Computational Neuroscience

Session 5: ***Mathematical Modeling of Neuronal Populations (1)***

Instructor: Ashkan Damavandi

September 10, 2025

Slides for this session are inspired by Neuromatch's Computational Neuroscience course. You can access the original slides, videos and codes of this course via this link:

https://compneuro.neuromatch.io/tutorials/W2D4_DynamicNetworks/chapter_title.html

1 Neuronal Spiking and Population-Averaged Firing Rate

Single neurons interact with each other by producing electrical pulses, also known as spikes, in response to various stimuli.

Firing Rate: The firing rate of a single neuron is the rate at which these spikes occur over time. This quantity of neurons is typically expressed in spikes per second (Hz). This is a fundamental way to describe how active a neuron is, especially in response to stimulus.

When we consider populations of neurons, we often want to understand and analyze the collective behavior of all the neurons in that network.

Population-Averaged Firing Rate: The population-averaged firing rate is the average firing rate of all neurons in the network at any given moment t. This quantity gives us insights into the overall activity level of a neuronal system.

1.1 F-I Curves and Transfer Functions

In electrophysiology, a neuron is typically characterized by its response to injected currents, specifically by the firing rate. This relationship is often visualized using the F–I curve, which plots the output firing rate (F) against the input current (I). A typical F–I curve shows that low input currents lead to no firing, intermediate input currents cause a rapid increase in firing, and very high input currents cause saturation, where the neuron can no longer increase its firing rate despite higher inputs. For modeling neuronal populations, we often transition from single neuron behavior to describing the population firing rate. To do that, we use transfer functions $F(\cdot)$. Transfer functions are mathematical representations of the relationship between the total input to a population of neurons and their collective output firing rate. In computational neuroscience, transfer functions are used to model how neurons respond to synaptic inputs, which is essential in understanding network dynamics and processes like learning, memory, and decision-making. These functions can take various forms depending on the nature of the neuron or population of neurons being modeled. They typically map the input current (I) to the output firing rate (F) in a way that captures the nonlinearities inherent in biological neuronal systems. The transfer function's shape dictates how sensitive a neuron is to change in input and how quickly its firing rate reaches saturation. The sigmoidal function is one of the most widely used transfer functions due to its simplicity and biological relevance. It models a neuron that shows a nonlinear response to input, with an onset threshold and saturation at high input levels. The general form of the sigmoidal transfer function is:

$$F(x; a, \theta) = \frac{1}{1 + e^{-a(x-\theta)}} - \frac{1}{1 + e^{a\theta}}$$

Where x is the total input of the neuronal population, a is the gain which controls the steepness of the curve, and θ is the threshold which is the point at which the neurons begin firing. The Sigmoid transfer function has several advantages, including its biological realism, as it captures the threshold behavior of neurons, meaning they only fire when the input reaches a certain level. It also naturally models saturation, where further increases in input no longer lead to a significant increase in firing rate, representing the neuron "saturating" at high input levels. Additionally, the sigmoid provides a smooth transition from low to high firing rates, which is characteristic of many biological systems. However, it has limitations, such as its saturation, which, while biologically realistic, can limit the model’s flexibility in scenarios where neurons might continue firing linearly at high inputs. Furthermore, the exponential function in the sigmoid can be computationally expensive, particularly in large-scale neural network simulations.

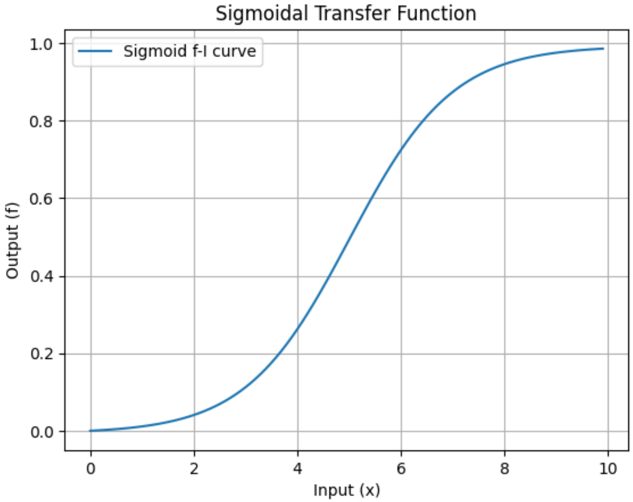


Figure 1: Simulated Sigmoidal Transfer Function. $a = 1, \theta = 0.5$

The Rectified Linear Unit (ReLU) is another kind of transfer function which is defined as:

