An Introductory Course in Computational Neuroscience—Paul Miller (Notes)

Malcolm

Started 14 Dec 2024

Contents

0.1 xLIF

### 0.1.1 Modelling the Leaky membrane potential

Nernst Potential

The *Nernst potential* E_A of an ion A of charge z_A with intracellular concentration $[A_{in}]$ and extracellular concentration $[A_{out}]$ is given by

$$E_A = \frac{k_B T}{z_A q_e} \ln \left(\frac{[A_{out}]}{[A_{in}]} \right)$$

where T is the temperature in Kelvin, k_B the Boltzmann constant ($1.39 \times 10^{-23} J K^{-1}$) (which converts units of temperature to units of thermal energy). q_e is the fundamental electronic charge ($1.60 \times 10^{-19} C$).

Model

Considering this representation of a neuron's membrane:
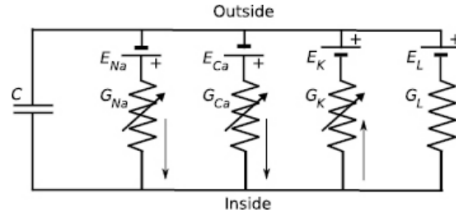
If all channels with variable conductance are closed, then the current will only flow through the leak channels (subscript L) until the cell membrane is at the leak potential E_L. The current through a channel is given by

$$I_t = G_t(V_m - E_t)$$

Where G_t represents conductance and E_t the nernst potential; t represents the type of channel.

(next page)

Equilibrium
When the cell is at equilibrium the different currents balance each other out and sum to zero:
$$I_m = \sum_t I_t = \sum_t G_t(V_m - E_t) = 0$$

In the context of this current model this can be rewritten as

$$G_{Na}(V_m - E_{Na}) + G_{Ca}(V_m - E_{Ca}) + G_K(V_m - E_K) + G_L(V_m - E_L)$$

Solving for V_m we can see that the *resting membrane potential*—where no net current flows, is the weighted average of the individual Nernst potentials:

$$V_m = \frac{G_{Na}E_{Na} + G_{Ca}E_{Ca} + G_K E_K + G_L E_L}{G_{Na} + G_{Ca} + G_K + G_L}$$

The derivation of the resting membrane potential is typically more complicated.

Leaky membrane potential
Here we consider the passive properties of the cell, where the variable conductance of all channels are fixed. With this the we treat the circuit as having a single 'leak' conductance and potential.

The membrane potential is generated by the charge stored on the membrane; it depends on both the stored charge and the membrane's capacitance C_m via the equation
$$Q = C_m V_m$$

The current is defined as positive when it flows *out* of the cell; with that we have
$$\frac{dQ}{dt} = -I_m = -G_L(V_m - E_L)$$

Fixing the capacitance we obtain the dynamics of the resting membrane potential as
$$C_m \frac{dV_m}{dt} = G_L(E_L - V_m)$$

This is a linear first order ODE.

0.1.2 Solution for Leaky ODE

We had the dynamics of the resting membrane potential as

$$C_m \frac{dV_m}{dt} = G_L(E_L - V_m)$$

Expressing in standard form and solving by integrating factor:

$$\frac{dV_m}{dt} + \frac{G_L}{C_m} V_m = \frac{G_L}{C_m} E_L$$

$$V_m = \frac{1}{\exp\left(\frac{G_L}{C_m} t\right)} \left(\int \exp\left(\frac{G_L}{C_m} t\right) \cdot \frac{G_L}{C_m} E_L \, dt + c \right)$$

