# Exact simulation of integrate-and-fire models with exponential currents

Romain Brette
Odyssee Lab (ENPC Certis/ENS Paris/INRIA Sophia)
Département d'Informatique
Ecole Normale Supérieure
45, rue d'Ulm
75230 Paris Cedex 05
France

email: brette@di.ens.fr
tel: +33 1 44 32 21 76
fax: +33 1 44 32 21 56

September 29, 2006

## Abstract

Neural networks can be simulated exactly using event-driven strategies, in which the algorithm advances directly from one spike to the next spike. It applies to neuron models for which we have 1) an explicit expression for the evolution of the state variables between spikes and 2) an explicit test on the state variables which predicts whether and when a spike will be emitted. In a previous work, we proposed a method which allows exact simulation of an integrate-and-fire model with exponential conductances, with the constraint of a single synaptic time constant. In this note we propose a method, based on polynomial root finding, which applies to integrate-and-fire models with exponential currents, with possibly many different synaptic time constants. Models can include biexponential synaptic currents and spike-triggered adaptation currents.

Keywords: integrate-and-fire, numerical methods, spiking neurons, event-driven, simulation

# 1 Introduction

General event-driven strategies have been devised for the case when spikes can be emitted only at times of incoming spikes (Mattia & Del Giudice, 2000). The relevant models are mostly simple pulse-coupled integrate-and-fire models, but it is also possible to include the effect of synaptic conductances (Rudolph & Destexhe, 2006). Event-driven strategies can be extended to the more realistic case when outgoing spikes are delayed (Rochel & Martinez, 2003; Lee & Farhat, 2001), by using provisory events which can be cancelled by incoming spikes. However, fitting neuron models to the requirements of exact event-driven simulation is generally not trivial, because one needs a guaranteed method that predicts whether and when a spike will be emitted given the present values of the state variables. In a previous work, we devised a method which allows exact event-driven simulation of integrate-and-fire models with exponential synaptic conductances (Brette, 2006), but it was limited to the case when all synaptic time constants are identical. Here we turn to integrate-and-fire models with exponential synaptic currents, in which the membrane potential $V$ and the synaptic inputs $I_i$ evolve according to the following system of differential equations:

$$
\begin{aligned}
\tau V' &= V_0 - V + \sum_{i=1}^{k} I_i \\
\tau_1 I_1' &= -I_1 \\
\tau_2 I_2' &= -I_2 \\
&\ldots \\
\tau_k I_k' &= -I_k
\end{aligned}
$$

where $V_0$ is the rest potential, $\tau$ is the membrane time constant, and $\tau_i$ are the synaptic time constants (note that for the sake of clarity, we have not included the membrane resistance; it is contained in the inputs $I_i$, which have the dimension of voltages). Incoming spikes trigger instantaneous additive changes in one or several input variables $I_j$, and an outgoing spike is emitted when $V = V_t$, upon when the voltage is reset ($V \rightarrow V_r$). This formalism includes: 1) exponential synaptic currents; 2) biexponential synaptic currents ($\exp(-t/\tau_1) - \exp(-t/\tau_2)$, simply use two variables $I_i$); 3) exponential adaptation currents (each outgoing spike triggers a negative exponential current).

Since the differential system is linear, it can be solved analytically in intervals with no spike:

$$V(t) = V_0 + (V(0) - V_0)e^{-t/\tau} + \sum_{i=1}^{k} I_i(0)\frac{\tau_i}{\tau - \tau_i}(e^{-t/\tau} - e^{-t/\tau_i}) \qquad (1)$$

It is however not trivial to tell whether and when $V(t)$ will cross the threshold $V_t$ for a given initial state $(V(0), I_1(0), \ldots, I_k(0))$. Figure 1.A shows the three difficulties than can arise (here with a response to a biexponential current): 1) evaluating the trajectory at the end of a short interval (as in discrete-time simulations) can miss the threshold crossing; 2) there are in general several threshold crossings; 3) non-linear root finding methods (e.g. Newton-Raphson) may not converge or may converge to the wrong crossing time, because the trajectory of $V(t)$ is not concave. Our contribution in this note is to describe a method which tells with $100\%$ certainty whether and when a threshold crossing will occur given an initial state. The method relies on polynomial root finding algorithms and applies to the case when the time constants are commensurable (i.e., they are related to each other by rational factors), which is the case in practice.

