EUGLOH Summer School in Neuroscience 2023

Single cell modelling with Python

Domenico Guarino

June 22, NeuroPSI

Today we will

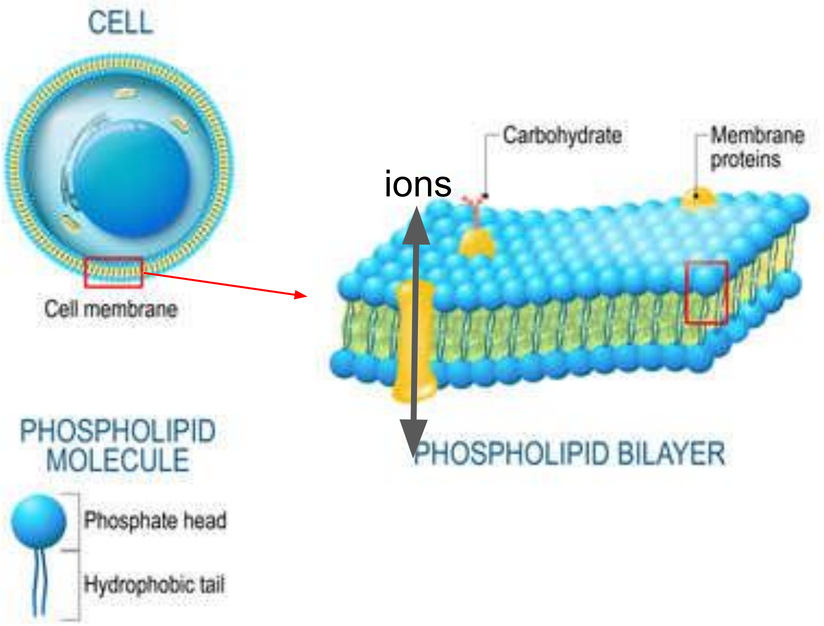
- introduce the concept of "point" neuron
- describe mathematically the evolution of its membrane potential
- start with the simplest model of a neuron: **Integrate-and-Fire**
- look at its **phase portrait**
- pros and cons of the model, identify desiderata
- introduce a more complex model: **Adaptive Exponential Integrate-and-Fire**

Cells have a membrane

All cells have a bilayer membrane.

The membrane is immersed on both sides into a fluid containing ions and molecules.

There are protein openings on the membrane, called **channels**, that let ions and molecules pass through.

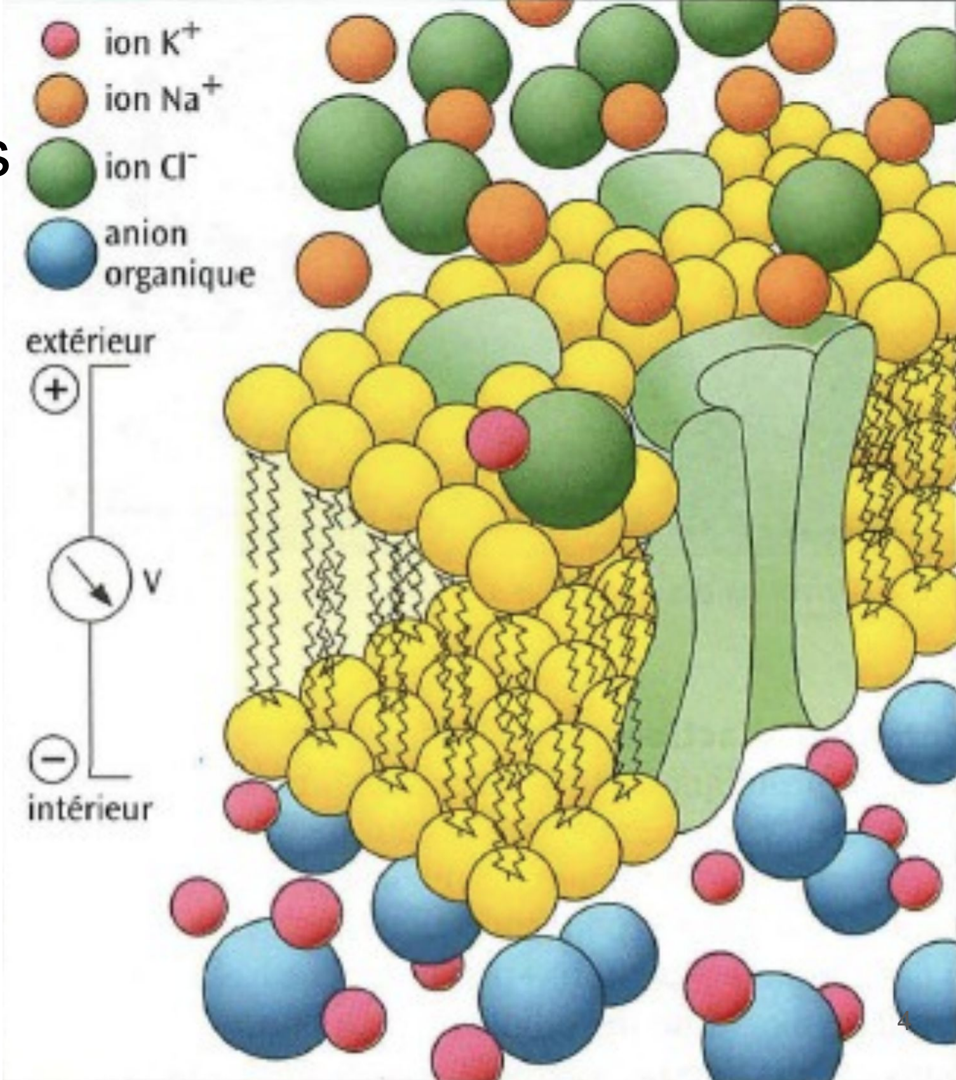


Membranes separate charges

Ions are atoms or molecules **electrically charged**.

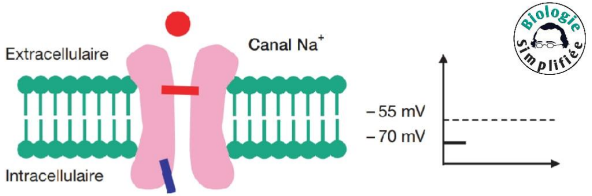
The distribution of ions across the membrane creates an **electrochemical gradient**.

Such that <u>all</u> cells have a certain **membrane potential**.



Neurons change their membrane potential over time

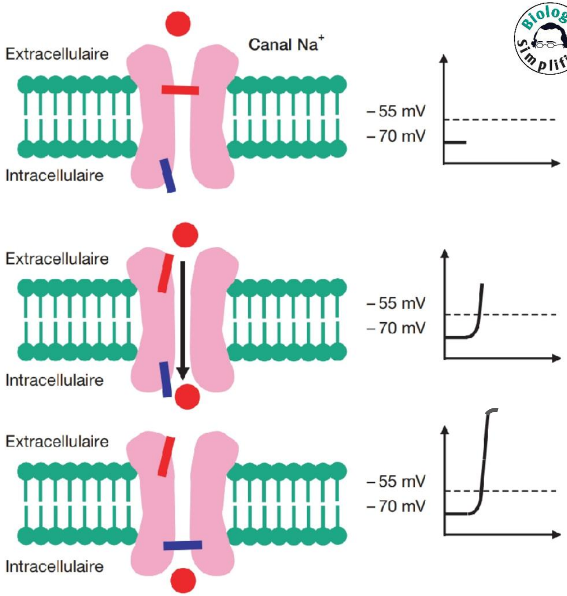
Neurons are cells equipped with **voltage-dependent** and **ion-selective** channels.



Neurons change their membrane potential over time

Neurons are cells equipped with **voltage-dependent** and **ion-selective** channels.

When the membrane potential reaches a **threshold** value,
these channels **open** shortly
and then **close**

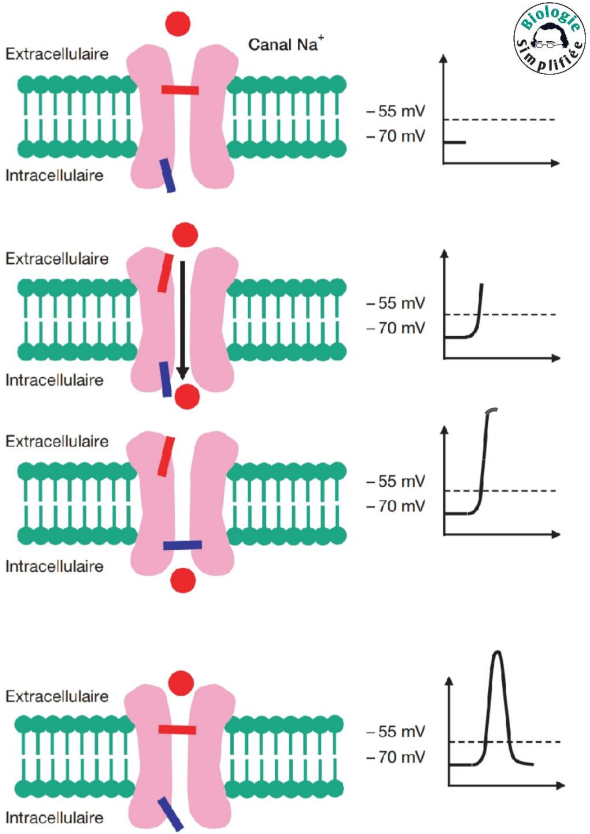


Neurons change their membrane potential over time

Neurons are cells equipped with **voltage-dependent** and **ion-selective** channels.

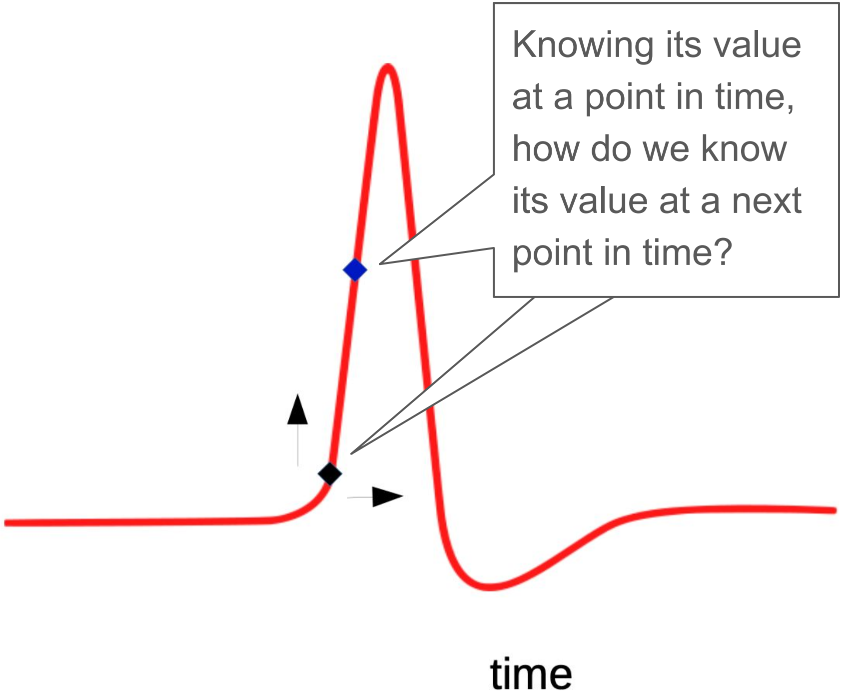
When the membrane potential reaches a **threshold** value, these channels **open** shortly and then **close** causing a surge of membrane potential.

We call these surges **spikes** or **action potentials**. And we treat them as **signals**.



Neurons **change** their membrane potential **over time**

How to describe membrane potential evolution in time?



Neurons **change** their membrane potential **over time**

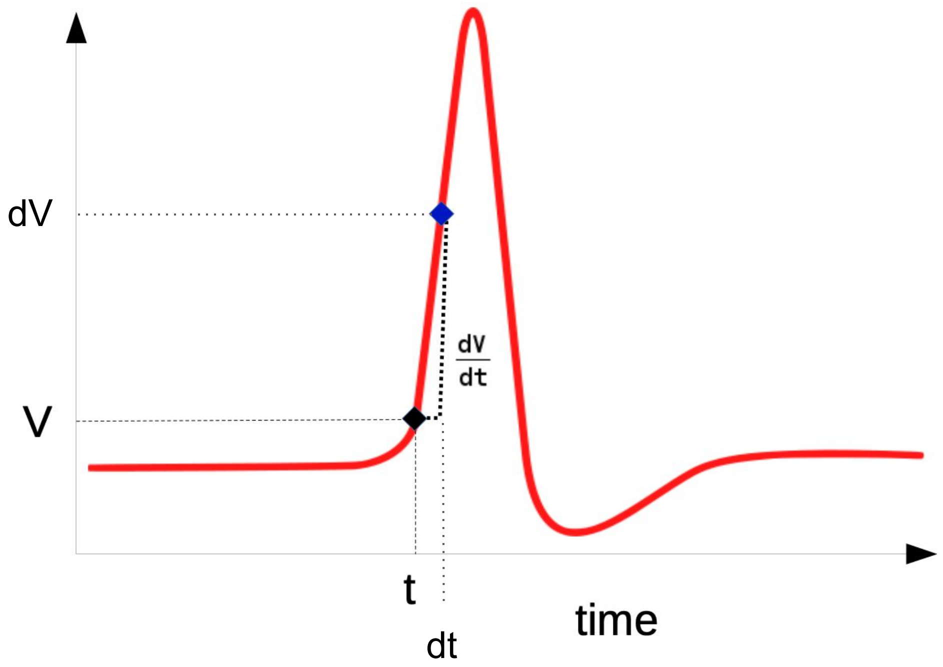
How to describe membrane potential evolution in time?

To know the “next state”, we consider small **increment** and devise a **function** of the current state V.

This can be expressed mathematically as:

$$dV/dt = \mathrm{f}(V)$$

$$\text{increment of V} = \text{function of V}$$



Neurons **change** their membrane potential **over time**

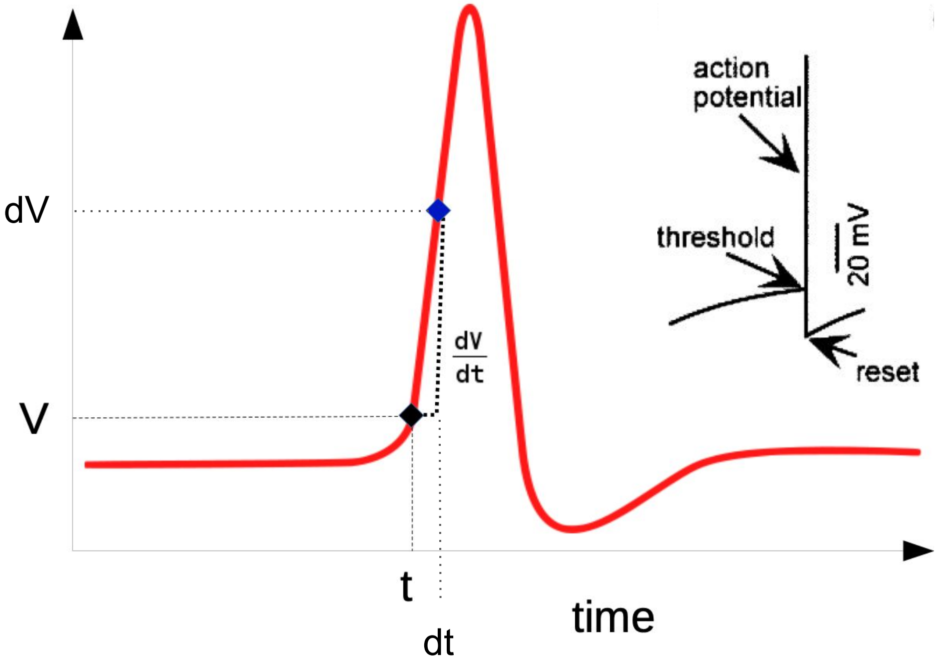
How to describe membrane potential evolution in time?

To know the “next state”,
we consider small **increment**
and devise a **function** of the
current state V.