To simplify we define the *time constant* $\tau_m = C_m/G_L$:

$$V_m = \frac{1}{\exp\left(\frac{t}{\tau_m}\right)} \left(\int \exp\left(\frac{1}{\tau_m} t\right) \cdot \frac{1}{\tau_m} E_L \, dt + c \right)$$

$$= \exp\left(-\frac{t}{\tau_m}\right) \left(\exp\left(\frac{t}{\tau_m}\right) E_L + A \right)$$

$$= E_L + \exp\left(-\frac{t}{\tau_m}\right) \cdot A$$

At initial condition $V_m(0)$:

$$V_m(0) = E_L + A \implies A = V_m(0) - E_L$$

With that we have the solution

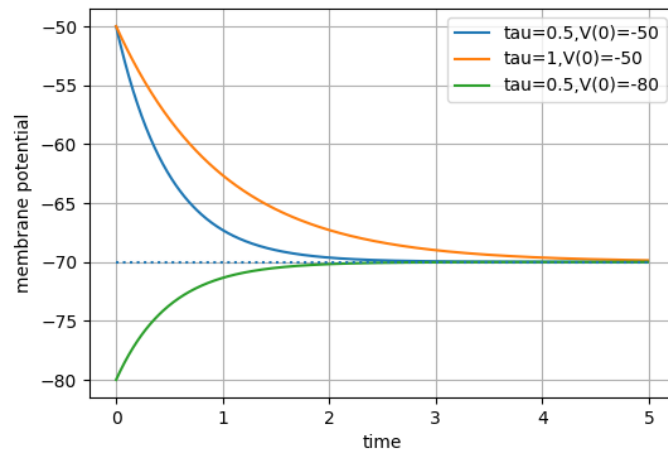$$V_m = E_L + (V_m(0) - E_L) \exp\left(-\frac{t}{\tau_m}\right)$$

(next page)

Illustrated

See that the equation tends to E_L, and that τ_m dictates how fast this decay occurs (thus the name). Illustrated here using code from `leakymembrane.py`:

$$V_m = E_L + (V_m(0) - E_L) \exp\left(-\frac{t}{\tau_m}\right)$$

0.1.3 Leaky-Integrate-and-Fire

The LIF introduces a framework on which we can build more realistic models of neurons. It is essentially the intial model for the leaky membrane with an additional term I_{app} modelling an applied current. The spike is modelled by the membrane potential being reset to a low (hyperpolarised) value V_{reset} after the potential reaches some threshold V_{th} (threshold potential):

$$C_m \frac{dV_m}{dt} = G_L(E_L - V_m) + I_{\text{app}}; \text{if } V_m > V_{th} \text{ then } V_m \mapsto V_{\text{reset}}$$

Notice that an actual 'spike' shape hasn't been modelled, and would have to be put in by hand before V_m is reset; spike times are recorded at the time when the membrane potential crosses the threshold.

The following was simulated using code from `LIFfunc.py`:
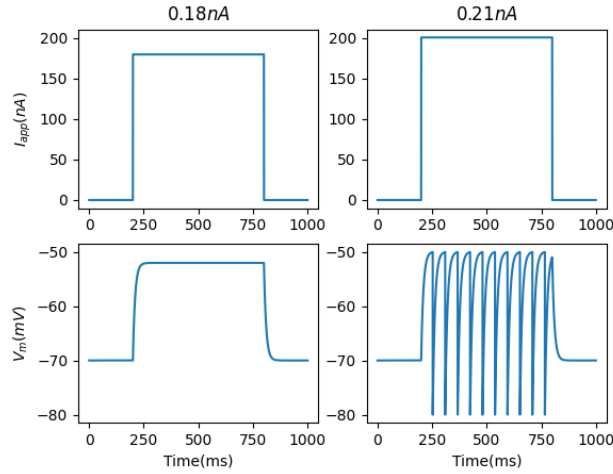
Threshold current
See that there is an insufficient level for I_{app} where no 'firing' occurs (the membrane potential does not reach threshold and the model does not spike). Setting $dV_m/dt = 0$ allows us to obtain the *steady state* membrane potential as

$$V_m^{ss} = E_L + I_{\text{app}}/G_L$$

If the steady state is below threshold then the model does not fire. By setting $V_m^{ss} = V_{th}$ we can obtain the *threshold current* I_{th}—the minimum applied current required to elicit firing:

$$I_{th} = G_L(V_{th} - E_L)$$

0.1.4   Firing rate-Current (F-I) curve of the LIF model

For the LIF (and fixed applied current), we can come up with a closed-form way of determining the time for the neuron's membrane potential to increase from its reset value to the threshold.