# 2 Spike timing computation

## 2.1 Polynomial formulation

Let $\tau_{\text{lcm}}$ be the least common multiple of $\tau$, $\tau_1$, ..., $\tau_k$. Define $c$, $c_1$, ..., $c_k$ the integers such that $\tau_{\text{lcm}} = c\tau$ and $\tau_{\text{lcm}} = c_i\tau_i$ for all $i$, and let $x = e^{-t/\tau_{\text{lcm}}}$. Then $V(t)$ (Equation 1) is the value of a polynomial at $x$, and finding $t > 0$ such that $V(t) = V_t$ means finding a root in $[0, 1]$ of the polynomial $P$ defined by:

$$P(X) = V_0 - V_t + \left( V(0) - V_0 + \sum_{i=1}^{k} I_i(0) \frac{\tau_i}{\tau - \tau_i} \right) X^c - \sum_{i=1}^{k} I_i(0) \frac{\tau_i}{\tau - \tau_i} X^{c_i} \quad (2)$$

## 2.2 The spike test

There are several algorithms to compute the number of roots of a real polynomial in a given interval $[a, b]$ (Collins & Loos, 1983; Pan, 1992; Mignotte & Stefanescu, 1999). The most popular one is based on *Sturm sequences*, which we briefly describe here.

Define $F_0 = P$, $F_1 = P'$ and for $i > 1$,$-F_i$ is the remainder of the Euclidian division of $F_{i-2}$ by $F_{i-1}$. The sequence terminates in $m \leq n$ steps (where $n$ is the degree of $P$). Then the number of roots of $P$ in any interval $[a, b]$ is $S(a) - S(b)$, where $S(x)$ is the number of sign changes in the sequence $[F_0(x), F_1(x), .., F_m(x)]$. Note that in the present case, evaluation of the polynomials is simple since $F_i(0)$ is the constant coefficient of $F_i$ and $F_i(1)$ is the sum of all coefficients. Thus, Sturm's algorithm provides us with a $O(1)$ (constant time) method to decide whether a spike will occur.

## 2.3 Computation of spike timing

Computing the spike timing means finding the largest root of $P$ in $[0, 1]$. Sturm's algorithm also answers this problem. One can use for example a simple bisection method: divide the interval in two; select the right-hand interval if it contains a root (calculate $S(a) - S(b)$ as above), otherwise select the left-hand interval; iterate until the required precision is reached. The number of iterations is proportional to the number of digits. Convergence can be accelerated using a hybrid bisection/Newton-Raphson method.

## 2.4 A quicker spike test

Although the spike test based on Sturm sequences requires a constant number of operations, it is useful to have a faster test because an event-driven simulator will call the spike test every time an incoming spike is received.

It turns out that in balanced networks the answer of the spike test is negative most of the time because the average input current is subthreshold. In the model considered in (Brette, 2006), we showed that most of the time, simple inequalities could discard the possibility of a spike. Here we describe a simple test based on another standard algorithm for polynomials, *Descartes' rule of signs* (Collins & Loos, 1983).

Consider a polynomial with real coefficients:

$$P(X) = a_0 + a_1 X + a_2 X^2 + \dots a_n X^n$$

According to Descartes' rule of signs, the number of positive roots of $P$ does not exceed the number of sign changes in the sequence $a_0$, $a_1$, ..., $a_n$ (null coefficients are discarded). In the present case, we know that the number of roots in $[0, 1]$ is even because $P(0) < 0$ and $P(1) < 0$ (provided $V_0 < V_t$). Therefore if Descartes' rule of signs predicts at most 0 or 1 root, then we know that there is none in $[0, 1]$. If the maximum number of roots predicted by the rule is 2 or more, then we must use the full Sturm's algorithm in order to obtain a certain answer.

In practice, this method is useful when there are exponential excitatory and inhibitory currents for which the excitatory time constant $\tau_e$ is smaller than the inhibitory time constant $\tau_i$ (which is fortunately the typical case). Then we can see from Equation 2 that Descartes's rule of signs predicts that no spike will occur if

$$V(0) - V_0 + I_e(0)\frac{\tau_e}{\tau - \tau_e} + I_i(0)\frac{\tau_i}{\tau - \tau_i} \leq 0$$

The rule also applies when there is a slow adaptation current ($\tau_a > \tau$).

