1 Integrate-and-Fire Neuron Model

We consider a probabilistic space $(\Omega, \mathcal{F}, \mathbb{P})$ equipped with a filtration $(\mathcal{F}_t)_{t\geq 0}$ satisfying the usual conditions.
Let Π be a Poisson random measure on $\mathbb{R}_+ \times \mathbb{R}_+$ with intensity measure $ds\,du$, where ds and du denote the Lebesgue measures on $\mathbb{R}_+$. We study the membrane potential $(V_t)_{t\geq 0}$ defined by

$$V_t = v + \int_0^t b(V_s)\,ds - \int_0^t \int_{\mathbb{R}_+} V_{s-}\,\mathbb{1}_{\{u \leq f(V_{s-})\}}\,\Pi(ds, du).$$

The process (V_t) is a piecewise deterministic Markov process: between jumps, it evolves according to an ordinary differential equation driven by the drift b, and at random times it undergoes downward jumps of size V_{s-} (i.e. it is reset to 0).

1.1 Choice of the drift b

In the simulations, we choose a classical *leaky integrate-and-fire* drift:

$$b(V) \;=\; -\frac{1}{\tau}\,(V - \mu), \tag{1}$$

where $\tau > 0$ is the membrane time constant and $\mu \in \mathbb{R}$ is a baseline or equilibrium potential. Between jumps, V_t solves the ODE

$$\frac{dV_t}{dt} \;=\; -\frac{1}{\tau}\,(V_t - \mu),$$

whose explicit solution between two times s and $t = s + \Delta t$ is

$$V_t \;=\; \mu + (V_s - \mu)\,\mathrm{e}^{-\Delta t/\tau}. \tag{2}$$

We thus do not need to discretize the drift term numerically: we can update V_t exactly between jump times by using (2).

1.2 Choice of the intensity f

The function $f : \mathbb{R} \to \mathbb{R}_+$ controls the firing intensity (spiking rate) as a function of the current potential. We take a simple threshold–linear form

$$f(V) \;=\; \alpha\,(V - V_{\mathrm{th}})_+ \;=\; \alpha \max\{V - V_{\mathrm{th}}, 0\}, \tag{3}$$

where $\alpha > 0$ is a gain parameter and V_{th} is a threshold. For $V \leq V_{\mathrm{th}}$ the intensity vanishes, so the neuron does not spike, while for $V > V_{\mathrm{th}}$ the intensity grows linearly with V.

With the leaky drift (2) and a reset to 0 after each spike, the potential V_t remains in a bounded interval (typically $[0, \mu]$). Therefore we can bound the intensity f on the relevant range of V:

$$f(V) \;\leq\; f_{\max} \;:=\; \alpha\,(\mu - V_{\mathrm{th}})_+.$$

In practice we choose a constant $A > 0$ such that

$$A \;\geq\; f_{\max}, \tag{4}$$

and use A as a *dominating rate* in a Poisson thinning algorithm (see below). For numerical robustness we can even take $A = c\,f_{\max}$ with a safety factor $c > 1$ (e.g. $c = 1.2$).

1.3 Representation via a Poisson random measure

Formally, the Poisson random measure Π with intensity $ds\,du$ on $\mathbb{R}_+ \times \mathbb{R}_+$ can be seen as a random collection of points

$$\{(s_k, u_k)\}_{k\geq 1} \subset \mathbb{R}_+ \times \mathbb{R}_+,$$

such that, for any Borel set $B \subset \mathbb{R}_+ \times \mathbb{R}_+$, the number of points in B is Poisson distributed with mean equal to the Lebesgue measure of B, and these counts are independent on disjoint sets.

The jump term in the equation can be rewritten as a random sum over these atoms:

$$\int_0^t \int_{\mathbb{R}_+} V_{s-}\, \mathbb{1}_{\{u\leq f(V_{s-})\}}\, \Pi(ds, du) \;=\; \sum_{k:s_k\leq t} V_{s_k-}\, \mathbb{1}_{\{u_k\leq f(V_{s_k-})\}}.$$

At a point (s_k, u_k), a jump of size V_{s_k-} occurs if and only if $u_k \leq f(V_{s_k-})$; in that case $V_{s_k} = V_{s_k-} - V_{s_k-} = 0$. This exactly implements the reset mechanism of the integrate-and-fire neuron.

1.4 Simulation via thinning with an exponential clock

To simulate (V_t) up to a fixed time horizon $T > 0$, we proceed as follows.