Solution for LIF model with constant I_{app}
Ignoring the hard reset part of the LIF (this doesn't matter since we are trying to find time between a period where the model doesn't reset), we have

$$C_m \frac{dV_m}{dt} = G_L(E_L - V_m) + I_{\text{app}}$$

When the applied current is fixed this equation can be solved:

$$V_m(t) = V_m^{ss} + (V_m(0) - V_m^{ss}) \exp\left(-\frac{t}{\tau_m}\right)$$

where V_m^{ss} denotes the steady state membrane potential.

Interspike Interval (ISI)
We want the time T when the next spike is produced, so we have $V_m(T) = V_{th}$ and solve for it. We also set the initial condition to V_{reset} since we want the time between reset and threshold:

$$V_{th} = V_m(T) = V_m^{ss} + (V_{\text{reset}} - V_m^{ss}) \exp\left(-\frac{T}{\tau_m}\right)$$

Rearranging:
$$\exp\left(-\frac{T}{\tau}\right) = \frac{V_{th} - V_m^{ss}}{V_{\text{reset}} - V_m^{ss}} = \frac{V_m^{ss} - V_{th}}{V_m^{ss} - V_{\text{reset}}}$$

See that the term on the right must be positive and less than 1 for a solution to exist with $T > 0$; since $0 < e^{-x} < 1$ for all real $x > 0$.

This reflects the fact that we can only calculate the time between spikes if $V_m^{ss} > V_{th}$ for spiking to occur in the first place. (If $V_{\text{reset}} < V_m^{ss} < V_{th}$ the right side would be negative; if $V_m^{ss} < V_{\text{reset}} < V_{th}$ it would be greater than 1.)

Solving for T gives us the time from one spike to the next—the *interspike interval* (ISI):

$$\text{ISI} = T = -\tau_m \ln\left(\frac{V_m^{ss} - V_{th}}{V_m^{ss} - V_{\text{reset}}}\right) = \tau_m \ln\left(\frac{V_m^{ss} - V_{th}}{V_m^{ss} - V_{\text{reset}}}\right)$$

(next page)

Firing rate

The firing rate is the inverse of the ISI:

$$f(I_{app}) = \frac{1}{\text{ISI}} = \frac{1}{\tau_m \ln\left(\frac{V_m^{ss} - V_{th}}{V_m^{ss} - V_{\text{reset}}}\right)} = \frac{1}{\tau_m \ln\left(\frac{E_L + I_{\text{app}}/G_L - V_{th}}{E_L + I_{\text{app}}/G_L - V_{\text{reset}}}\right)}$$

This is a rare case where we have a closed form solution for the firing rate curve.

0.1.5 Solution for LIF with fixed applied current and no reset

Ignoring the hard reset part of the LIF and fixing I_{app}, we have

$$C_m \frac{dV_m}{dt} = G_L(E_L - V_m) + I_{\text{app}}$$

This can be solved. Where $\tau = C_m/G_L$:

$$\frac{dV_m}{dt} = \frac{G_L}{C_m}(E_L - V_m) + \frac{I_{\text{app}}}{C_m}$$

In standard form:

$$\frac{dV_m}{dt} + \frac{1}{\tau}V_m = \frac{1}{\tau}E_L + \frac{I_{\text{app}}}{C_m}$$