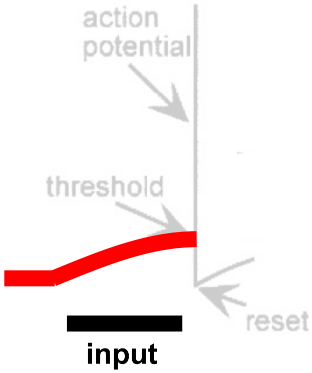
This can be expressed
mathematically as:

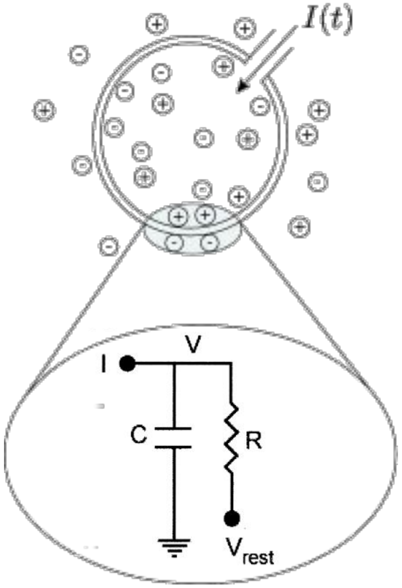
$$dV/dt = \mathrm{f}(V)$$

$$\text{increment of V} = \text{function of V}$$

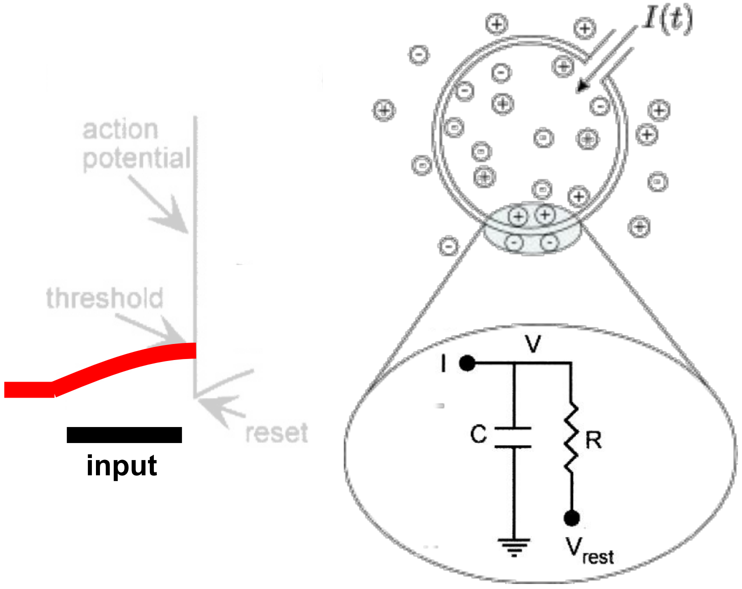


Membrane potential as an electrical circuit





Membrane potential as an electrical circuit



$$\frac{dV}{dt} = (-g_l(V - E_l) + I)/C_m$$

Leaky Integrate-and-Fire model

V membrane voltage

I injected current

Cm membrane capacitance

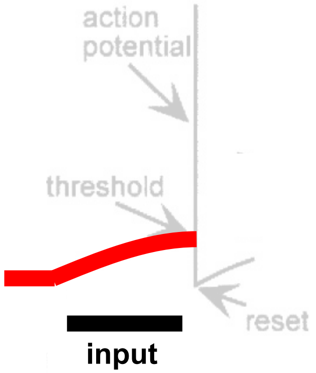
R resistance, or $g_L = 1/R$

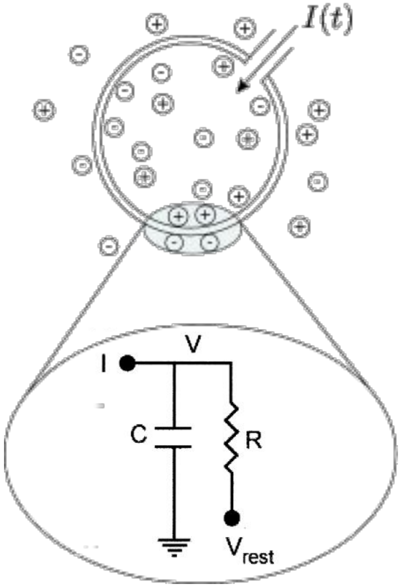
E_L equilibrium potential of the leak/rest current

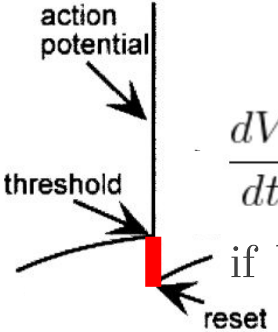
I, Cm, g_L, E_L, V_{thresh}, V_{reset}

are all fixed parameters

Membrane potential as an electrical circuit







$$\frac{dV}{dt} = (-g_l(V - E_l) + I)/C_m$$

$$\text{if } V > V_{thresh} \rightarrow V = V_{reset}$$

Leaky Integrate-and-Fire model

V membrane voltage

I injected current

Cm membrane capacitance

R resistance, or $g_L = 1/R$

E_L equilibrium potential of the leak/rest current

I, Cm, g_L, E_L, V_{thresh}, V_{reset}

are all fixed parameters

Hands on the code !!!

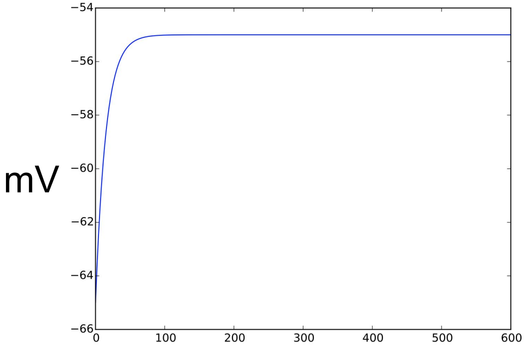
1. Open a terminal

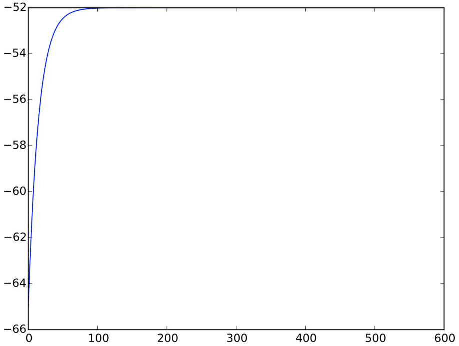
2. type:
```
cd Desktop/EUGLOH
python3 LIF.py
```

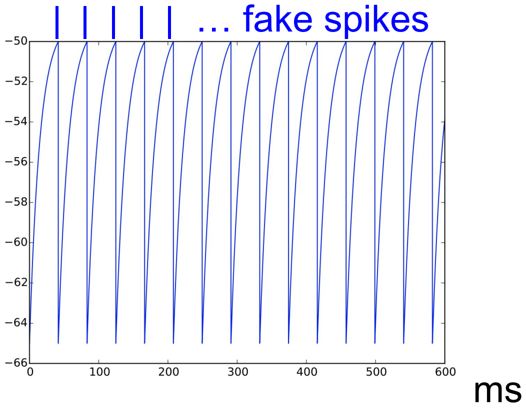
3. look into the folder `EUGLOH` for the file `LIF.png`

4. Let’s edit the file `LIF.py`

Let's comment some results







$\boxed{-g_L(V\text{-}E_L) + I \,/\, Cm}$

$V{+}I$ (for simplicity)

I = 10 pA	I = 13 pA	I = 16 pA
-65 +10	-(+65) +13	-(+65) +16
-55 mV	**-52** mV	**-49** mV > threshold

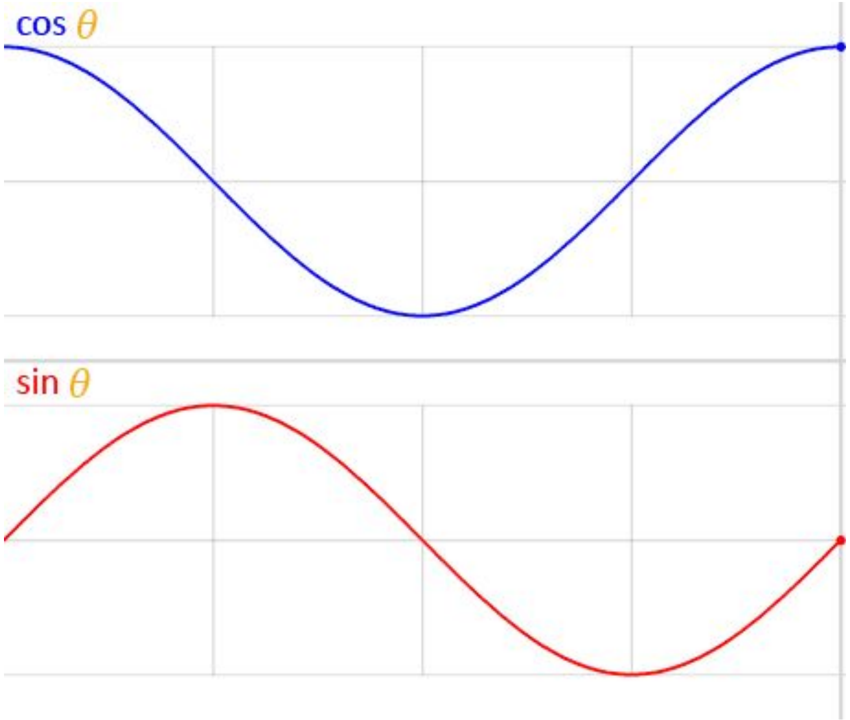
Phase portrait - a better way to see the dynamic

We introduce three concepts:

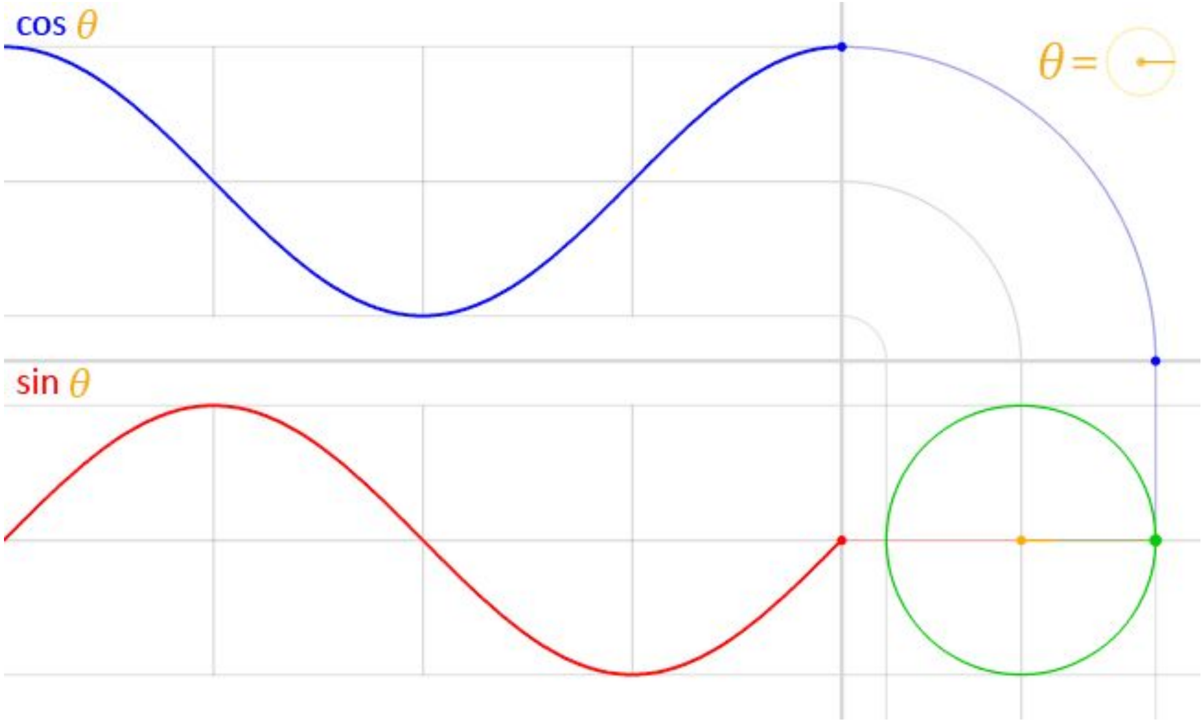
- **fixed point** where the $dV/dt = 0$
- **stability** is a property of fixed points (they can be stable or unstable)
- **nullcline** is the set of points where $\mathrm{f}(V) = 0$

The **phase plane** puts together these concepts in a portrait of the system that is different from a portrait against time.

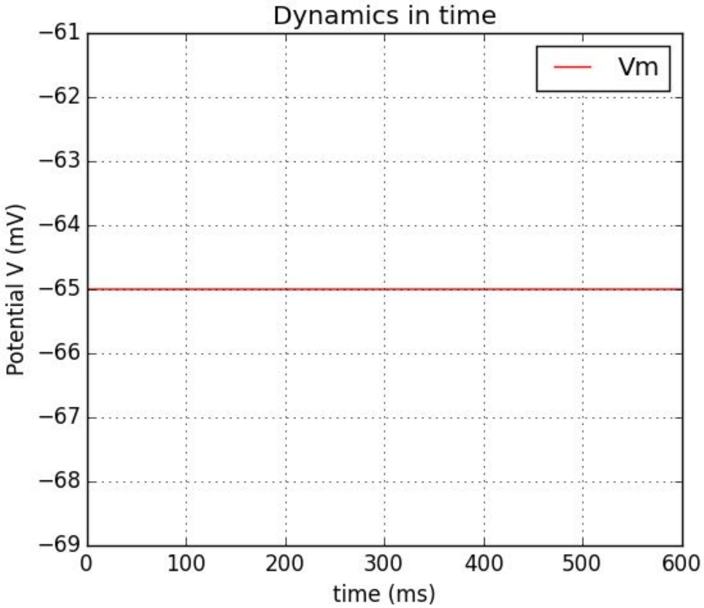
Phase portrait - a better way to see the dynamic

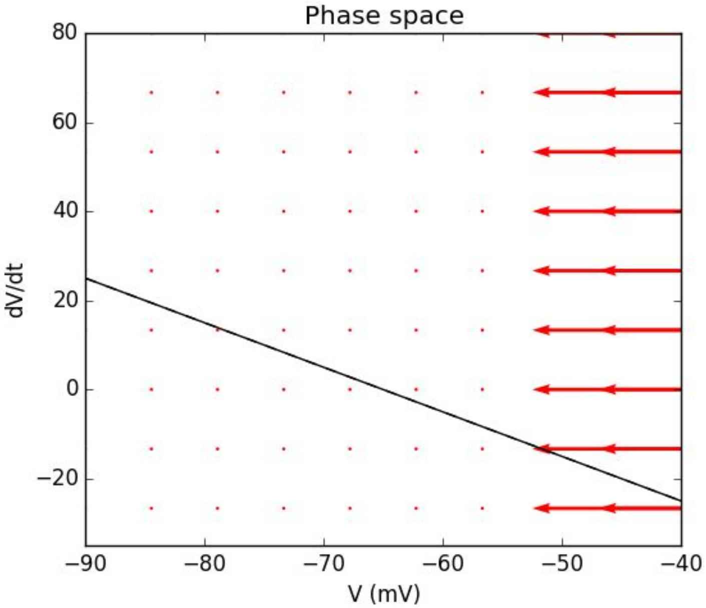


Phase portrait - a better way to see the dynamic



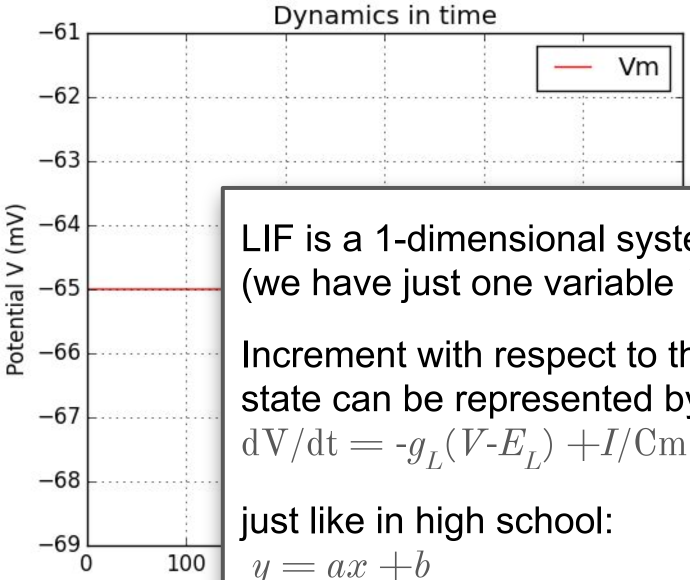
Phase portrait: understanding the system

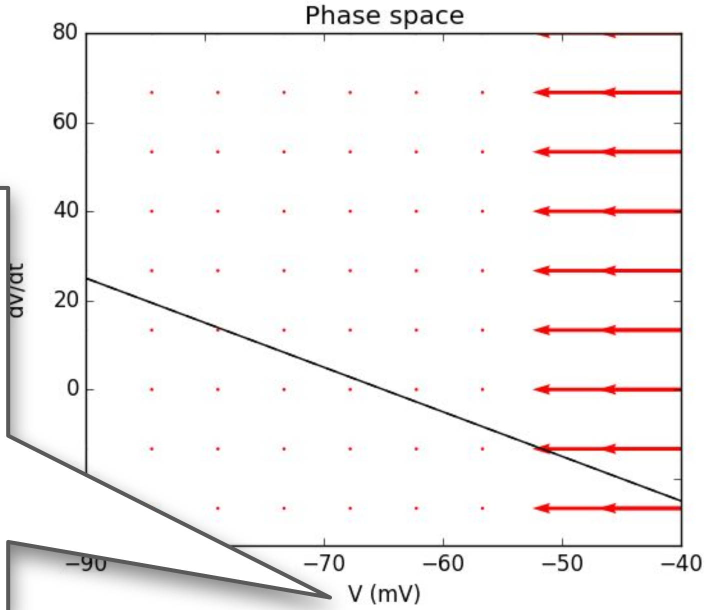




If we set $I = 0$, nothing happens.

Phase portrait: understanding the system





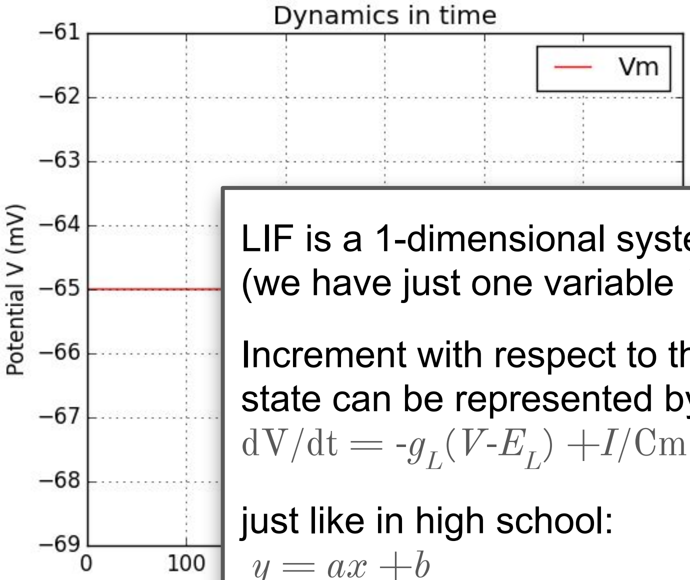
LIF is a 1-dimensional system
(we have just one variable V).