Step 1: dominating Poisson process. We first consider a homogeneous Poisson process on $\mathbb{R}_+$ with rate A:

$$N_t \sim \mathrm{Poisson}(At),$$

which can be represented by its jump times

$$0 < S_1 < S_2 < \dots$$

with i.i.d. inter-arrival times

$$\Delta S_k := S_k - S_{k-1} \sim \mathrm{Exp}(A).$$

In practice, this means that, starting from the current time t, the next *candidate* event time is given by

$$\Delta t = -\frac{1}{A}\log U,$$

where $U \sim \mathrm{Uniform}(0, 1)$, and the proposed event occurs at time $t + \Delta t$.

Step 2: deterministic evolution between candidate events. Suppose we are at time t with membrane potential V_t. We draw an exponential waiting time Δt with rate A as above and compute the *proposed* event time

$$t_{\mathrm{prop}} = t + \Delta t.$$

If $t_{\mathrm{prop}} > T$, we simply integrate the deterministic ODE from time t to time T using the exact solution (2), and the simulation stops.

Otherwise, we evolve the membrane potential deterministically from t to t_{prop} by

$$V_{t_{\mathrm{prop}}-} = \mu + (V_t - \mu)\, \mathrm{e}^{-(t_{\mathrm{prop}}-t)/\tau}.$$

Step 3: sampling the mark and thinning. At time t_{prop} we draw independently a *mark*

$$U' \sim \text{Uniform}(0,1), \quad \text{and set } u := A\,U' \sim \text{Uniform}(0,A).$$

This corresponds to sampling a point (t_{prop}, u) from the Poisson random measure on $\mathbb{R}_+ \times [0, A]$ with intensity $ds\,du$.

We compute the current intensity

$$\lambda := f(V_{t_{\text{prop}}-}).$$

We then apply the thinning criterion:

- If $u \leq \lambda$, we declare that a spike occurs at time t_{prop}. The potential then jumps by $-V_{t_{\text{prop}}-}$, i.e.

  $$V_{t_{\text{prop}}} = V_{t_{\text{prop}}-} - V_{t_{\text{prop}}-} = 0.$$

  We record t_{prop} as a spike time.

- If $u > \lambda$, we reject the candidate event: no jump occurs at t_{prop}, and we simply set

  $$V_{t_{\text{prop}}} = V_{t_{\text{prop}}-}.$$

In both cases, we then continue the simulation from time $t := t_{\text{prop}}$ with the updated value of V_t, and repeat Steps 2–3 until we reach time T.

1.5 Recording the trajectory

For plotting purposes, we additionally record the trajectory V_t on a regular time grid $0 = t_0 < t_1 < \cdots < t_n = T$ with time step Δt_{rec} (e.g. $\Delta t_{\text{rec}} = 10^{-3}$). This recording grid is *independent* of the actual jump times.

Whenever a proposed event time t_{prop} lies between two recording times, we:

1. Integrate the ODE exactly (2) from the current time up to each intermediate recording time, storing the corresponding V_{t_k}.
2. Then integrate once more from the last recorded time up to t_{prop} to obtain $V_{t_{\text{prop}}-}$, and apply the thinning step (spike or no spike).

1.6 Summary of the algorithm

Putting everything together, the simulation of $(V_t)_{0\leq t\leq T}$ is:

1. Initialize $t \leftarrow 0$, $V \leftarrow v$, spike time list $\mathcal{S} \leftarrow \emptyset$.
2. While $t < T$:
   (a) Sample $\Delta t \sim \text{Exp}(A)$ and set $t_{\text{prop}} = t + \Delta t$.
   (b) If $t_{\text{prop}} \geq T$, integrate the ODE from t to T using (2), record V_T, and stop.
   (c) Otherwise, integrate the ODE from t to t_{prop} using (2) to obtain $V_{t_{\text{prop}}-}$, and set $t \leftarrow t_{\text{prop}}$.
   (d) Compute $\lambda = f(V_{t-})$ and sample $u \sim \text{Uniform}(0, A)$.
   (e) If $u \leq \lambda$, then:

   $$\text{spike occurs at } t, \quad V_t \leftarrow 0, \quad \mathcal{S} \leftarrow \mathcal{S} \cup \{t\}.$$

   Otherwise, leave V_t unchanged.

This algorithm is exactly what is implemented in the Python code: the evolution between spikes uses the closed-form solution of the drift ODE, and the spike times are generated by repeatedly sampling exponential inter-arrival times (for the dominating Poisson process of rate A) together with uniform marks in $[0, A]$ to recover the desired state-dependent intensity $f(V_{t-})$ via thinning of the Poisson random measure.