Solving by integrating factor:

$$V_m(t) = \frac{1}{\exp(t/\tau)}\left(\int \exp\left(\frac{t}{\tau}\right)\left(\frac{1}{\tau}E_L + \frac{I_{\text{app}}}{C_m}\right) dt + c\right)$$

$$= \exp\left(-\frac{t}{\tau}\right)\left(\tau \exp\left(\frac{t}{\tau}\right)\left(\frac{1}{\tau}E_L + \frac{I_{\text{app}}}{C_m}\right) + A\right)$$

$$= \exp\left(-\frac{t}{\tau}\right)\left(\exp\left(\frac{t}{\tau}\right)E_L + \exp\left(\frac{t}{\tau}\right)\tau\frac{I_{\text{app}}}{C_m} + A\right)$$

$$= E_L + \frac{C_m}{G_L}\frac{I_{\text{app}}}{C_m} + A\exp\left(-\frac{t}{\tau}\right)$$

$$= E_L + \frac{I_{\text{app}}}{G_L} + A\exp\left(-\frac{t}{\tau}\right)$$

Initial condition:

$$V_m(0) = E_L + \frac{I_{\text{app}}}{G_L} + A$$

$$A = V_m(0) - \left(E_L + \frac{I_{\text{app}}}{G_L}\right)$$

We have the solution

$$V_m(t) = E_L + \frac{I_{\text{app}}}{G_L} + \left(V_m(0) - \left(E_L + \frac{I_{\text{app}}}{G_L}\right)\right)\exp\left(-\frac{t}{\tau}\right)$$

(next page)

9

In relation to steady state potential

We had

$$V_m(t) = E_L + \frac{I_{\text{app}}}{G_L} + \left(V_m(0) - \left(E_L + \frac{I_{\text{app}}}{G_L}\right)\right)\exp\left(-\frac{t}{\tau}\right)$$

Recall that setting $dV_m/dt = 0$ in the LIF equation (again ignoring the hard reset) gave us the steady state membrane potential V_m^{ss}:
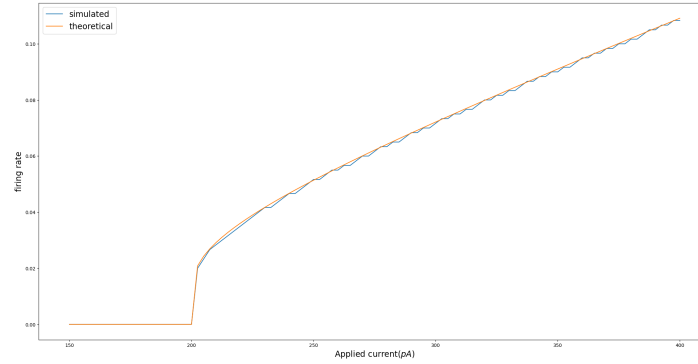
$$V_m^{ss} = E_L + I_{\text{app}}/G_L$$

See that this can be slotted into our solution:

$$V_m(t) = V_m^{ss} + (V_m(0) - V_m^{ss})\exp\left(-\frac{t}{\tau}\right)$$

0.1.6 Simulated firing rate vs Calculated firing rate

The following was plotted using code from `f-i_curves.py`:

LIF neurons were initialised with default parameters outlined in `LIFclass.py`. Applied currents ranging from 150 to $400pA$ were applied; simulated firing rate was calculated and compared to separately calculated theoretical firing rate values.

The threshold current, if calculated from the default values for this simulation, is $200pA$. See that this coincides with the simulation—spiking only starts after applied current exceeds threshold current.

0.1.7 Euler-Mayamara method—LIF with noise

Noise can be incorporated using a white noise process $w(t)$, defined as having zero mean.

Definition
Here we define noise as having variance equal to a delta function in time, so that

$$\mathbb{E}[w(t)] = 0 \quad \text{and} \quad \mathbb{E}[w(t)w(t')] = \delta(t - t')$$

(the second equality holds since mean is zero) The delta function is defined to be 0 unless $t = t'$; the idea here is that the value of $w(t)$ is uncorrelated with the value of $w(t')$ unless $t = t'$ (which would also mean that $w(t) = w(t')$).

Simulating
The delta-function has the property where its integral over all time is equal to one:

$$\int_{-\infty}^{\infty} \delta(t - t') \, dt = 1$$

it has units of inverse-time (s^{-1}); this means that the white noise function has units of the square-root of inverse time $(s^{-0.5})$ (this is important in simulations— it indicates how to scale the term with each timestep).