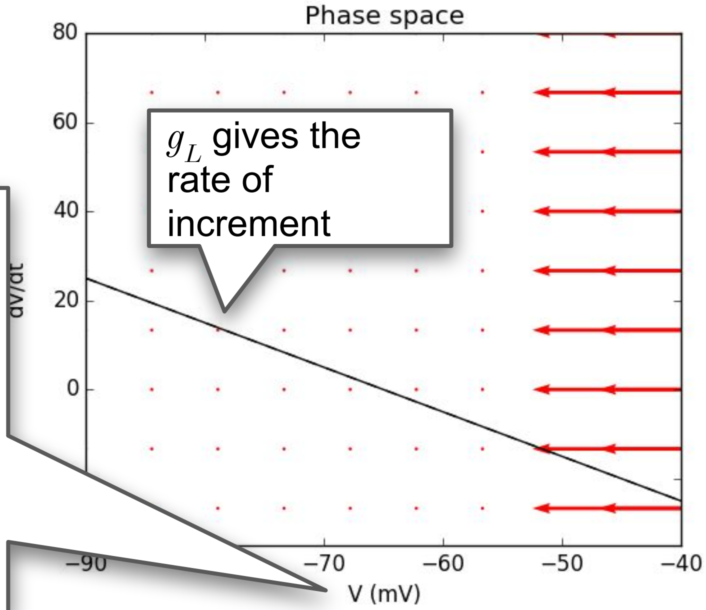
Increment with respect to the current state can be represented by a line:
$\mathrm{dV/dt} = -g_L(V\text{-}E_L) + I/\mathrm{Cm}$

just like in high school:
$y = ax + b$

If we set $I = 0$, nothing happens.

Phase portrait: understanding the system





g_L gives the rate of increment

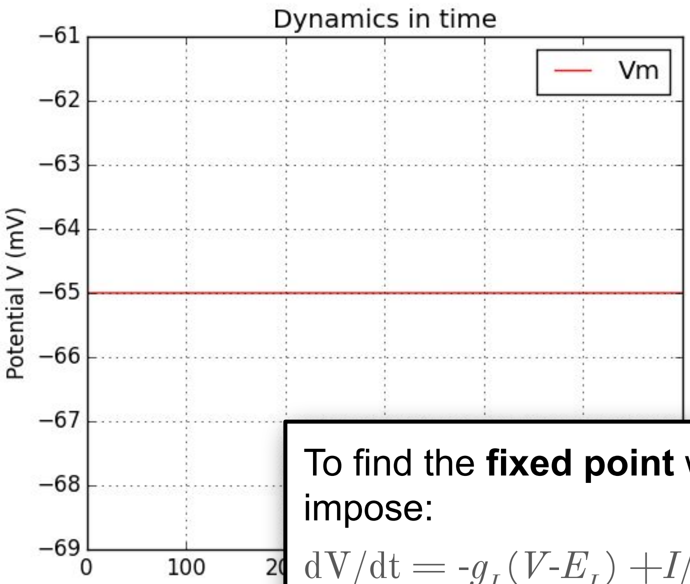
LIF is a 1-dimensional system (we have just one variable V).

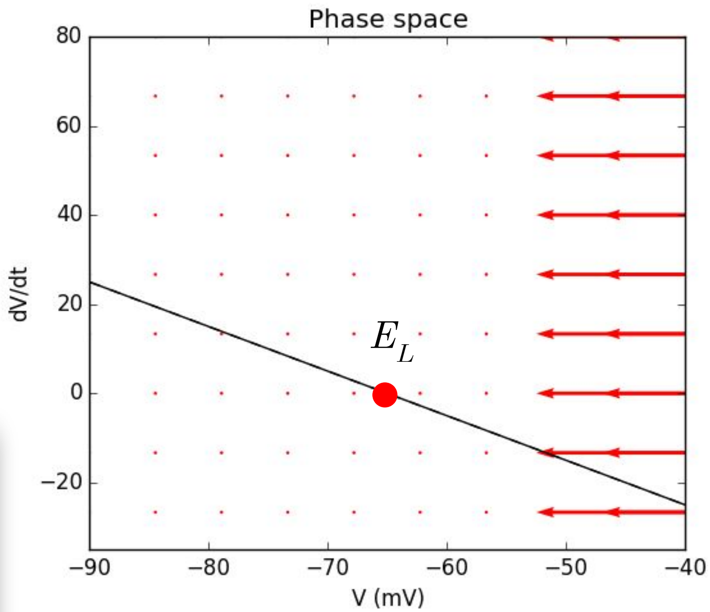
Increment with respect to the current state can be represented by a line:
$\mathrm{dV/dt} = -g_L(V\text{-}E_L) + I/\mathrm{Cm}$

just like in high school:
$y = ax + b$

If we set $I=0$, nothing happens.

Phase portrait: understanding the system





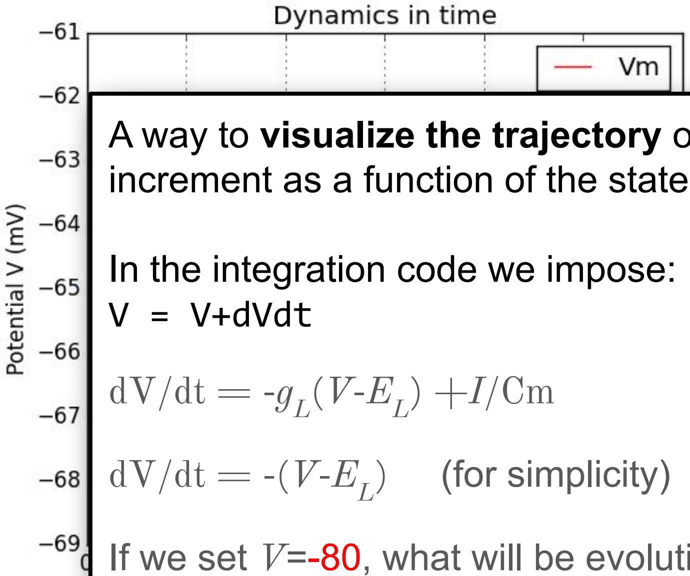
To find the **fixed point** we impose:

$\mathrm{dV/dt} = -g_L(V\text{-}E_L) + I/\mathrm{Cm} = \mathbf{0}$

when $V{=}E_L$ then $\mathrm{dV/dt} = 0$

If we set I

Phase portrait: understanding the system



A way to **visualize the trajectory** of increment as a function of the state of *V*.

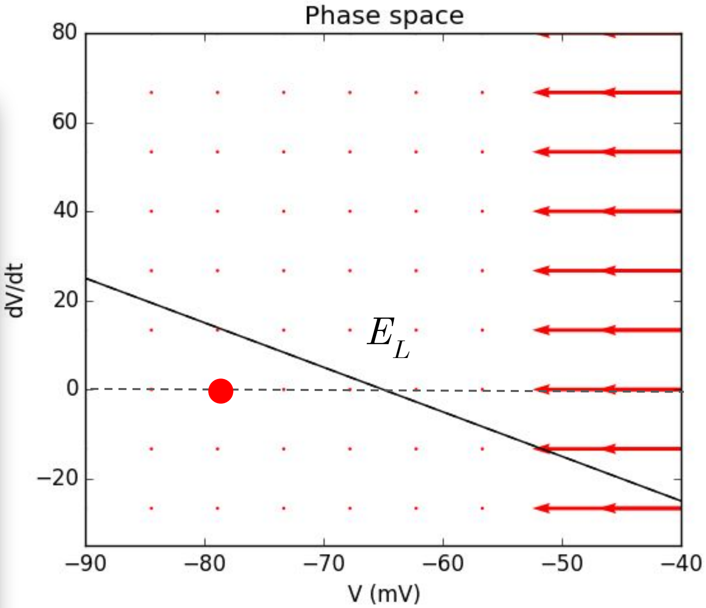
In the integration code we impose:

```
V = V+dVdt
```

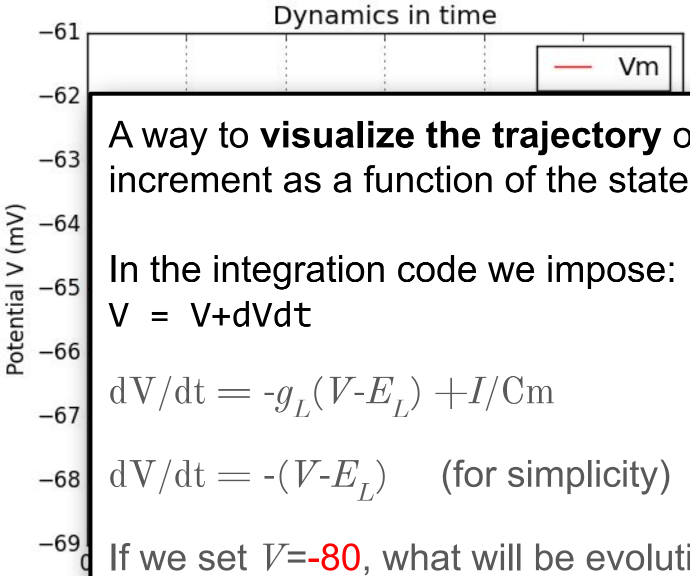
$$\mathrm{dV/dt} = -g_L(V-E_L) + I/\mathrm{Cm}$$

$$\mathrm{dV/dt} = -(V-E_L)$$ (for simplicity)

If we set V=-80, what will be evolution?



Phase portrait: understanding the system



A way to **visualize the trajectory** of increment as a function of the state of *V*.

In the integration code we impose:

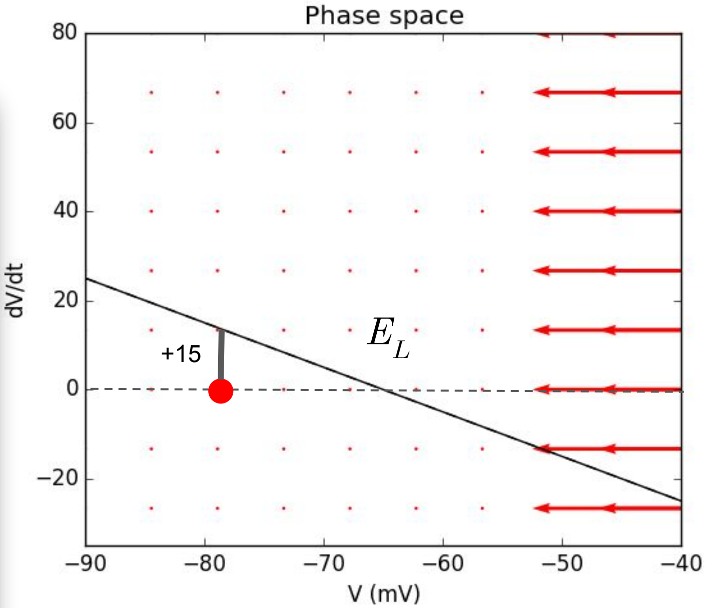
```
V = V+dVdt
```

$\mathrm{dV/dt} = -g_L(V\text{-}E_L) + I/\mathrm{Cm}$

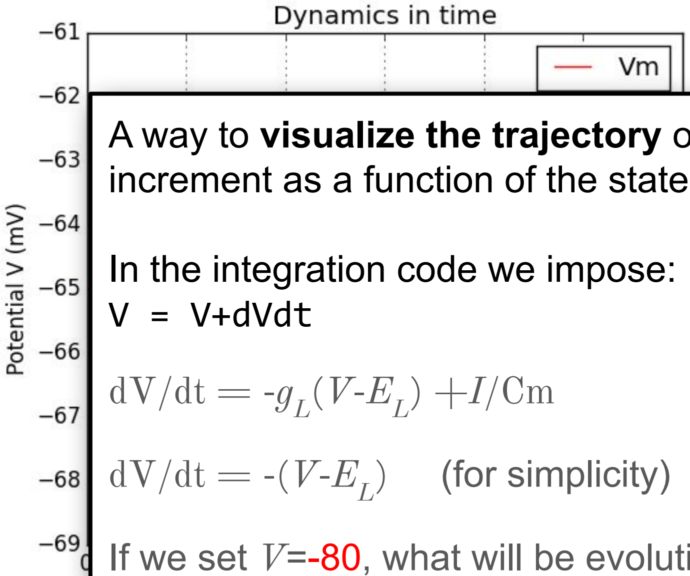
$\mathrm{dV/dt} = -(V\text{-}E_L)$ (for simplicity)

If we set V=-80, what will be evolution?

-(-80 -(-65)) = -(-80 +65) = -(-15) = **+15**



Phase portrait: understanding the system



A way to **visualize the trajectory** of increment as a function of the state of *V*.

In the integration code we impose:

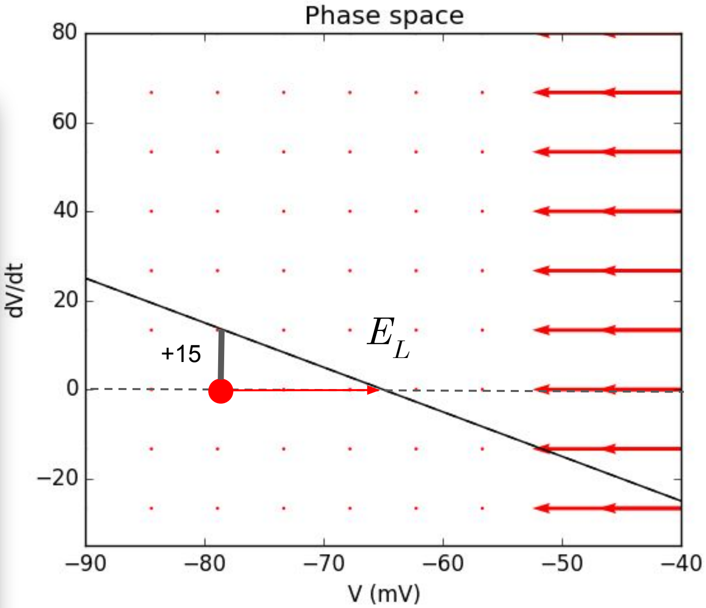
```
V = V+dVdt
```

$\mathrm{dV/dt} = -g_L(V\text{-}E_L) + I/\mathrm{Cm}$

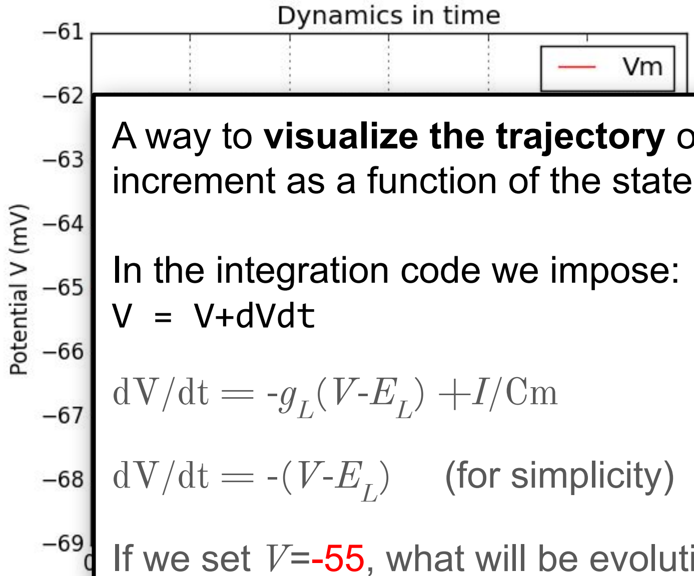
$\mathrm{dV/dt} = -(V\text{-}E_L)$ (for simplicity)

If we set V=-80, what will be evolution?

-(-80 -(-65)) = -(-80 +65) = -(-15) = **+15**



Phase portrait: understanding the system



A way to **visualize the trajectory** of increment as a function of the state of *V*.

In the integration code we impose:

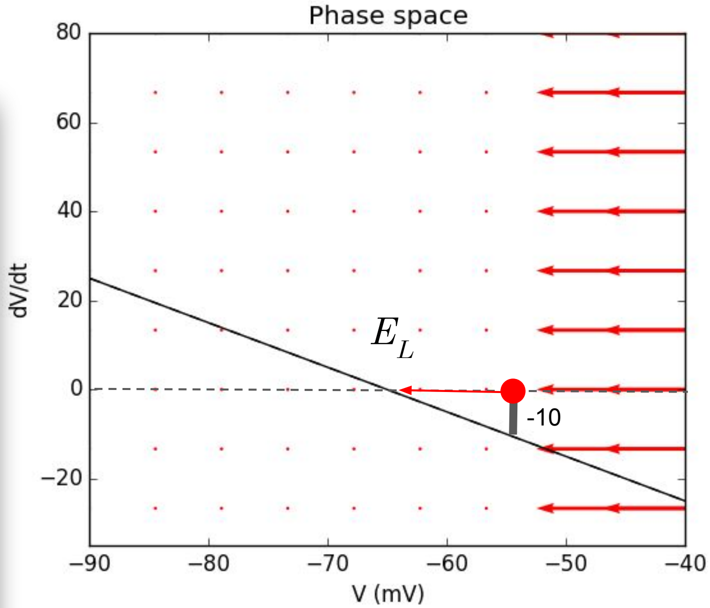
```
V = V+dVdt
```

$\mathrm{dV/dt} = -g_L(V\text{-}E_L) + I/\mathrm{Cm}$

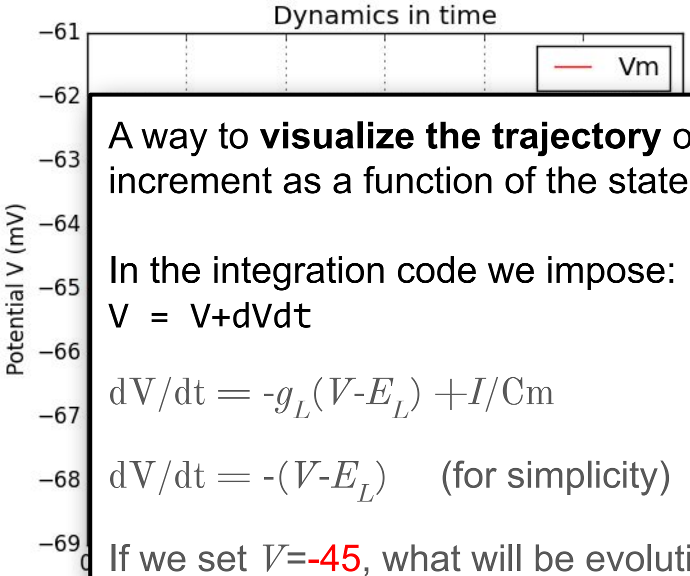
$\mathrm{dV/dt} = -(V\text{-}E_L)$ (for simplicity)

If we set V=-55, what will be evolution?

