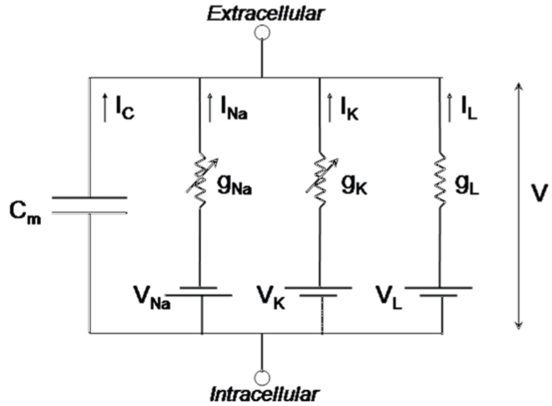
THE HODGKIN AND HUXLEY MODEL

The Hodgkin and Huxley model (HH, *J Physiol 1952*) is a mathematical description of the action potentials (AP) in neurons. It consists of a set of nonlinear differential equations, which have to be solved with numerical analysis techniques.



The cell membrane is represented as a capacitance (**C_m**), with three different ionic currents between the intra- and extra-cellular spaces:

- Na$^+$ current (**I_{Na}**): Depolarisation
- K$^+$ current (**I_K**): Repolarisation
- Leak current (**I_{leak}**): Every other current

Each membrane current (**I_X**) is computed as the product of the ionic conductance (**g_X**) and the driving force, i.e. the difference between the actual membrane potential (**V**) and the equilibrium potential of the specific ion (**V_X**). Some current conductances depend on voltage and time, and this dependence is accounted for by one or more gating variables, which represent the opening/closing of the channels, and always vary between 0 and 1.

The total current (**I_C**) consists of the sum of all the ionic currents, and the external stimulus current (**I_{Stim}**).

The Hodgkin and Huxley model has **4 state variables**:

- **V**: membrane voltage
- **m**: I_{Na} activation gate
- **n**: I_{Na} inactivation gate
- **h**: I_K activation gate

When stimulated, the neuron fires a train of APs. The shape and frequency of these APs depend on the type of stimulus.

$$C \cdot \frac{dV}{dt} = -I_C$$

$$I_C = -(I_{Na} + I_K + I_{leak})$$

$$I_{Na} = \; g_{Na} \cdot m^3 \cdot h \cdot (V - V_{Na})$$
$$I_K = \; g_K \cdot n^4 \cdot (V - V_K)$$
$$I_{leak} = \; g_{leak} \cdot (V - V_{leak})$$

$$\frac{dm}{dt} = \alpha_m(V) \cdot (1 - m) - \beta_m(V) \cdot m$$
$$\frac{dh}{dt} = \alpha_h(V) \cdot (1 - h) - \beta_h(V) \cdot h$$
$$\frac{dn}{dt} = \alpha_n(V) \cdot (1 - n) - \beta_n(V) \cdot n$$

The H&H model has been implemented in Matlab/Octave in the function file *modHH*, which takes as input the time (**t**) and the state variable vector (**X**), and gives as output the derivative of the state variables in time (**dX**). The function is called by the *MainFile_modHH* which solves the model by using the built-in function *ode15s* and gives as outputs a series of figures, showing the stimulus current and the corresponding AP.

EXERCISE 0: TEST THE HODGKIN AND HUXLEY MODEL

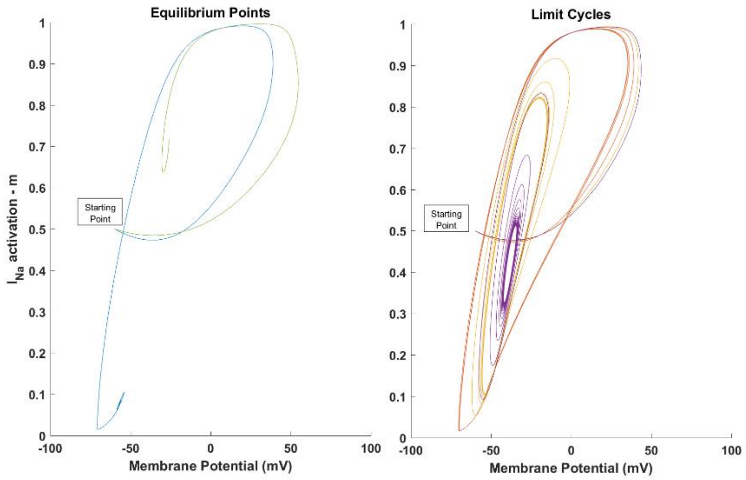
By using the Matlab/Octave files provided on our website, simulate the HH neuron model, and have a look at the outputs (as shown in the introductory slides):

- Set up Octave online and upload the two files: *MainFile_modHH.m* and *modHH.m*
- Run *MainFile_modHH.m* and explore the results.
- Have a look at the script and the function file, containing the HH model. Try to understand how these file works by reading the comments, and based on the model description given above.

- The code includes some lines that are commented (with a single %): you can uncomment them to generate additional figures, i.e. all the ionic currents (Figure 2) and all the state variables (Figure 3).
- There is also a 2D phase diagram (Figure 4), which shows the trajectory of the state variables V and m.