When we update the variable (in this case using the forward euler method), the noise term is thus included as:

$$x_{n+1} = x_n + f(x_n)\Delta t + \sigma \tilde{w}_n \sqrt{\Delta t}$$

Where $\tilde{w}_n$ is a random number selected from a distribution of 0 mean and unit variance (in order to satisfy the earlier definition—the noise at any given time point is not correlated with any other point, and has variance 1).
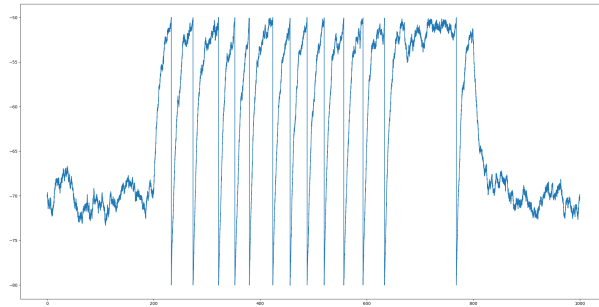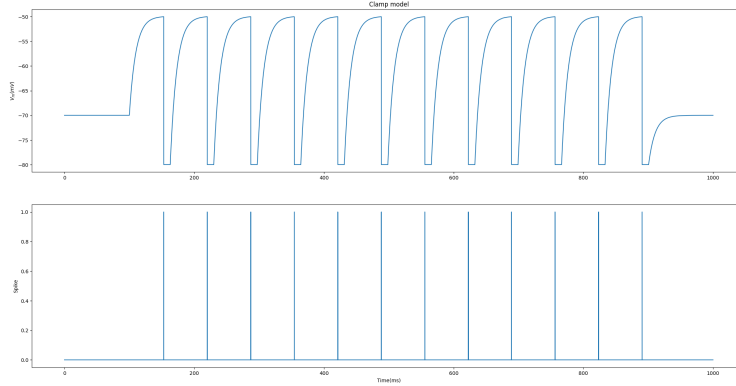

Figure: Noise incorporated with σ_I set to 0.5, see `LIFwithnoise.py`.

0.1.8 Models for the refractory period

Model 1: Forced Voltage Clamp

Fix the voltage at its reset value folllowing a spike for the duration of the refractory period.

(lower figure indicates when spike occurs, upper graph membrane potential) One disadvantage to this would be that the as the firing rate of the neuron increases, the neuron spends a greater proportion of its time in the refractory period (with the membrane potential at its low reset value). The mean membrane potential can decrease with increased input in such a model, unlike real neurons.

Model 2: Refractory Conductance

The refractory period can be mimicked by addition of a large conductance that produces a large outward (hyperpolarising) *potassium current*. This refractory conductance increases at the time of each spike and decays between spike times with a short time constant:

$$\frac{dG_{\text{ref}}(t)}{dt} = -\frac{G_{\text{ref}}(t)}{\tau_{\text{ref}}} \quad \text{after spike } G_{\text{ref}} \mapsto G_{\text{ref}} + \Delta G$$

See that since we model this as an outward potassium current, it yields an additional term of

$$G_{\text{ref}}[E_k - V_m(t)]$$

Where E_k denotes the nernst potential for potassium ions. See that when G_{ref} is much larger that the leak conductance this will essentially clamp the membrane potential at E_k.

Also see that this means that *the reset step of the LIF model can be omitted in simulations using this method*, because the step increase in G_{ref} will cause the desired drop in membrane potential. Our final model looks like

$$C_m \frac{dV_m}{dt} = G_L(E_L - V_m) + I_{\text{app}} + G_{\text{ref}}[E_k - V_m(t)]$$

(next page)

13

Refractory Conductance cont.