-(-55 -(-65)) = -(-55 +65) = -(+10) = **-10**



Phase portrait: understanding the system



A way to **visualize the trajectory** of increment as a function of the state of *V*.

In the integration code we impose:

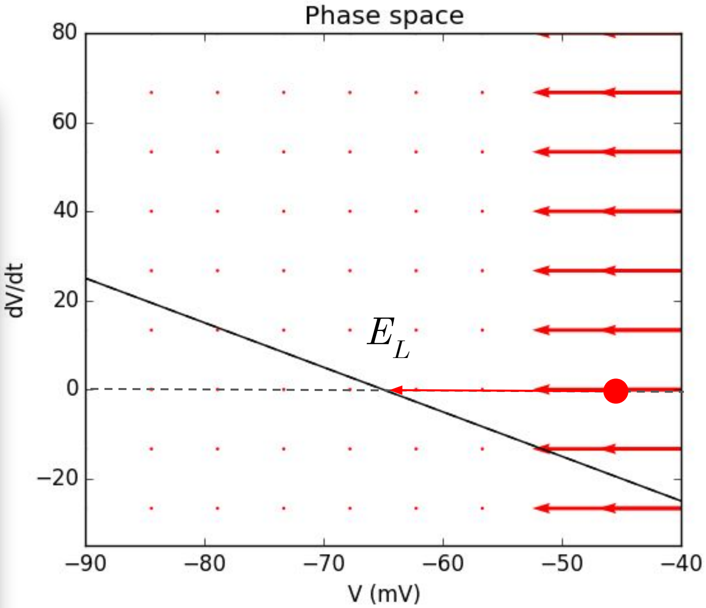
```
V = V+dVdt
```

$\mathrm{dV/dt} = -g_L(V\text{-}E_L) + I/\mathrm{Cm}$

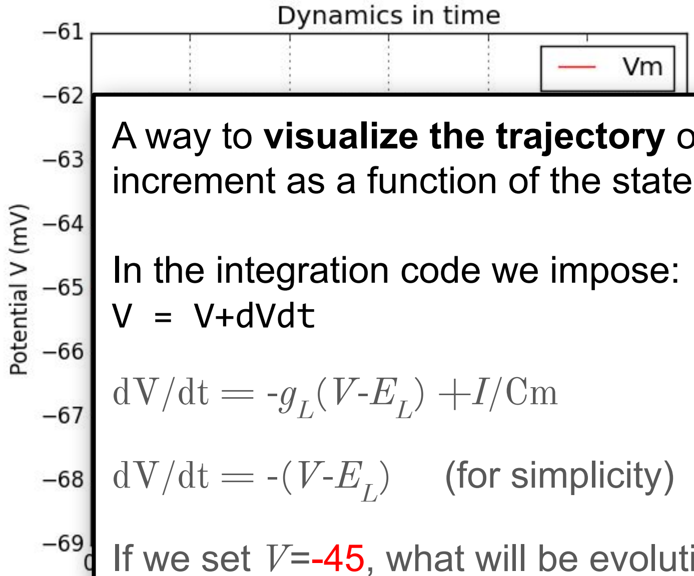
$\mathrm{dV/dt} = -(V\text{-}E_L)$ (for simplicity)

If we set V=-45, what will be evolution?

-(-45 -(-65)) = -(-45 +65) = -(+20) = **-20**



Phase portrait: understanding the system



A way to **visualize the trajectory** of increment as a function of the state of *V*.

In the integration code we impose:

```
V = V+dVdt
```

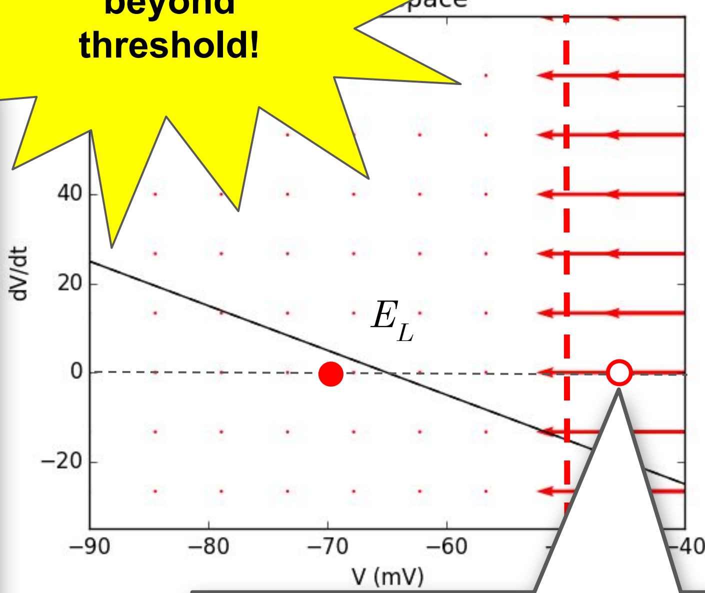
$\mathrm{dV/dt} = -g_L(V\text{-}E_L) + I/\mathrm{Cm}$

$\mathrm{dV/dt} = -(V\text{-}E_L)$ (for simplicity)

If we set V=-45, what will be evolution?

-(-45 -(-65)) = -(-45 +65) = -(+20) = **-20**

beyond threshold!



$\text{if } V > V_{thresh} \rightarrow V = V_{reset}$

$V_{reset} = -70mV$

Hands on the code !!!

1. Open a terminal

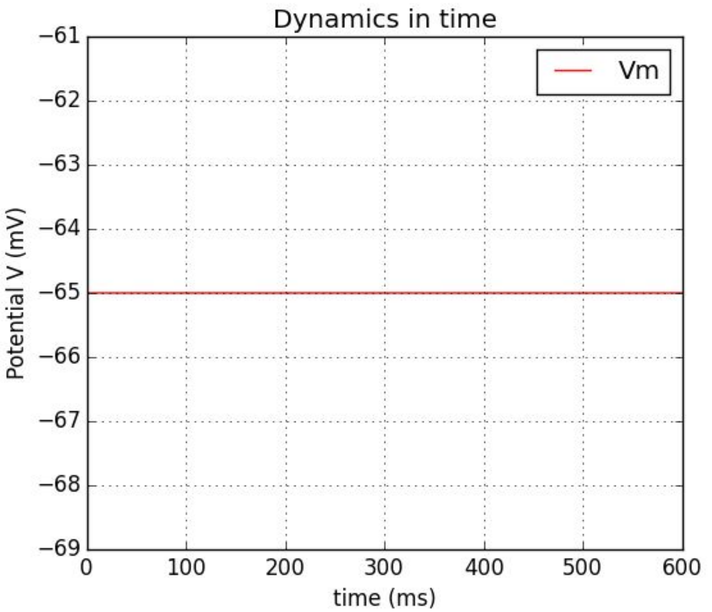
2. type:

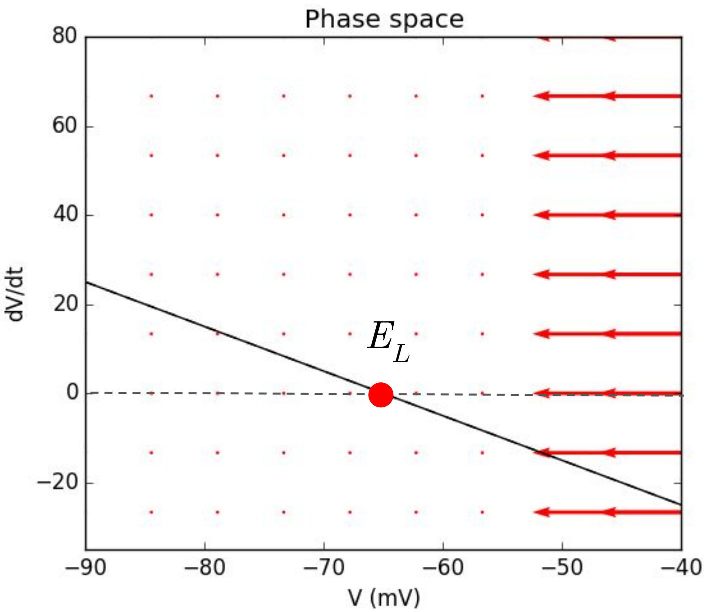
```
cd Desktop/EUGLOH
python3 Portrait_LIF.py
```

3. look into the folder `EUGLOH` for the file `DynamicalAnalysis_LIF.png`

4. Let’s edit the file `Portrait_LIF.py`

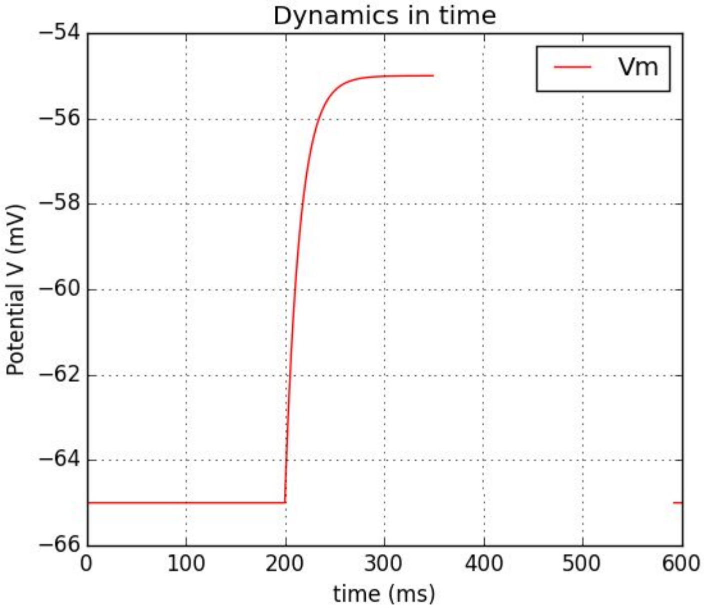
Phase portrait: Integrate-and-Fire

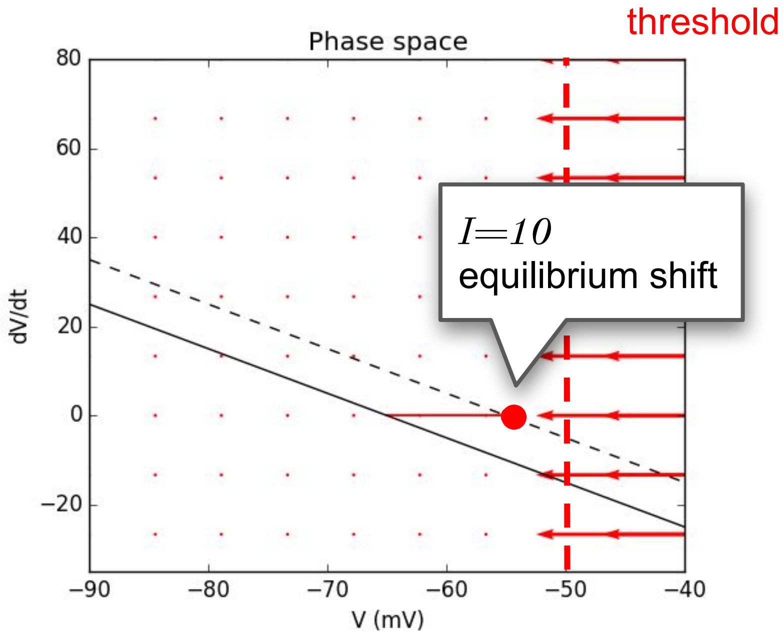




If we set $I = 0$, nothing happens.

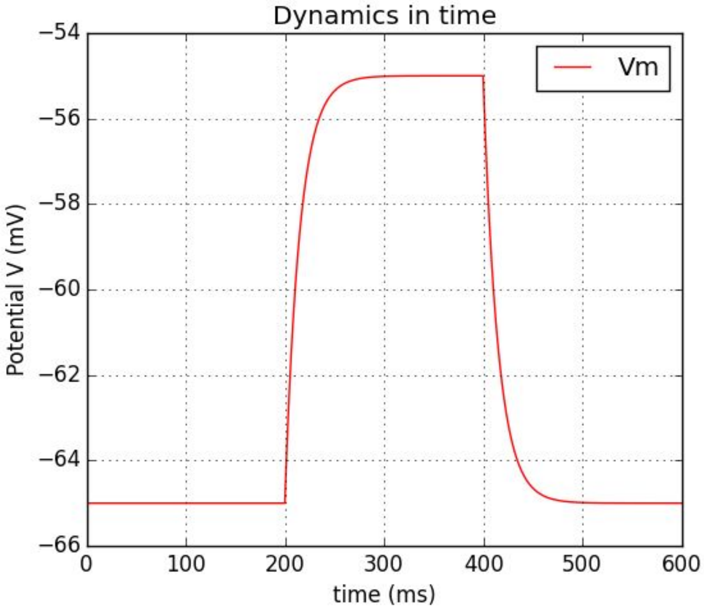
Phase portrait: Integrate-and-Fire

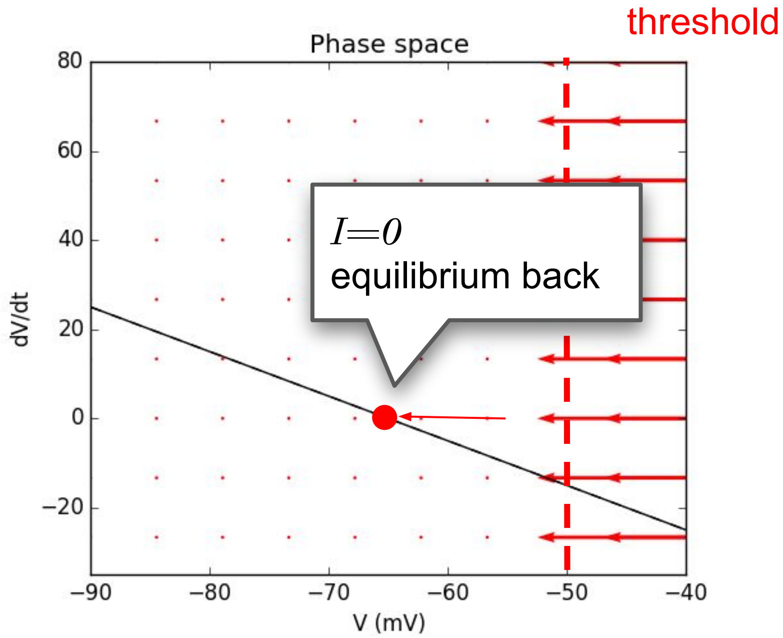




We set $I = 10$ (depolarizing current).

