7 (2012), e36670, https://doi.org/10.1371/journal.pone.0036670.
[17] C.E. Dangerfield, D. Kay, K. Burrage, Modeling ion channel dynamics through reflected stochastic differential equations, Phy. Rev. E Stat. Nonlin. Soft Matt. Phys. 85 (2012), 051907, https://doi.org/10.1103/PhysRevE.85.051907.
[18] Y. Huang, S. Rüdiger, J. Shuai, Channel-based Langevin approach for the stochastic Hodgkin-Huxley neuron, Phy. Rev. E Stat. Nonlin. Soft Matt. Phys. 87 (2013), 012716, https://doi.org/10.1103/PhysRevE.87.012716.
[19] Y. Huang, S. Rüdiger, J. Shuai, Accurate Langevin approaches to simulate Markovian channel dynamics, Phys. Biol. 12 (2015), 061001, https://doi.org/10.1088/1478-3975/12/6/061001.
[20] D. Pezo, D. Soudry, P. Orio, Diffusion approximation-based simulation of stochastic ion channels: which method to use? Front. Comput. Neurosci. 8 (2014) 139, https://doi.org/10.3389/fncom.2014.00139.
[21] E. Stadlober, Sampling from Poisson, Binomial and Hypergeometric Distributions: Ratio of Uniforms as a Simple and Fast Alternative, 1989. https://graz.pure.elsevier.com/en/publications/sampling-from-poisson-binomial-and-hypergeometric-distributions-r. (Accessed 19 June 2020).
[22] D.R. Schmidt, R.F. Galán, P.J. Thomas, Stochastic shielding and edge importance for Markov chains with timescale separation, PLoS Comput. Biol. 14 (2018), e1006206, https://doi.org/10.1371/journal.pcbi.1006206.
[23] E. Carlier, V. Sourdet, S. Boudkkazi, P. Déglise, N. Ankri, L. Fronzaroli-Molinieres, D. Debanne, Metabotropic glutamate receptor subtype 1 regulates sodium currents in rat neocortical pyramidal neurons, J. Physiol. 577 (2006) 141–154, https://doi.org/10.1113/jphysiol.2006.118026.
[24] A. Hasenstaub, S. Otte, E. Callaway, T.J. Sejnowski, Metabolic cost as a unifying principle governing neuronal biophysics, Proc. Natl. Acad. Sci. U.S.A. 107 (2010) 12329–12334, https://doi.org/10.1073/pnas.0914886107.
[25] L.S. Milescu, T. Yamanishi, K. Ptak, J.C. Smith, Kinetic properties and functional dynamics of sodium channels during repetitive spiking in a slow pacemaker neuron, J. Neurosci. 30 (2010) 12113–12127, https://doi.org/10.1523/JNEUROSCI.0445-10.2010.
[26] R.E. Dixon, M.F. Navedo, M.D. Binder, L.F. Santana, Mechanisms and physiological implications of cooperative gating of clustered ion channels, Physiol. Rev. 102 (2022) 1159–1210, https://doi.org/10.1152/physrev.00022.2021.

[27] J. Zhang, C.M. Carver, F.S. Choveau, M.S. Shapiro, Clustering and functional coupling of diverse ion channels and signaling proteins revealed by super-resolution STORM microscopy in neurons, Neuron 92 (2016) 461–478, https://doi.org/10.1016/j.neuron.2016.09.014.
[28] D. Zhou, S. Lambert, P.L. Malen, S. Carpenter, L.M. Boland, V. Bennett, AnkyrinG is required for clustering of voltage-gated Na channels at axon initial segments and for normal action potential firing, J. Cell Biol. 143 (1998) 1295–1304, https://doi.org/10.1083/jcb.143.5.1295.
[29] H. Misonou, D.P. Mohapatra, E.W. Park, V. Leung, D. Zhen, K. Misonou, A.E. Anderson, J.S. Trimmer, Regulation of ion channel localization and phosphorylation by neuronal activity, Nat. Neurosci. 7 (2004) 711–718, https://doi.org/10.1038/nn1260.
[30] L. Lewin, E. Nsasra, E. Golbary, U. Hadad, I. Orr, O. Yifrach, Molecular and cellular correlates in Kv channel clustering: entropy-based regulation of cluster ion channel density, Sci. Rep. 10 (2020), 11304, https://doi.org/10.1038/s41598-020-68003-4.
[31] N. Zandany, S. Marciano, E. Magidovich, T. Frimerman, R. Yehezkel, T. Shem-Ad, L. Lewin, U. Abdu, I. Orr, O. Yifrach, Alternative splicing modulates Kv channel clustering through a molecular ball and chain mechanism, Nat. Commun. 6 (2015) 6488, https://doi.org/10.1038/ncomms7488.
[32] W.N. Zagotta, T. Hoshi, J. Dittman, R.W. Aldrich, Shaker potassium channel gating. II: transitions in the activation pathway, J. Gen. Physiol. 103 (1994) 279–319, https://doi.org/10.1085/jgp.103.2.279.
[33] W.N. Zagotta, T. Hoshi, R.W. Aldrich, Shaker potassium channel gating. III: evaluation for activation, J. Gen. Physiol. 103 (1994) 321–362, https://doi.org/10.1085/jgp.103.2.321.
[34] N.E. Schoppa, F.J. Sigworth, Activation of Shaker potassium channels. III. An activation gating model for wild-type and V2 mutant channels, J. Gen. Physiol. 111 (1998) 313–342, https://doi.org/10.1085/jgp.111.2.313.
[35] R.R. da Silva, D.G. Goroso, D.M. Bers, J.L. Puglisi, MarkoLAB: a simulator to study ionic channel's stochastic behavior, Comput. Biol. Med. 87 (2017) 258–270, https://doi.org/10.1016/j.compbiomed.2017.05.032.