The upper graph plots membrane potential. The lower graph plots refractory K conductance.
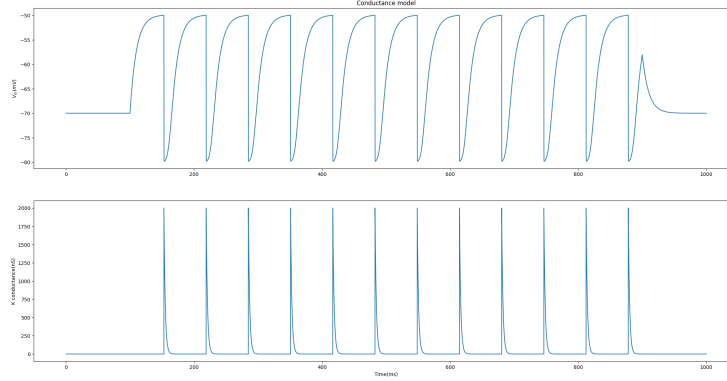
Unlike the first method, in this method the time spent at the reset value depends on the strength of the other currents entering the neuron—the stronger the other inputs the more quickly they can overcome the decaying refractory current.

Model 3: Raised Threshold

The voltage threshold for producing a spike can be raised immediately following a spike, before decaying back to its baseline level with a short time constant:

$$\frac{dV_{th}(t)}{dt} = -\frac{V_{th}^{(0)} - V_{th}(t)}{\tau_{\text{ref}}} \quad \text{after spike } V_{th} \mapsto V_{th} + \Delta V$$

This method allows the mean membrane potential to increase with firing rate—as is typically observed—while preventing a spike during the refractory period. Also see that in this model, like the second model, the refractory period is not absolute—the greater the input, the sooner the membrane potential can reach the decaying threshold potential.

See that the 'reset' in this method can be implemented either with the usual hard membrane potential reset upon reaching threshold, or with a refractory conductance (method 2).
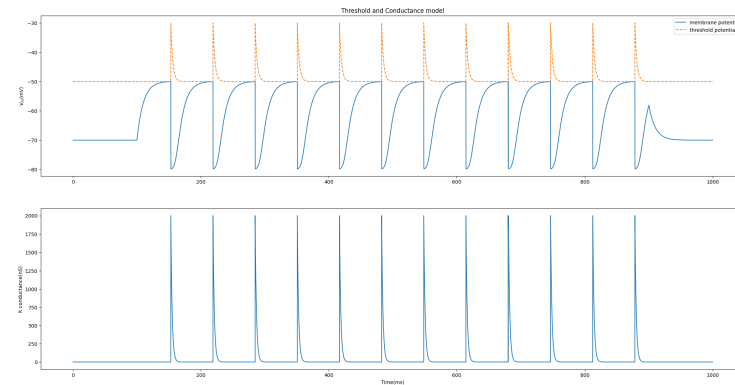
(next page)

14

Raised Threshold Cont.

Raised threshold model plotted with dynamic threshold potential dotted. (lower figure indicates when spike occurs, upper graph membrane potential)
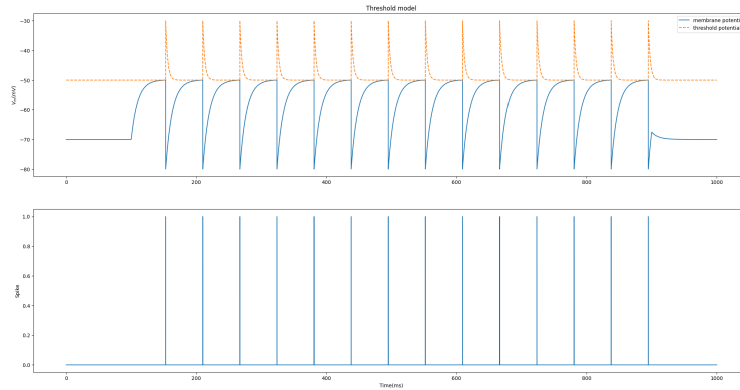
Raised Threshold with Refractory Conductance

As mentioned, methods 2 and 3 can be combined. (lower graph representing refractory conductance)

Figures were generated using `LIF_ref_models.py` using models from `LIFclass.py`.