Quick and Easy Definitions:

A phase diagram shows how state variables evolve over time, until an equilibrium point or a limit cycle is reached. An equilibrium point is reached when the state variables remain constant over time (the trajectory in Figure 4 would end in a point). A limit cycle is reach when the state variables keep varying, but their values repeat in time (the trajectory in Figure 4 would end up drawing a loop).



- **EXERCISE 1: CHANGES OF THE STIMULUS AMPLITUDE**

By default, the model is stimulated with an impulse current of 80 uA/uF for 0.5 ms, with a cycle length (CL) of 100 ms. This means that the neuron fires every 100 ms (10 times per second): you can see one AP is elicited after each external stimulus is applied.

Modify the main file to change the magnitude of the stimulus current, run the file again, and try to understand how the AP shape and frequency change. **What happens if the stimulus current is too low or too high? Do you notice any change in the neuron behaviour and in the phase diagram?**

Hint: look at lines 24-44 in modHH and lines 18-30 in the MainFile. Change the global variable input_args.

Send your answers on Slido!

Tip: An easy and quick way to compare different stimulus magnitude is to write a for cycle that run the same MainFile multiple times, changing the stimulus current and showing the corresponding APs, all in the same figure. Open and run the file *Exercise1_solution*, to see how this can be done using Octave.

N.B. Stimulus current appear negative in the figures, because it is an inward current for the cell.

- **EXERCISE 2: CHANGES OF THE STIMULUS FREQUENCY**

With the default cycle length (CL) of 100 ms, the neuron fires every 100 ms (10 times per second), just after each external stimulus is applied. The CL defines the distance between one stimulus and the next one.

Modify the main file to change the frequency of the stimulus current, and look at how the AP shape and frequency change. **What happens when the CL is too small?**

Hint: You can do this as in Exercise 1, by modifying lines 18-30 in MainFile_modHH to change the global variable input_args, or as in Exercise1_solution, by using a for loop to compare difference CLs in one figure.

Send your answers on Slido!

NEURON AUTO-OSCILLATIONS AND RESTING: STEADY STATE ANALYSIS

Guttman et al. (*J Physiol 1980*) in experiments on squid giant axons observed that the injection of a constant current across the membrane may lead to two different cell behaviours: the **oscillatory repetitive firing state** and the **time-independent steady state**.

The transition between these two different modes of operation depend on the amplitude of the stimulus current used (**I_{Stim}**). When I_{Stim} is below a certain threshold (<**I_1**), the neuron always remains quiescent, in the **time-independent steady state** (**OFF**). When increasing I_{Stim}, the neuron switches to an **oscillatory repetitive firing state** (**ON**), producing a train of action potentials (APs). A further increase of I_{Stim} (>**I_3**) takes the neuron to a new **OFF** state, which is called **saturation**.

- **EXERCISE 3: CONSTANT STIMULUS CURRENT and ON/OFF STATES**

A) **By modifying MainFile_modHH, explore different amplitudes of a constant stimulus current (I_{Stim}), and find an estimate of the values for I_1 and I_3. How do the oscillations change with the stimulus amplitude in the ON area?**

Send your answers on Slido!

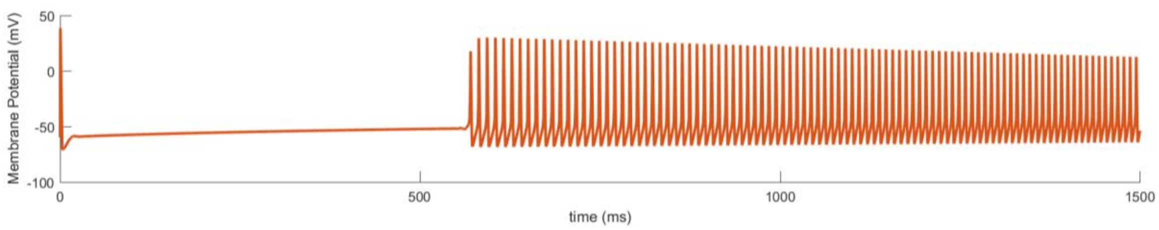
B) **Have a look at the phase diagrams (Figure 4) for at least 4 values of I_{Stim} (one in the OFF area, two in the ON area, and one in the saturation area), and observe the trajectory. What do you see?**

Send your answers on Slido!

Tip: try replacing the function *plot* with the function *comet* in Figure 4, to visualise the actual trajectory.

- **EXERCISE 4: RAMP STIMULUS CURRENT and BIFURCATIONS (OPTIONAL)**

If you use a ramp as stimulus (increasing or decreasing over time), you can cross the thresholds identified above (I_1 and I_3), going from an equilibrium point to a limit cycle, or vice versa. This is called **Bifurcation**.



Try to reproduce the figure above using a ramp stimulus, and also to generate the other type of bifurcation, from limit cycle to equilibrium point.

Hint: *look at lines 24-42 in modHH.m and lines 18-30 in the MainFile_modHH.m. Change the global variable input_args. Again, you can also use a for loop based on what proposed in Exercise1_Solution.*