$$ReLU(x) = \max(0, x)$$

This function produces an output that is zero for any negative input and increases linearly with positive input. ReLU has several advantages, including its simplicity, which makes it computationally efficient and a preferred choice for large-scale simulations or deep neural networks. Despite being simple, ReLU introduces a nonlinearity by outputting zero for negative inputs, and it also promotes sparsity in the network, which can be beneficial for learning tasks in artificial neural networks. However, it has limitations, such as the absence of saturation, meaning that large inputs continue to cause large outputs, which may be unrealistic in some biological scenarios where neurons eventually reach a maximum firing rate. Additionally, ReLU suffers from the "dying ReLU" problem, where neurons with very small or negative inputs output zero indefinitely, preventing those neurons from contributing to the network's learning, a common issue in neural networks.

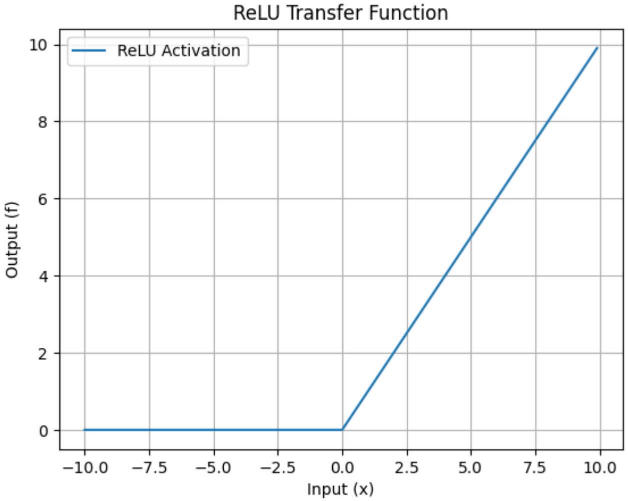


Figure 2: Simulated ReLU Transfer Function.

The tanh (hyperbolic tangent) function is another commonly used transfer function, which is defined as:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

It outputs values between -1 and 1, with a smooth and symmetric curve around zero. The hyperbolic tangent function has several advantages, including its symmetry around zero, which is useful for models where symmetry is biologically realistic, such as balancing excitatory and inhibitory signals. Like the sigmoid function, tanh also saturates both high and low input values, making it suitable for modeling neurons with bounded firing rates. Its output range of $[-1,1]$ ensures that the neuron's firing rate remains within bounds, which is desirable in certain network models. However, tanh has limitations, including saturation at both extremes, which can be disadvantageous if the model requires a greater range of firing rates at high and low inputs. Additionally, the sigmoid-like shape of tanh can lead to slower convergence during training in neural networks, particularly due to its steep gradient near the origin.

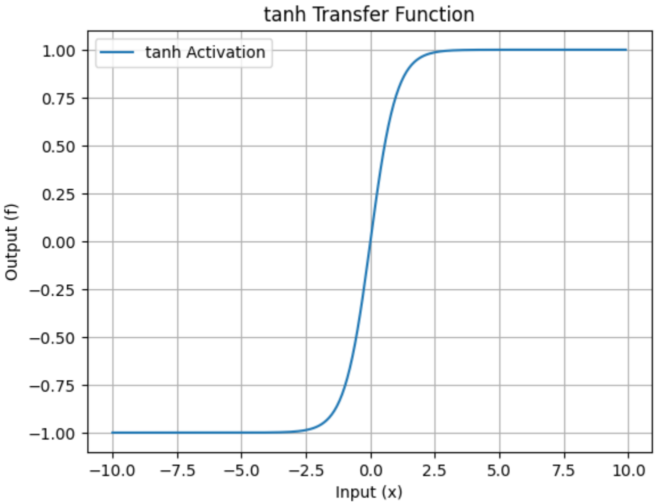


Figure 3: Simulated Hyperbolic Transfer Function.