0.1.9   Spike-Rate Adaptation

Many neurons respond to a pulse of constant current with a spike rate that decreases over time following the first spike (the first interspike interval is substantially shorter than the second and so on); this is called spike-rate/frequency adaptation.

Implementation
Spike rate adaptation can be simulated in a LIF model by incrementing a potassium conductance following a spike, similar to an earlier described model of refractory conductance, that decays between spikes.

Although the changes in the LIF model necessary for incorporating spike-rate adaptation are *qualitatively* identical to that of incorporating refractory conductance, the difference here is in the parameter values used for simulation; *the incremental conductance following each spike is much smaller for spike-rate adaptation* (we don't prevent spikes by making incrementing it large enough to clamp the membrane potential at a low value.)

Additionally the *time scale for the delay of conductance should be much longer in for spike-rate adaptation* (longer than the typical interval between spikes) so that the conductance can *accumulate over a spike train.* The full model looks like

$$C_m \frac{dV_m}{dt} = G_L(E_L - V_m) + I_{\text{app}} + G_{\text{ref}}[E_k - V_m(t)] + G_{SRA}[E_k - V_m(t)]$$

where G_{SRA} also decays to 0, just with a much larger time constant.

$$\frac{dG_{SRA}(t)}{dt} = -\frac{G_{SRA}(t)}{\tau_{SRA}} \quad \text{after spike } G_{SRA} \mapsto G_{SRA} + \Delta G$$

Illustrated:
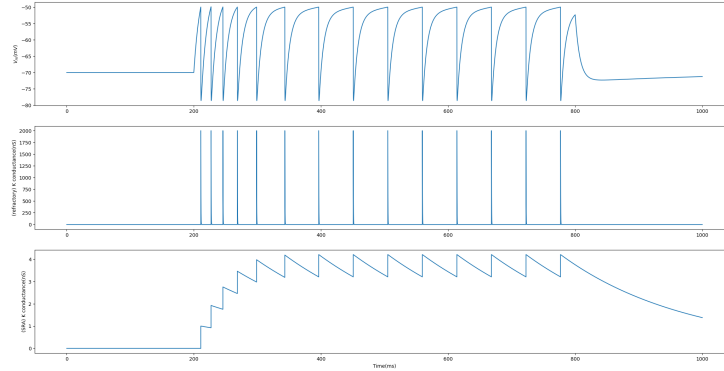
From `LIFsra.py` (this model uses a refractory conductance and raised threshold).

16

0.1.10   ELIF/AELIF model

Exponential Leaky Integrate-and-Fire (ELIF)
Up till now, the gradient of the membrane potential has decreased as it approaches threshold. In reality, an inflexion point—at which the decreasing gradient begins increasing—is visible in real cells, where the voltage accelerates up past a threshold.

Experiments also indicate that there is no fixed threshold for the membrane potential, but rather a range in which a spike may or may not be produced depending on prior inputs.

The Exponential Leaky Integrate-and-Fire (ELIF) model incorporates an additional spike-generating term in the LIF model to address these failings:

$$C_m \frac{dV_m}{dt} = G_L \left[E_L - V_m + \Delta_{th} \exp\left(\frac{V_m - V_{th}}{\Delta_{th}} \right) \right] + I_{\text{app}}$$

Where V_{th} is the threshold as before, with Δ_{th} representing the voltage-range over which the spike-generating term becomes harder to overcome with inhibition.

This models an inward, depolarising current that arises when the membrane potential reaches the *vicinity* of the standard threshold that *increases* as the membrane potential increases, creating *postive feedback* to generate a rapidly accelerating increase in membrane potential (which is the spike). In simulations of this we would set a *maximum level* of the membrane potential V_{max} at which the potential is reset as usual (rather than at threshold):

$$\text{if } V_m > V_{max} \text{ then } V_m \mapsto V_{reset}$$

Adaptive ELIF (AELIF)
The