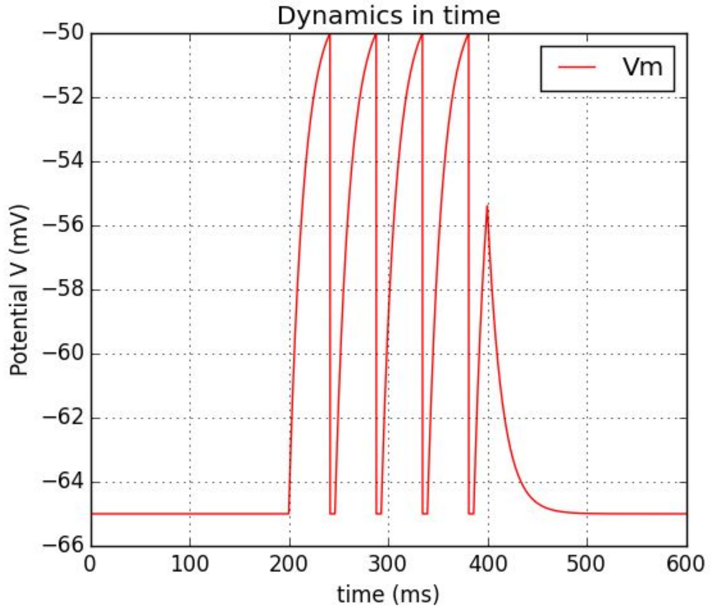
Phase portrait: Integrate-and-Fire

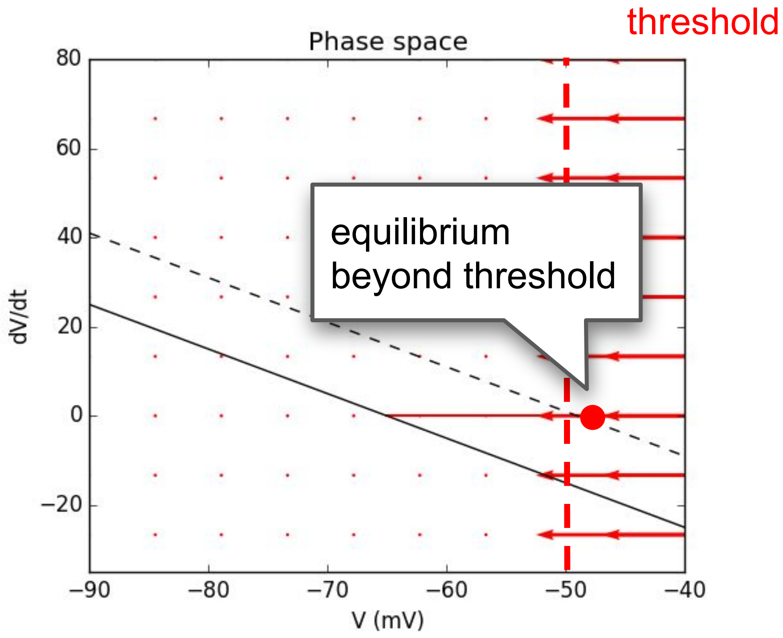




When we reset $I=0$

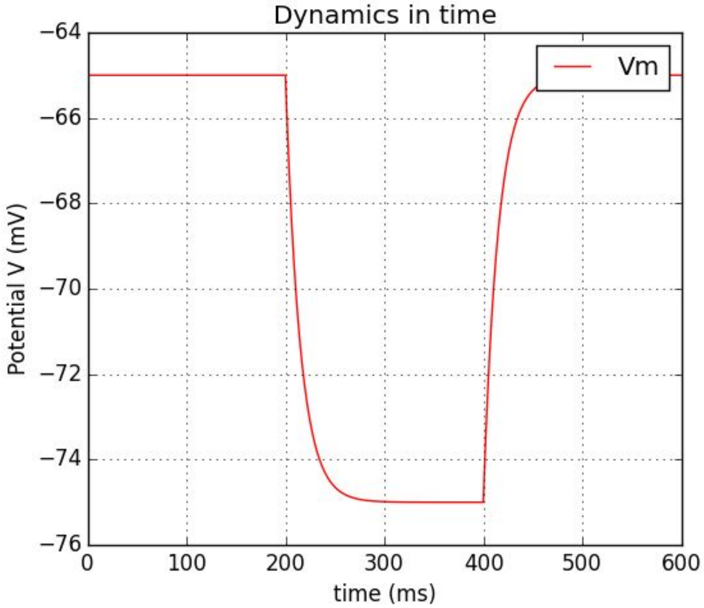
Phase portrait: Integrate-and-Fire

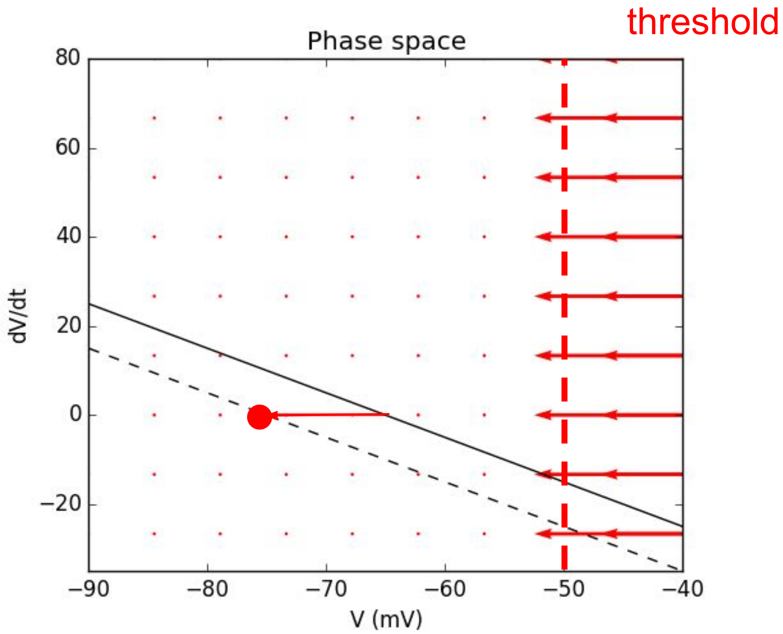




We set $I=16$

Phase portrait: Integrate-and-Fire





I negative = hyperpolarizing current

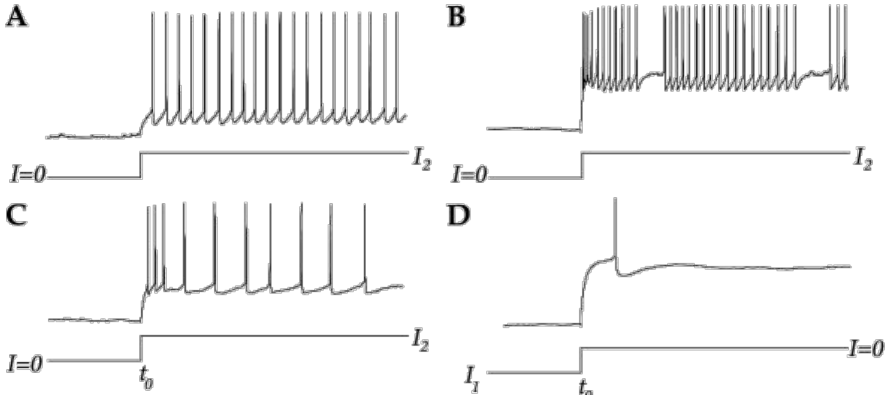
Integrate-and-Fire summary

The IF captures the neuron ability to add up (integrate) inputs.
But that's pretty much all it can do…

The membrane potential of real neurons has a richer dynamic than the Integrate-and-Fire.

Different neurons, in different conditions (neuromodulation) give raise to

- Adaptation (A vs C)
- Bursting (B)
- Inhibitory rebound (D)
- …



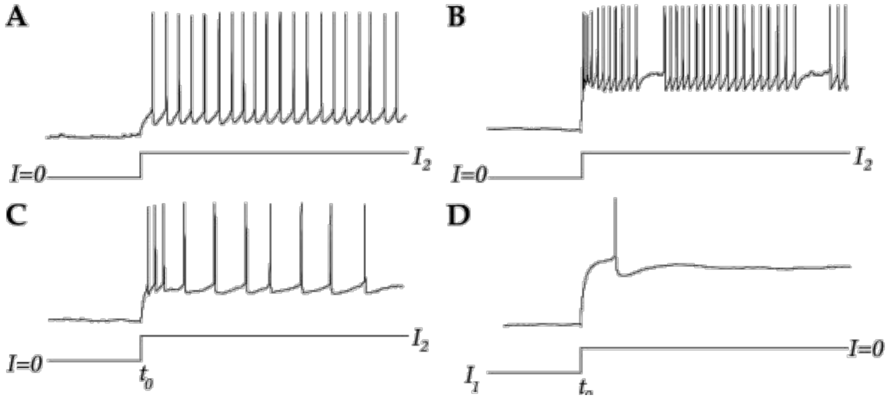
Integrate-and-Fire summary

The IF captures the neuron ability to add up (integrate) inputs.
But that's pretty much all it can do…

The membrane potential of real neurons has a richer dynamic than the Integrate-and-Fire.

Different neurons, in different conditions (neuromodulation) give raise to

- Adaptation (A vs C)
- Bursting (B)
- Inhibitory rebound (D)
- …

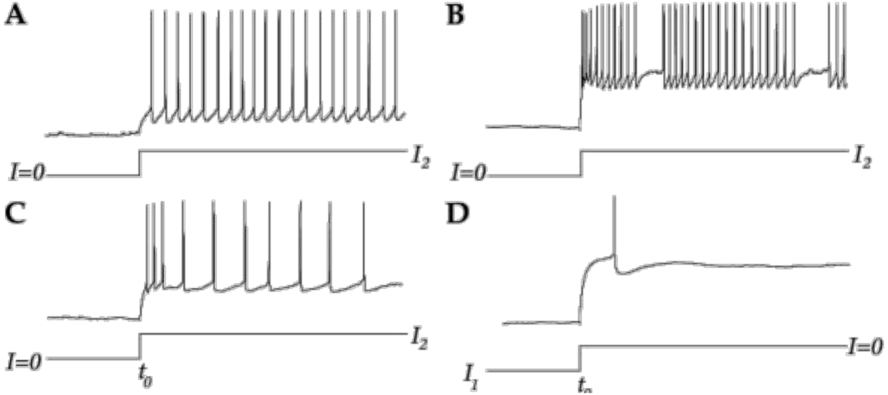


!! We have a better model !!

Adaptive Exponential Integrate-and-Fire

The LIF does not capture

- Rise of potential beyond threshold
- Adaptation
- Bursting
- Inhibitory rebound

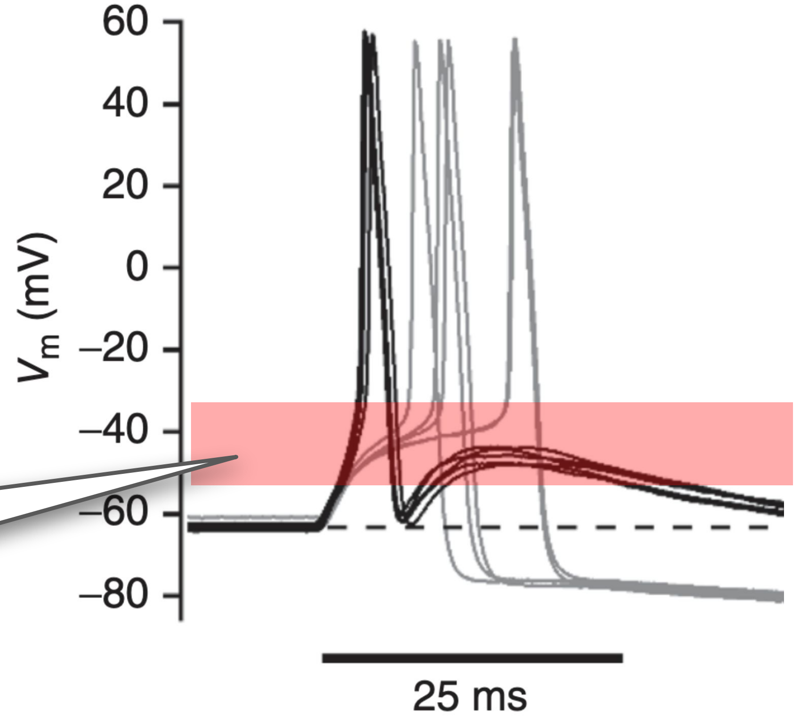


$$dV/dt = -g_L(V - E_L) + I \ /\mathrm{Cm}$$

Adaptive Exponential Integrate-and-Fire

The LIF does not capture

- **Rise of potential beyond threshold**
- Adaptation
- Bursting
- Inhibitory rebound



Threshold is not *fixed* neither *thin*.
But when the spike reises,
it does so **exponentially**.

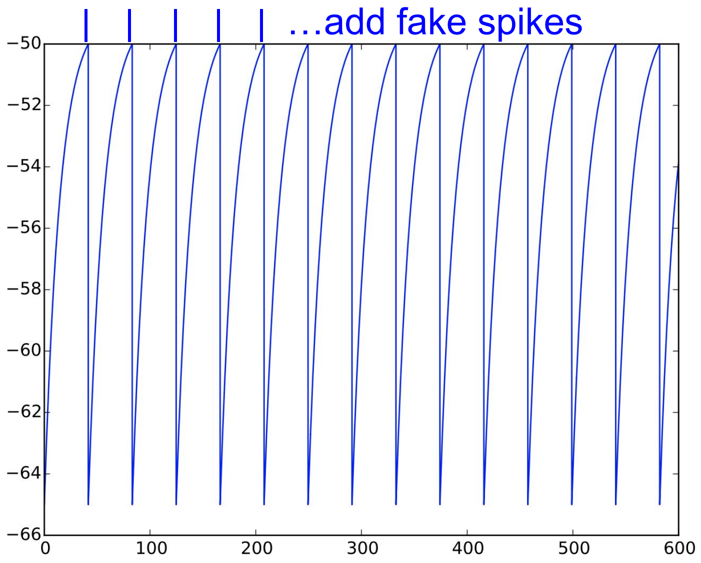
$$dV/dt = -g_L(V-E_L) + I \;/\mathrm{Cm}$$

Adaptive Exponential Integrate-and-Fire

The LIF does not capture

- **Rise of potential beyond threshold**
- Adaptation
- Bursting
- Inhibitory rebound

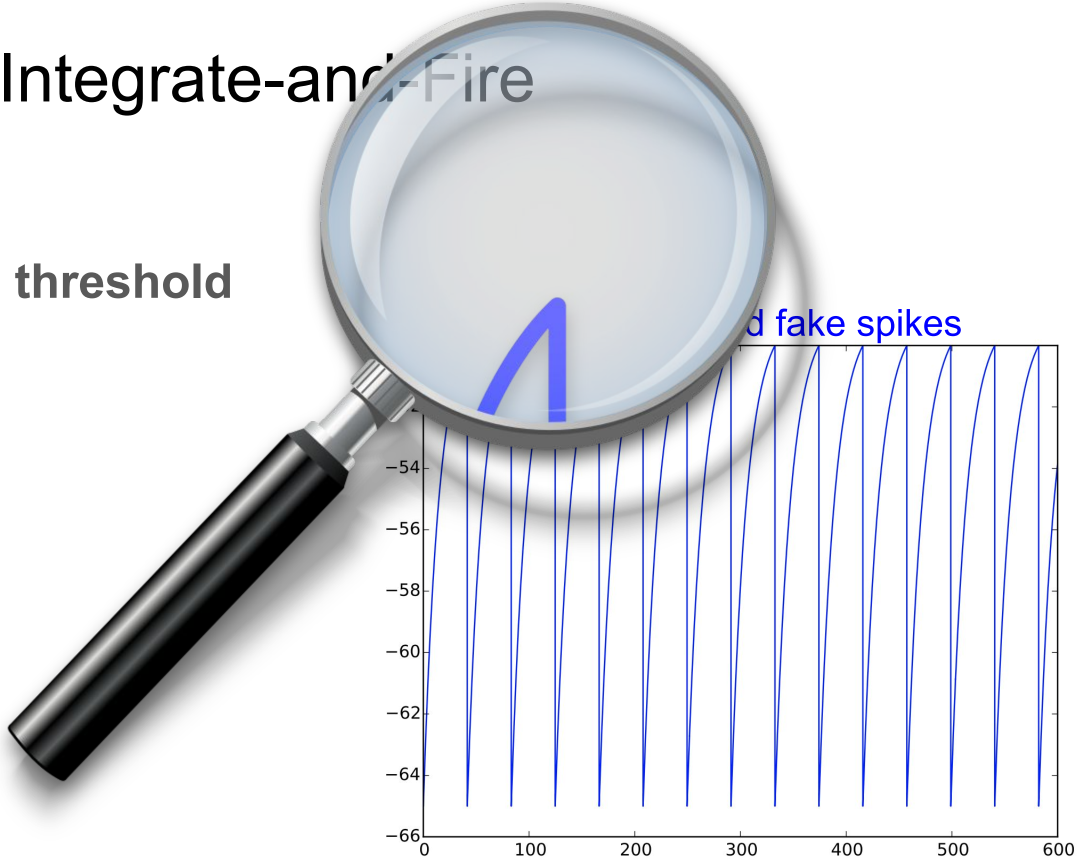
$$dV/dt = -g_L(V-E_L) + I \ /\mathrm{Cm}$$



Adaptive Exponential Integrate-and-Fire

The LIF does not capture

- **Rise of potential beyond threshold**
- Adaptation
- Bursting
- Inhibitory rebound



$$dV/dt = -g_L(V-E_L) + I \ /\mathrm{Cm}$$

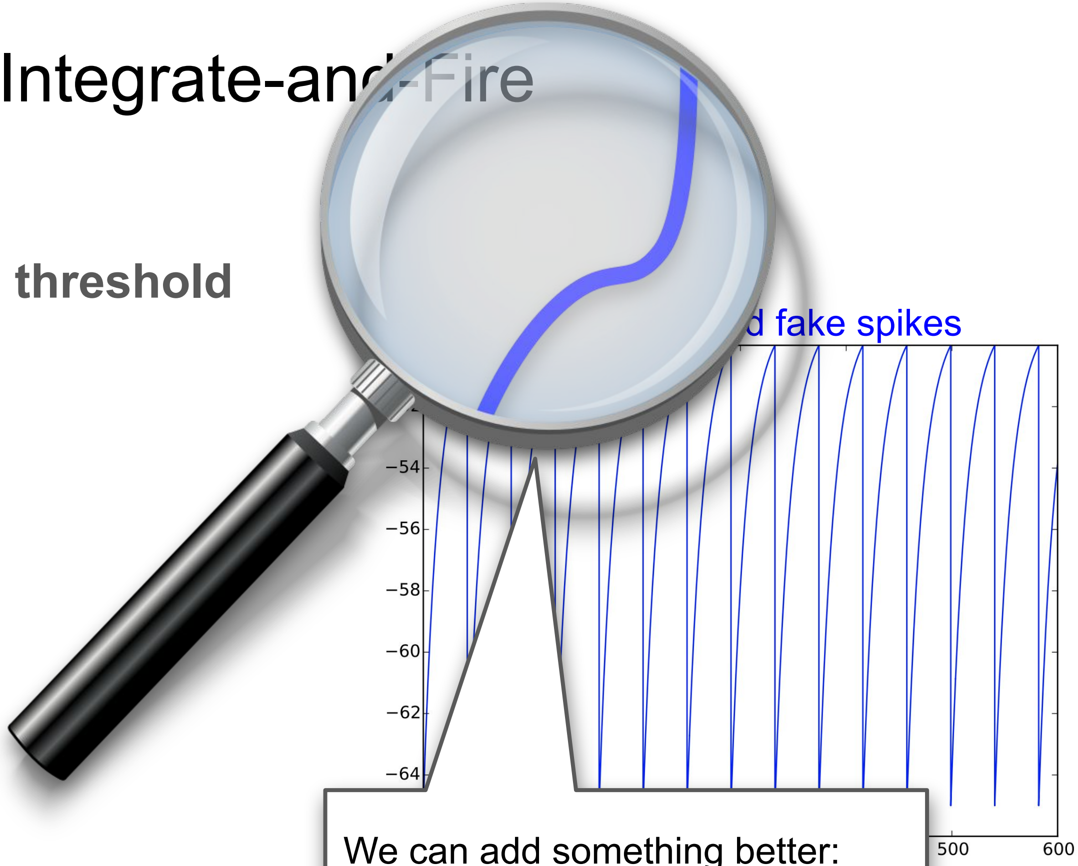
Adaptive Exponential Integrate-and-Fire

The LIF does not capture

- **Rise of potential beyond threshold**
- Adaptation
- Bursting
- Inhibitory rebound

$$+ g_L \Delta_T \exp\left(\frac{V - V_T}{\Delta_T}\right)$$

$$\mathrm{d}V/\mathrm{dt} = -g_L(V\text{-}E_L) + I \ /\mathrm{Cm}$$



We can add something better:
an **exponential** function.