# 3 Example and discussion

## 3.1 A network of excitatory and inhibitory neurons

In order to illustrate our method, we simulated a randomly connected network (connection probability $20\%$) of $250$ excitatory and $150$ inhibitory integrate-fire neurons, with models described in (Vogels & Abbott, 2005); the network was driven by excitatory and inhibitory external events, modelled as Poisson spike trains (rates were $500$ Hz). Synaptic currents were exponential, with excitatory time constant $\tau_e = 5$ ms and inhibitory time constant $\tau_i = 10$ ms. The membrane time constant is $\tau = 20$ ms, so that $\tau_{\text{lcm}} = 20$ ms $= \tau = 4\tau_e = 2\tau_i$.

The code is available at `http://www.di.ens.fr/~brette/papers/Brette2006NC.htm`. We implemented our algorithm in C++ and compared it with a clock-driven algorithm with exact subthreshold integration and timestep dt$= 0.1$ ms (i.e., the differential equations are integrated exactly, but spike times are bound to a discrete time grid). The event-driven implementation is quite complex because it involves a queue management system and polynomial operations, so we also provide an implementation with Scilab, a free vector-based scientific software developed by INRIA and ENPC (`http://www.scilab.org`) which has built-in polynomial functions (note that the simulation is much slower with Scilab because programs are interpreted).

Figure 1.B shows that 97% of the time, the outcome of the spike test is negative, and that the simple test based on Descartes's rule of signs is negative 90% of the time, so that the Sturm sequence is calculated in only 10% of the cases. We simulated the network model for 10 min of biological time; the simulation time was about 20 min for the event-driven implementation and 5 min for the clock-driven implementation. The average firing rate of the network was about 8 Hz in both cases. However, Figure 1.D shows that the time-varying population rate in the clock-driven simulation lacked high frequencies ($> 100$ Hz), compared to the exact event-driven simulation (consistently, the high frequency power increased with decreasing dt).

## 3.2 Perspectives

We have described a method to implement current-based integrate-and-fire models with exponential (or biexponential) currents in an exact event-driven fashion. Our method also applies to models with spike-triggered adaptation (which is formally equivalent to an inhibitory self-connection). An important and difficult extension would be to design methods to simulate two-variable models such as those of Izhikevich (Izhikevich, 2003) and Brette and Gerstner (Brette & Gerstner, 2005), which are more realistic because the threshold is "soft" (with a quadratic or exponential term in the first differential equation).

## Acknowledgments

This work was partially supported by the EC IP project FP6-015879, FACETS, and the EADS Corporate Research Foundation.

## References

Brette, R. (2006). Exact simulation of integrate-and-fire models with synaptic conductances. *Neural Comp*, 18(8): 2004-2027.

Brette, R., & Gerstner, W. (2005). Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. *J Neurophysiol*, *94*, 3637-3642.

Collins, G., & Loos, R. (1983). Real zeroes of polynomials. In *Computer algebra: symbolic and algebraic computation (2nd ed.)* (p. 83-94). New York: Springer-Verlag.

Izhikevich, E. M. (2003). Simple model of spiking neurons. *IEEE Transactions on Neural Networks*, *14*(6), 1569-1572.

Lee, G., & Farhat, N. H. (2001). The double queue method: a numerical method for integrate-and-fire neuron networks. *Neural Netw*, *14*(6-7), 921-32.

Mattia, M., & Del Giudice, P. (2000). Efficient event-driven simulation of large networks of spiking neurons and dynamical synapses. *Neural Comput*, *12*(10), 2305-2329.

Mignotte, M., & Stefanescu, D. (1999). *Polynomials: An algorithmic approach*. Singapore: Springer-Verlag.

Pan, V. (1992). Complexity of computations with matrices and polynomials. *SIAM Rev.*, *34*(2), 225-262.