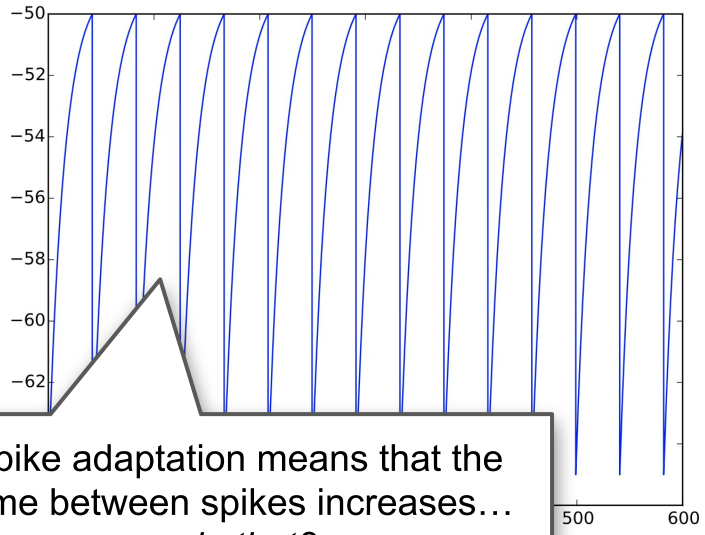
Adaptive Exponential Integrate-and-Fire

The LIF does not capture

- Rise of potential beyond threshold
- **Adaptation**
- Bursting
- Inhibitory rebound

$$\boxed{+ g_L \Delta_T \exp\left(\frac{V - V_T}{\Delta_T}\right)}$$

$$\mathrm{d}V/\mathrm{dt} = -g_L(V\text{-}E_L) + I \;\; /\mathrm{Cm}$$



Spike adaptation means that the time between spikes increases…
How can we do that?

Adaptive Exponential Integrate-and-Fir

We can act on V!
If we lower it by a fixed amount for each spike fired, V will take slightly more to reach threshold

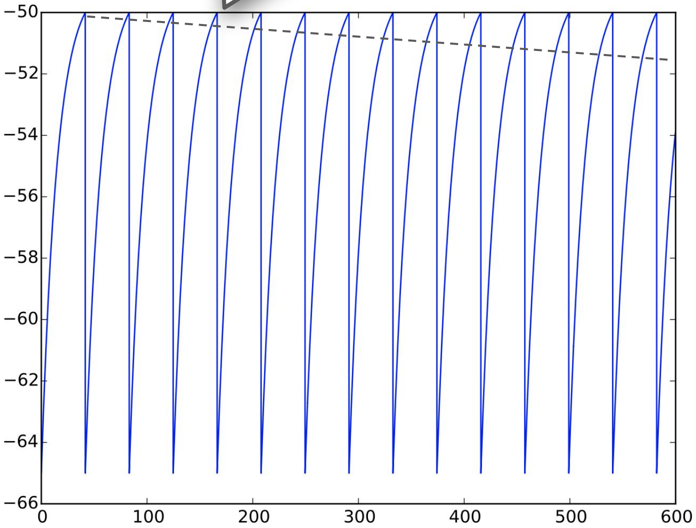
The LIF does not capture

- Rise of potential beyond threshold
- **Adaptation**
- Bursting
- Inhibitory rebound

$$\boxed{+\, g_L \Delta_T \exp\left(\frac{V - V_T}{\Delta_T}\right)}$$

$$\mathrm{d}V/\mathrm{dt} = -g_L(V\text{-}E_L) + I \;-\; \mathbf{w} \;/\mathrm{Cm}$$

$$\boxed{\text{if } V > 0\text{ mV} \quad \text{then} \begin{cases} V \to V_r \\ w \to w_r = w + b. \end{cases}}$$



Adaptive Exponential Integrate-and-Fire

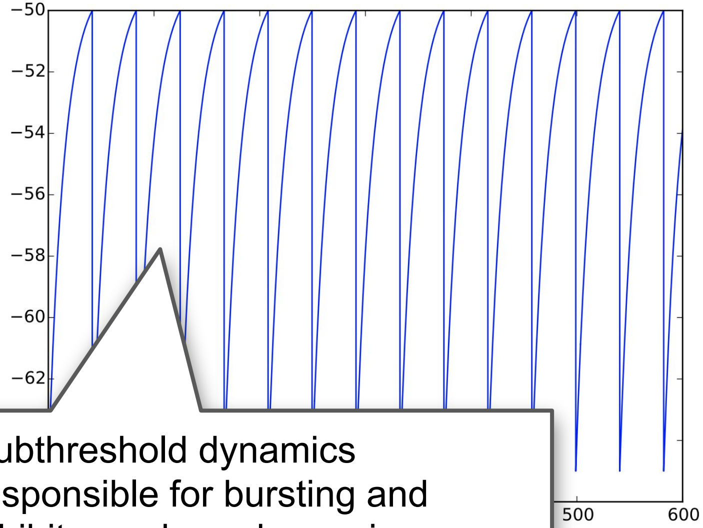
The LIF does not capture

- Rise of potential beyond threshold
- Adaptation
- **Bursting**
- **Inhibitory rebound**

$$\boxed{+ g_L \Delta_T \exp\left(\frac{V - V_T}{\Delta_T}\right)}$$

$$\mathrm{d}V/\mathrm{dt} = -g_L(V\text{-}E_L) + I \; - \mathbf{w} \; / \mathrm{Cm}$$

$$d\mathbf{w}/dt = \mathrm{a}(V - E_L) - w \; / \; \tau_w$$



Subthreshold dynamics responsible for bursting and inhibitory rebounds require a second variable evolving in time.

Adaptive Exponential Integrate-and-Fire

$$C\frac{dV}{dt} = -g_L(V - E_L) + g_L \Delta_T \exp\left(\frac{V - V_T}{\Delta_T}\right) + I - w, \quad (1)$$

$$\tau_w \frac{dw}{dt} = a(V - E_L) - w. \quad (2)$$

$$\text{if } V > 0 \text{ mV} \quad \text{then} \begin{cases} V \rightarrow V_r \\ w \rightarrow w_r = w + b. \end{cases} \quad (3)$$

Phase portrait - a better way to see the dynamic

Now the system is changed… we have two variables V and w

But the approach is the same:

- **fixed point** where the $dV/dt = 0$ and $dw/dt = 0$
- **stability** of fixed points (they can be stable or unstable)
- **nullclines** are the sets of points where $\mathrm{f}(V) = 0$ and $\mathrm{g}(w) = 0$

Their calculations are now a more long and complex in this system.

Let’s have an intuitive idea first …

Adaptive Exponential Integrate-and-Fire

$$C\frac{dV}{dt} = -g_L(V - E_L) + g_L \Delta_T \exp\left(\frac{V - V_T}{\Delta_T}\right) + I - w, \tag{1}$$

$$\tau_w \frac{dw}{dt} = a(V - E_L) - w. \tag{2}$$

$$\text{if } V > 0 \text{ mV} \quad \text{then} \begin{cases} V \to V_r \\ w \to w_r = w + b. \end{cases} \tag{3}$$

Nullclines

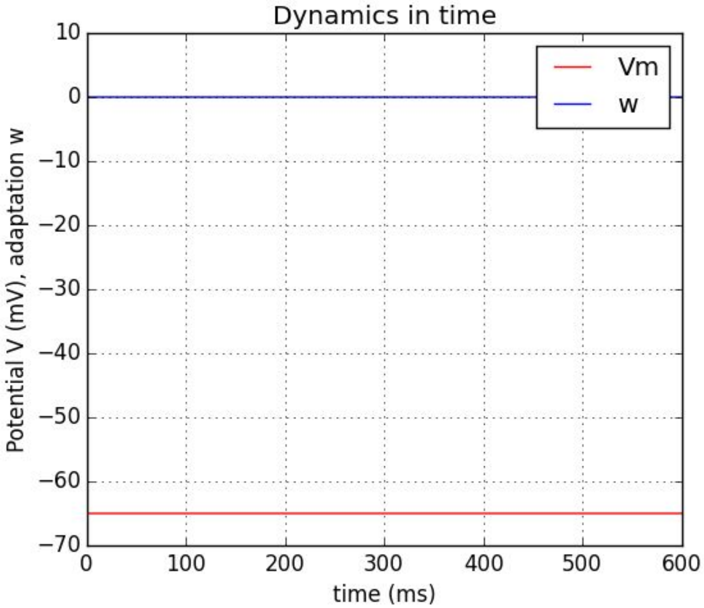
$$\boldsymbol{w} = -g_L(V - E_L) + g_L \Delta_T \exp\left(\frac{V - V_T}{\Delta_T}\right) + I$$

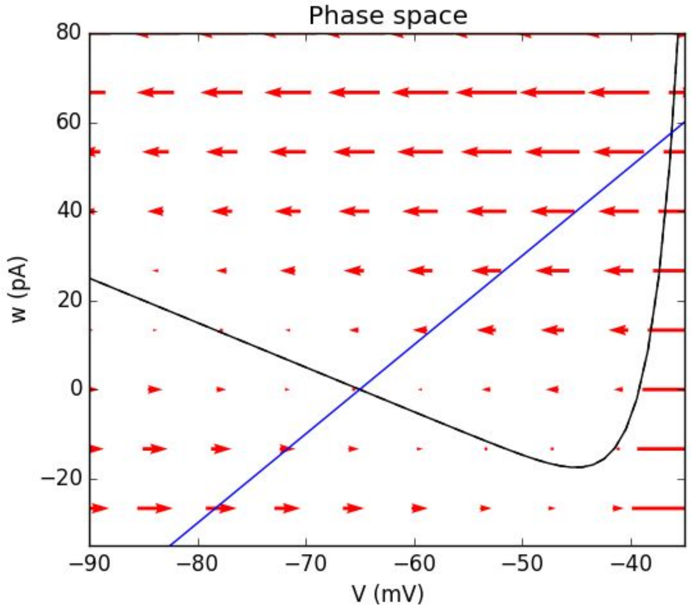
$$\boldsymbol{w} = a(V - E_L)$$

Fixed points: find them graphically

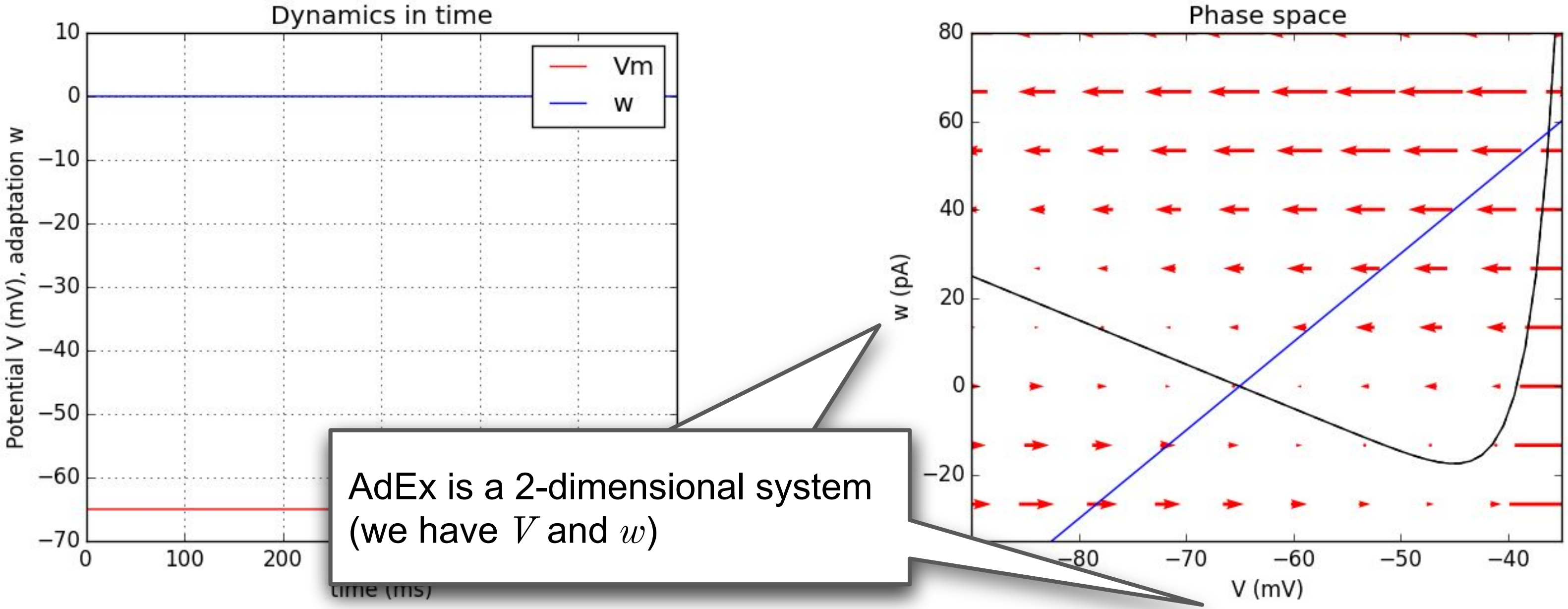
Stability: find it graphically

Phase portrait: understanding the system



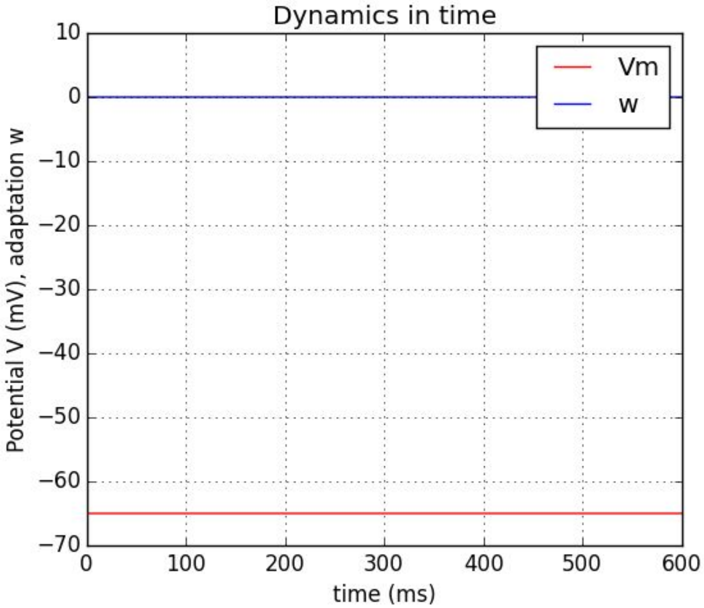


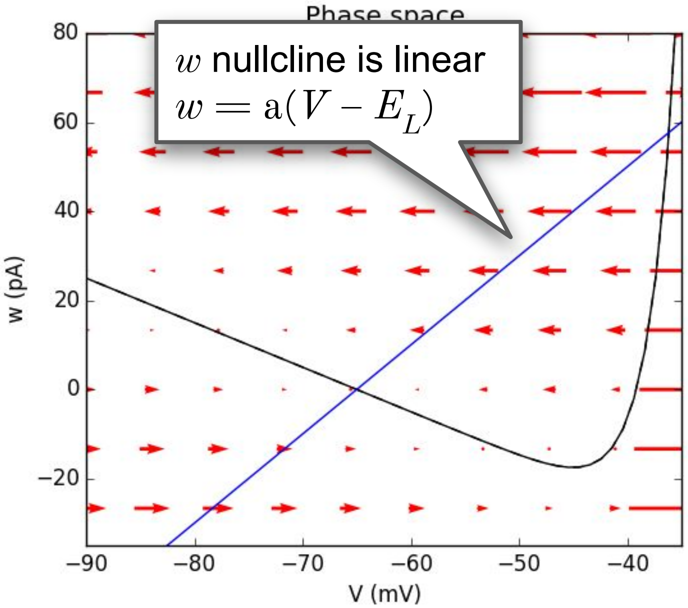
Phase portrait: understanding the system



AdEx is a 2-dimensional system
(we have V and w)

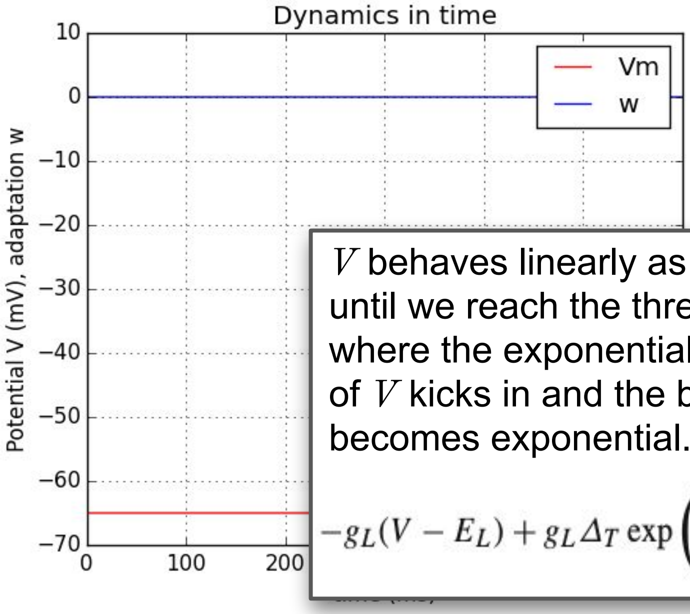
Phase portrait: understanding the system