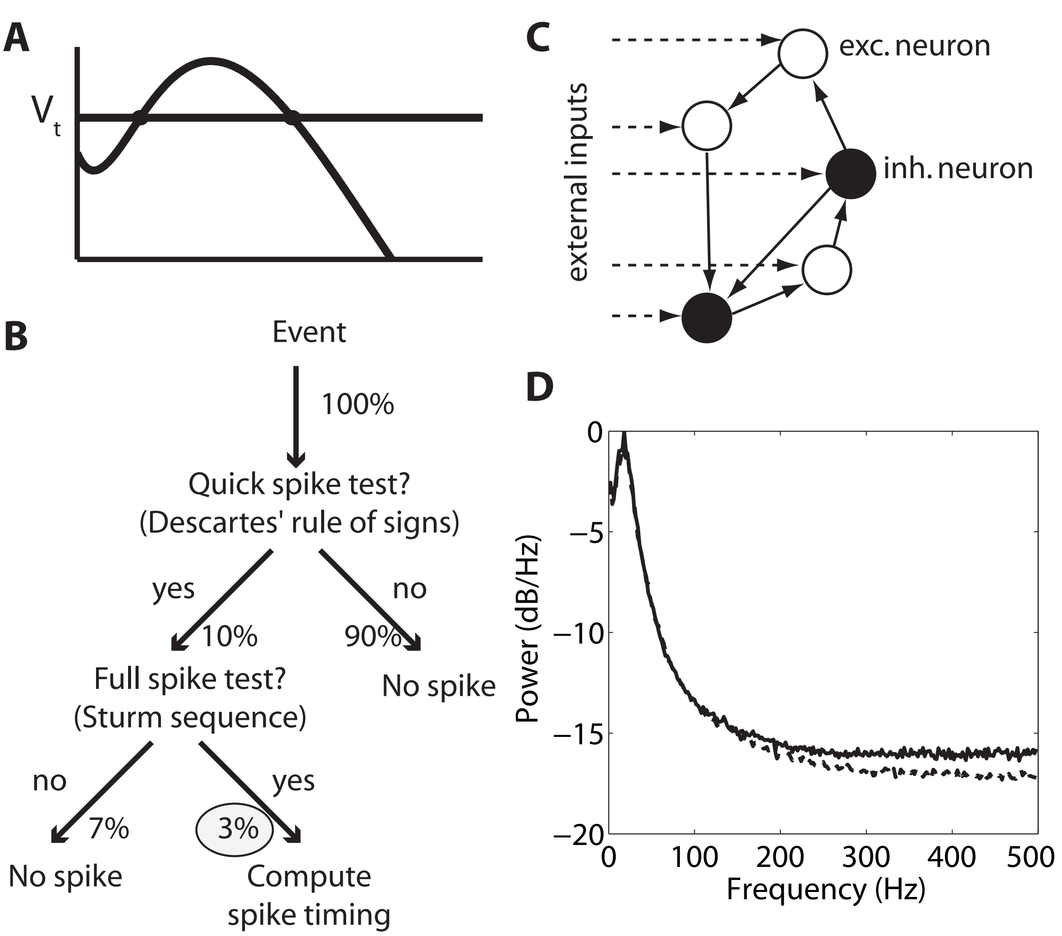
Rochel, O., & Martinez, D. (2003). An event-driven framework for the simulation of networks of spiking neurons. In *Proc. 11th european symposium on artificial neural networks* (p. 295-300). Bruges, Belgium.

Rudolph, M., & Destexhe, A. (2006). Analytical integrate-and-fire neuron models with conductance-based dynamics for event-driven simulation strategies. *Neural Comp*, *In press*.

Vogels, T. P., & Abbott, L. F. (2005). Signal propagation and logic gating in networks of integrate-and-fire neurons. *J Neurosci*, *25*(46), 10786-95.

# Figure legend

*Exact simulation of current-based integrate-and-fire models.* A. Response of an integrate-and-fire model to a biexponential synaptic current (here membrane time constant is $20$ ms, synaptic time constants are $3$ ms et $7$ ms). The horizontal line is the threshold potential ($V_t$). B. Statistics of the outcome of the spike test in the random network model. C. Architecture of the network model (see text). D. Power spectral density of the firing rate of the network simulated for 10 min with the present event-driven algorithm (thick line) and with clock-driven integration with timestep $0.1$ ms (dashed line).


A
$V_t$
B
Event
100%
Quick spike test?
(Descartes' rule of signs)
yes
no
10%
90%
Full spike test?
(Sturm sequence)
No spike
no
yes
7%
3%
No spike
Compute
spike timing
C
exc. neuron
inh. neuron
external inputs
D
Power (dB/Hz)
0
−5
−10
−15
−20
0
100
200
300
400
500
Frequency (Hz)