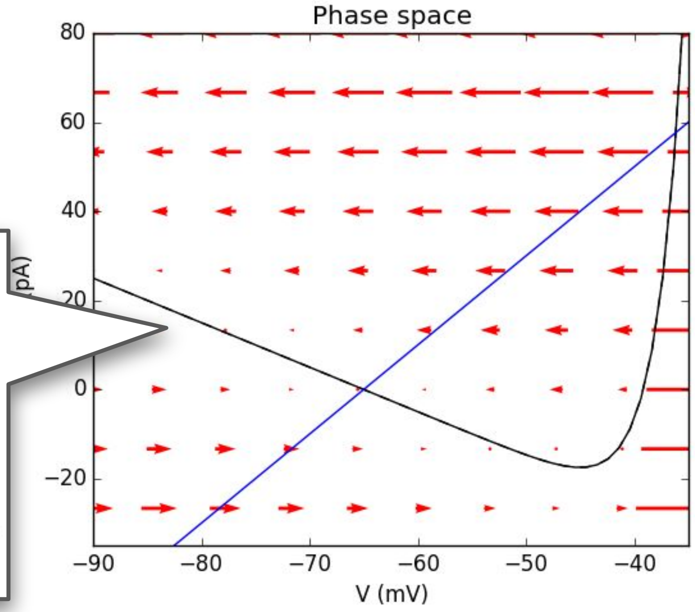
w nullcline is linear
$w = \mathrm{a}(V - E_L)$

Phase portrait: understanding the system

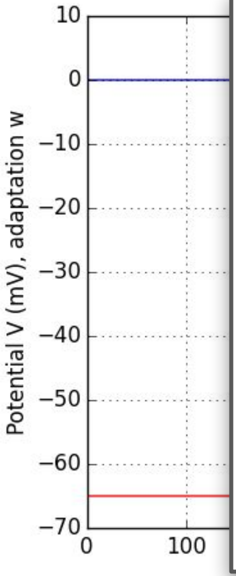


V behaves linearly as in the IF until we reach the threshold, where the exponential evaluation of V kicks in and the behavior becomes exponential.

$$-g_L(V - E_L) + g_L \Delta_T \exp\left(\frac{V - V_T}{\Delta_T}\right) + I$$



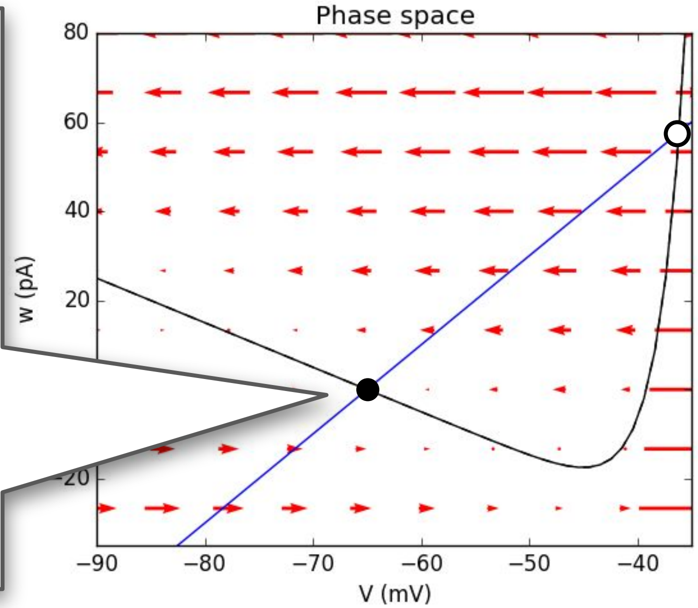
Phase portrait: understanding the system



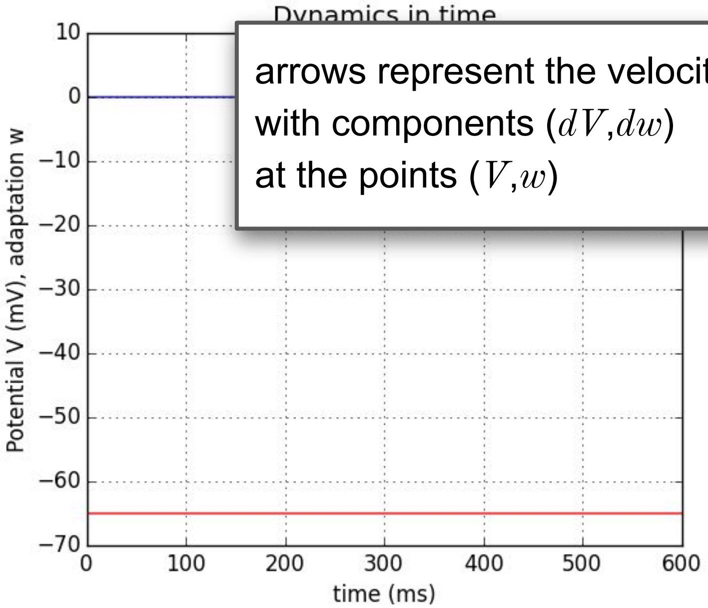
To find the **fixed points** we impose:
$dV/dt = 0$ and $dw/dt = 0$

This is a system of equations:
one of the second order, dV/dt
one of the first order, dw/dt

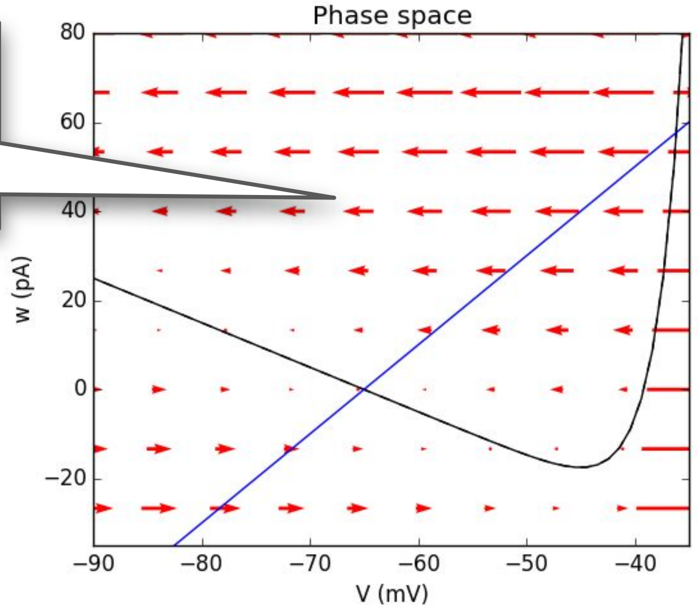
One point is $V = E_L$.
This fixed point is stable.
(the other point is not stable)



Phase portrait: understanding the system



arrows represent the velocity vectors
with components (dV, dw)
at the points (V, w)



Hands on the code !!!

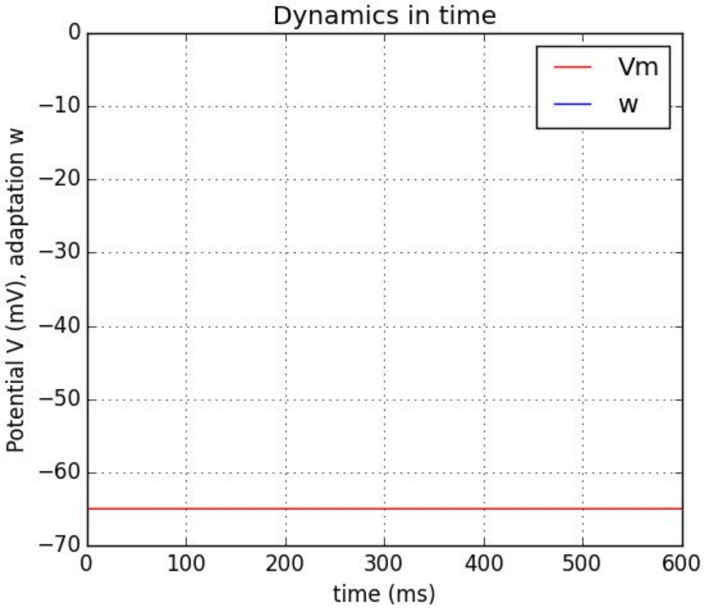
1. Open a terminal

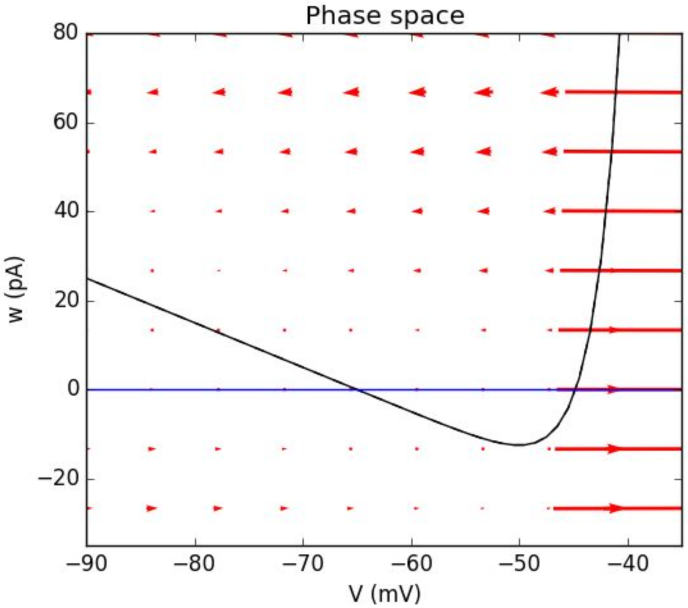
2. type:
```
cd Desktop/EUGLOH
python3 Portrait_AdEx.py
```

3. look into the folder `EUGLOH` for the file `DynamicalAnalysis_AdEx.png`

4. Let's edit the file `Portrait_AdEx.py`

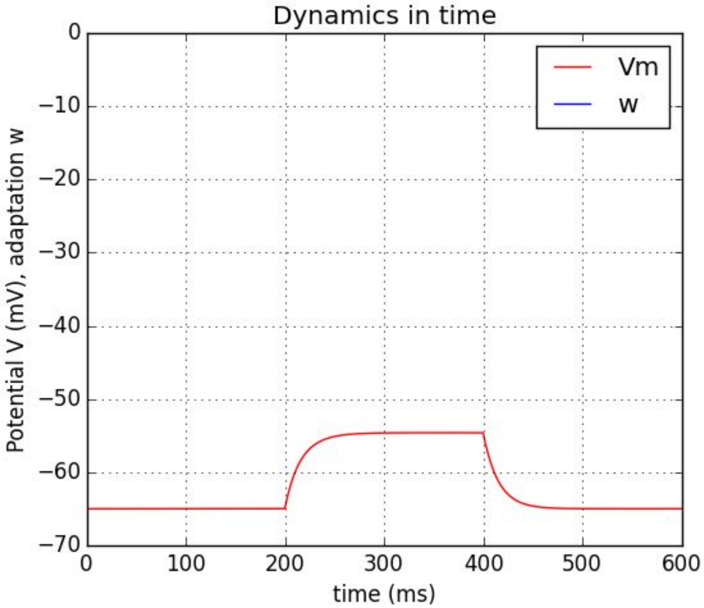
Phase portrait: Regular Spiking cell

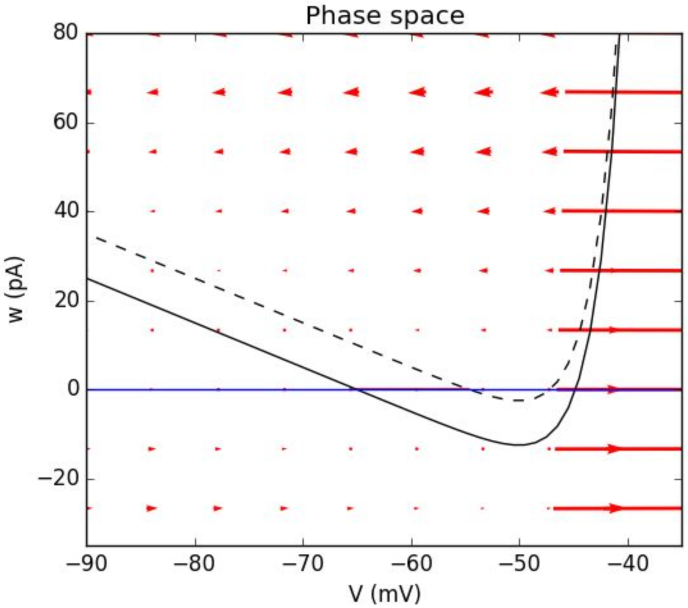




$I = 0$

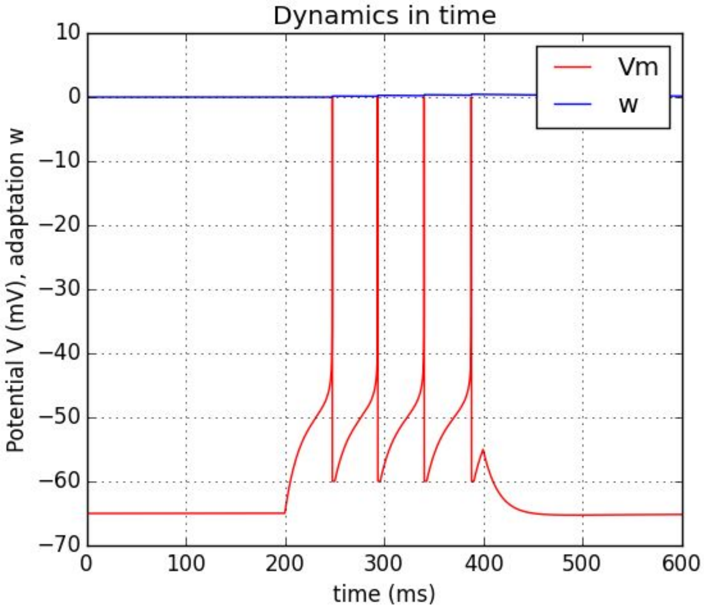
Phase portrait: Regular Spiking cell

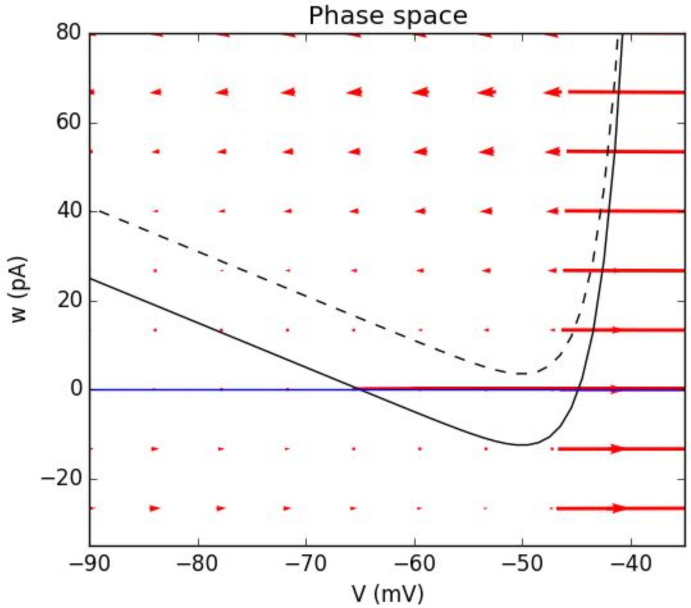




We set $I = 10$ (depolarizing current)

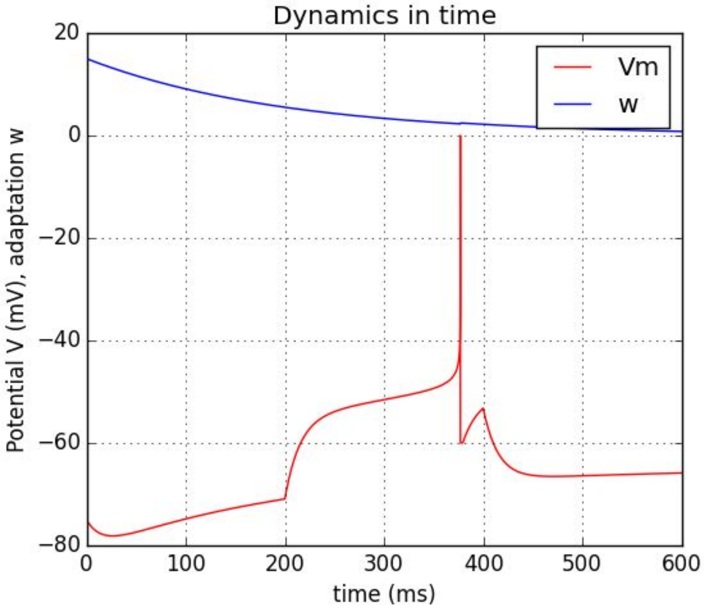
Phase portrait: Regular Spiking cell

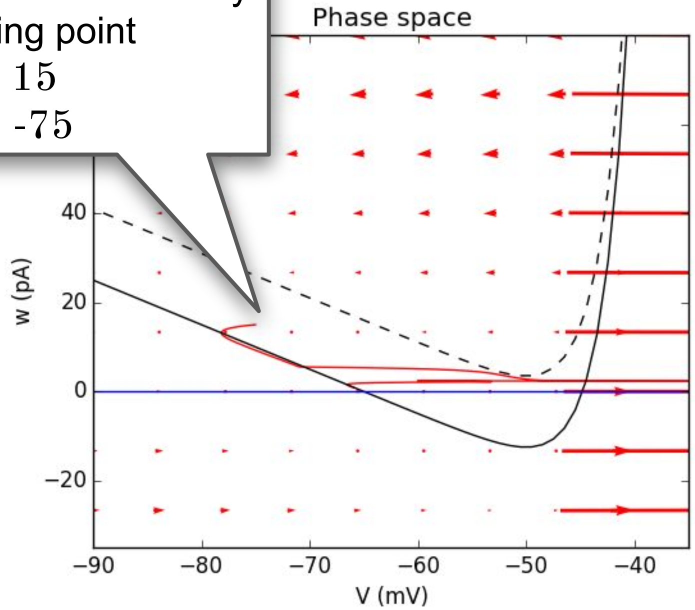




We set $I = 20$ (depolarizing current)

Phase portrait: Regular Spiking cell

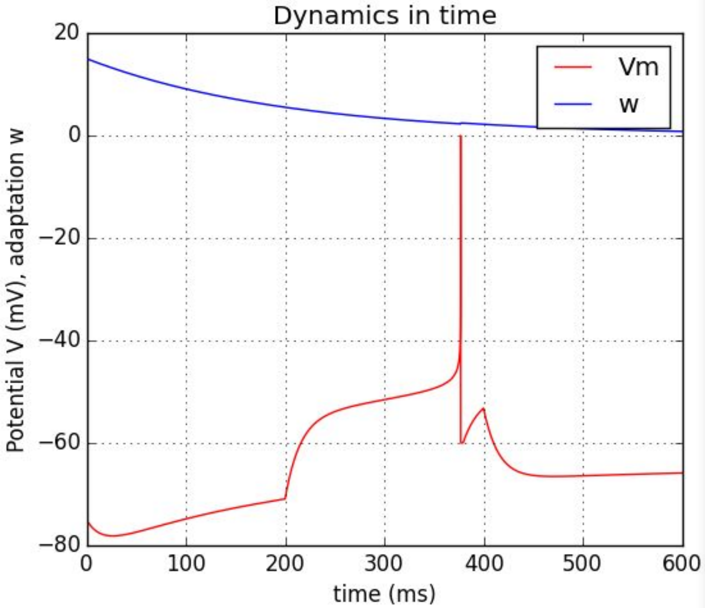


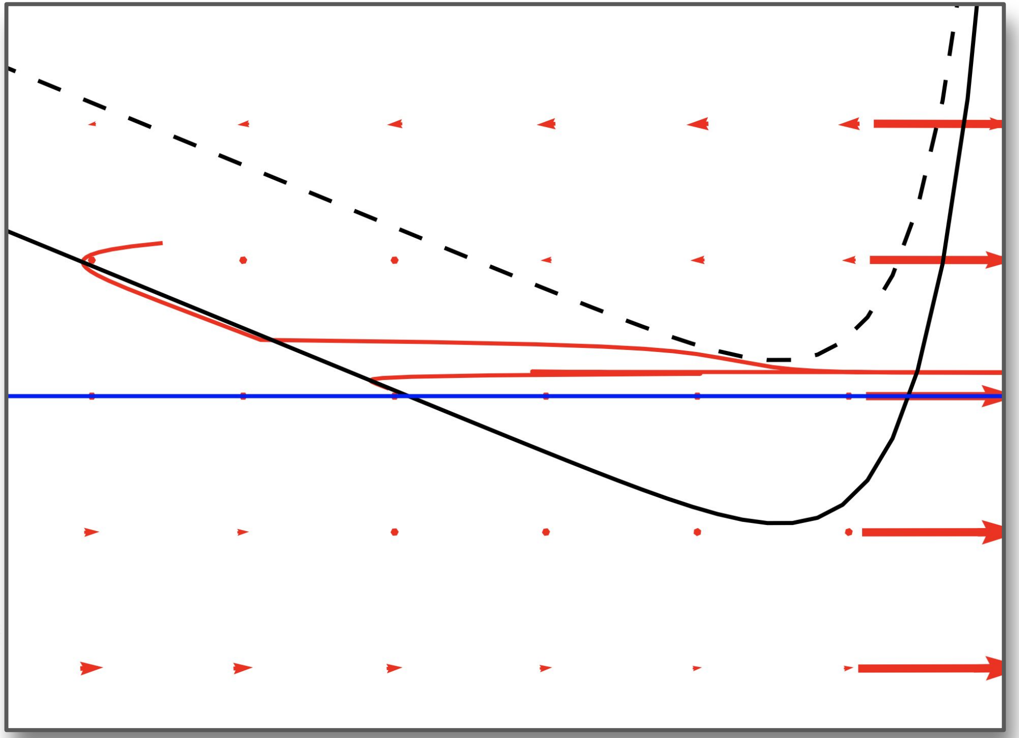


We can choose any starting point
$w = 15$
$V =$ -75

We set $I\!=\!20$ (depolarizing current)

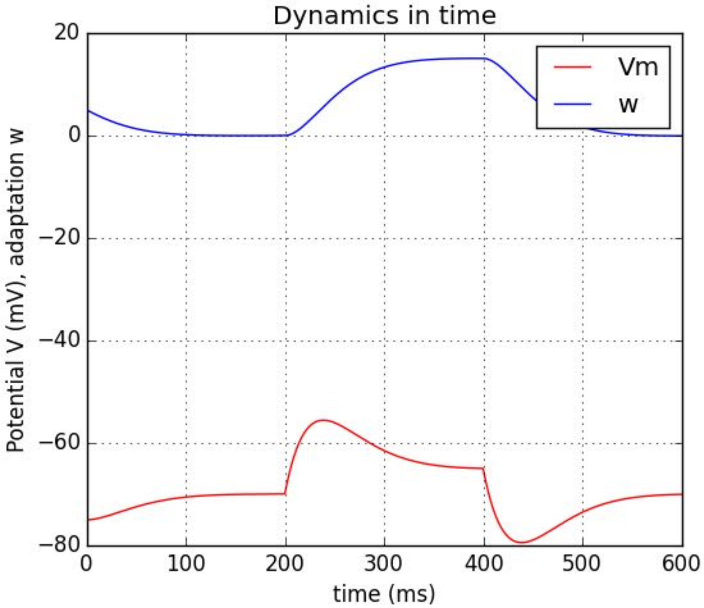
Phase portrait: Regular Spiking cell

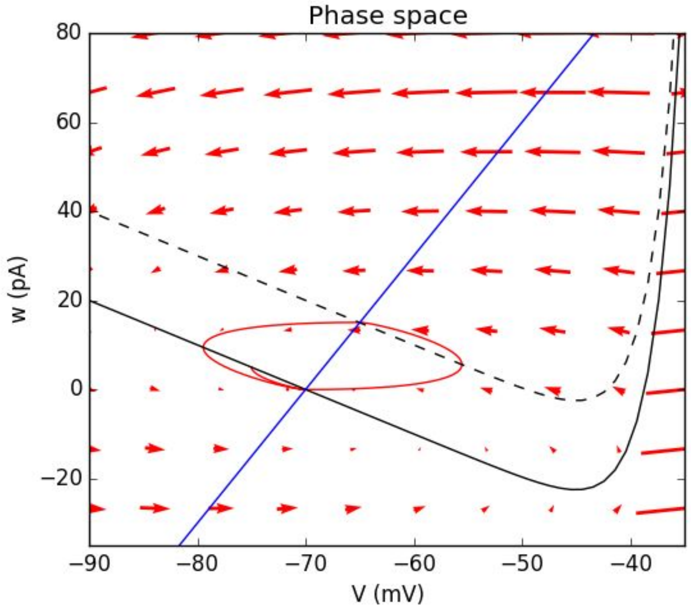




We set $I = 20$ (depolarizing current)

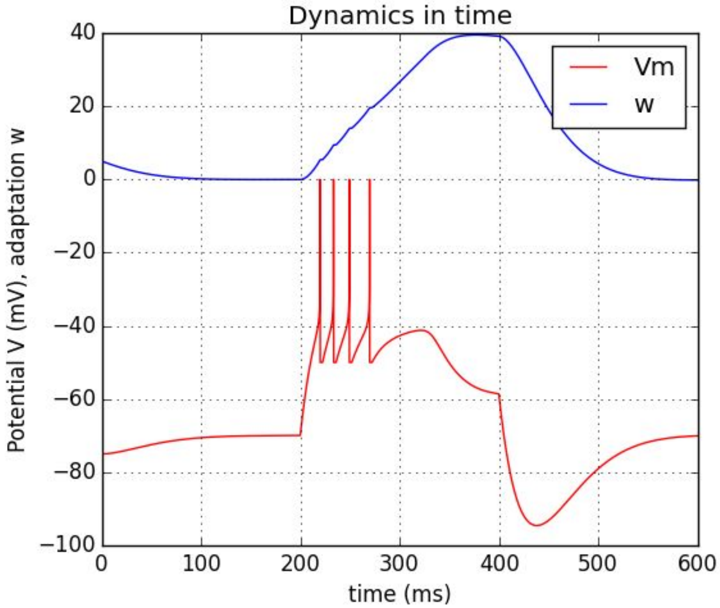
Phase portrait: Bursting cell

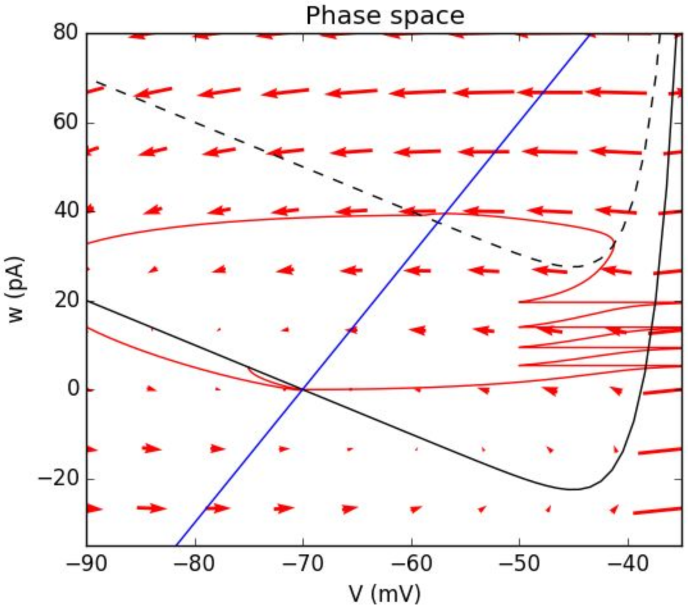




We set $I = 20$ (depolarizing current)

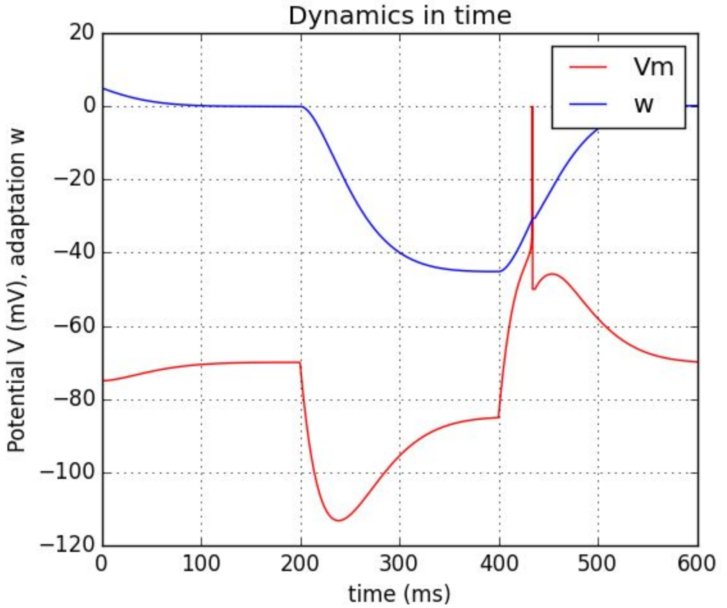
Phase portrait: Bursting cell

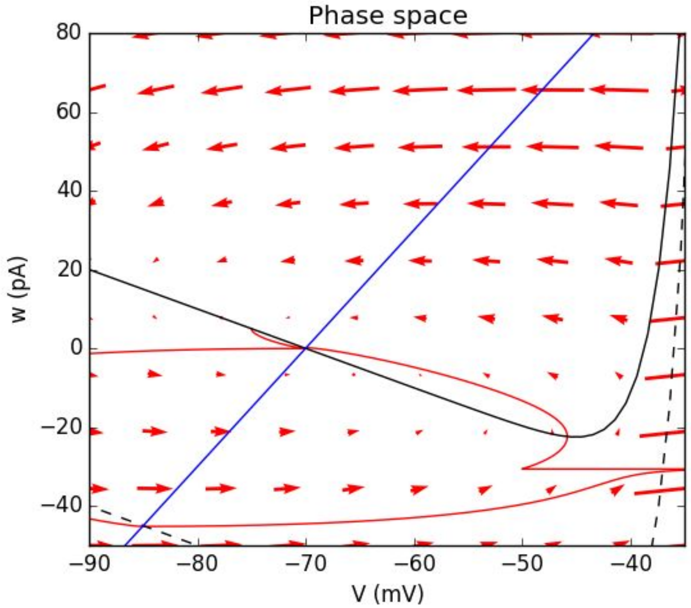




We set $I = 50$ (depolarizing current)

Phase portrait: Bursting cell





We set $I = -60$ (hyperpolarizing current)

Quick reference

- Gerstner et al. “**Neuronal Dynamics**”, https://neuronaldynamics.epfl.ch/online/Ch1.html
- Brette and Gerstener “**Adaptive exponential integrate-and-fire model as an effective description of neuronal activity**”, JNeurophysiology (2005)
- Touboul and Brette “**Dynamics and bifurcations of the adaptive exponential integrate-and-fire model**”, Biol Cybernetics (2008)

Thank you !

Modelling the spiking property of neurons

In neuroscience, the neuron with its spikes is a **"unitary" element**.

Smaller scales (structure and function of channels, membrane trafficking, …) are often interested in *converging* to this scale.

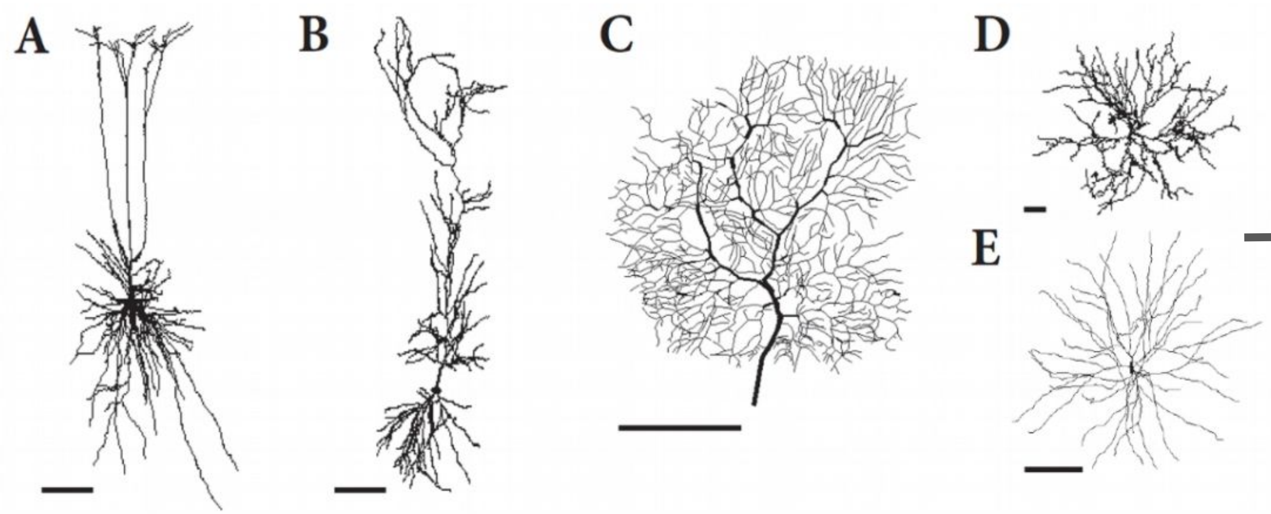
Larger scales (neuronal population dynamics, functional connectivity, …) need to be *explained by* this underlying level.

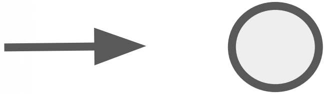
Therefore we are interested in describe, reproduce, understand ... “model” this single neuron scale.

But what do we want to model at this scale, really?

The “point” neuron

To be able to model something we often apply simplifying assumptions.



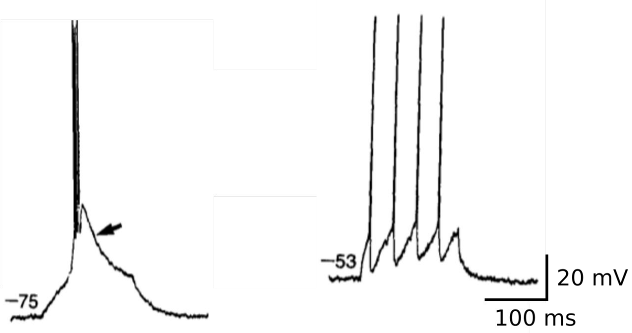


Point neuron

It allows us to concentrate on one particular feature: the **membrane potential**

Neurons **change** their membrane potential **over time**

The membrane potential of real neurons can be measured with a variety of techniques.



Neurons **change** their membrane potential **over time**

The membrane potential of real neurons can be measured with a variety of techniques.

We want to have a model to understand what happens and make predictions.

<u>Today we are going to do it!</u>