Another simple function is the Linear Threshold, which is defined as:

$$F(x) = \max(0, x - \theta)$$

Where θ is a threshold, and the output is a linear function of the input once the threshold is exceeded. The linear threshold function has several advantages, including its computational efficiency, as it is extremely simple and computationally inexpensive, and its linear output, which is useful for modeling neurons that respond proportionally to input once a threshold is crossed. However, it also has limitations, such as the lack of saturation, like ReLU, which may not be suitable in biological contexts where neurons eventually reach a maximum firing rate at high inputs. Additionally, the function is discontinuous at the threshold, which may not be biologically realistic in contexts where neurons exhibit smoother transitions.

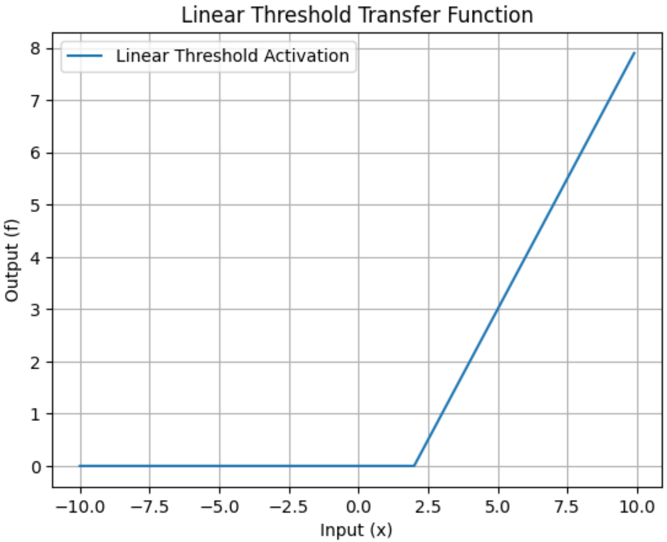


Figure 4: Simulated Linear Threshold Transfer Function. $\theta = 2$.

The choice of transfer functions is very important in modeling the dynamics of a neuronal network. Sigmoid functions are useful for modeling neurons with realistic saturation behavior, particularly when modeling populations of neurons in models like Wilson-Cowan. ReLU, on the other hand, is preferred for large-scale artificial neural networks due to its simplicity and computational efficiency. Tanh is a good compromise between the sigmoid and ReLU, offering saturation while maintaining symmetry. The linear threshold function is the simplest but lacks the desirable saturation and smoothness of other functions. For large-scale population models, the choice of transfer function should be guided by the biological realism required in the model and the specific behavior that needs to be captured, whether it be smooth transitions, saturation, or computational efficiency.

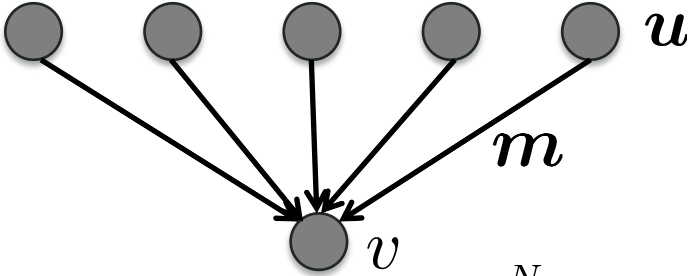
1.2 Excitatory and Inhibitory Neuronal Networks

Excitatory networks consist of neurons that increase the activity or firing rate of other neurons when they are activated, releasing excitatory neurotransmitters like glutamate, which cause depolarization and make the receiving neuron more likely to fire. These networks are essential for processes such as learning and memory, motor control, and cognitive functions like decision-making and attention. In contrast, inhibitory networks consist of neurons that decrease the activity of other neurons, releasing inhibitory neurotransmitters like GABA, which cause hyperpolarization and make receiving neurons less likely to fire. These networks are crucial for controlling excitatory activity, preventing overstimulation, modulating behavior, and generating brain rhythms essential for processes like sleep and attention. The balance between excitatory and inhibitory activity is vital for proper brain function, with an imbalance leading to conditions like epilepsy, anxiety, schizophrenia, depression, or cognitive disorders. This balance is necessary for synaptic plasticity, signal precision, and coordinated neural oscillations, ensuring the brain works efficiently by both stimulating and regulating neural activity.

A feedforward network

Consists of three components:

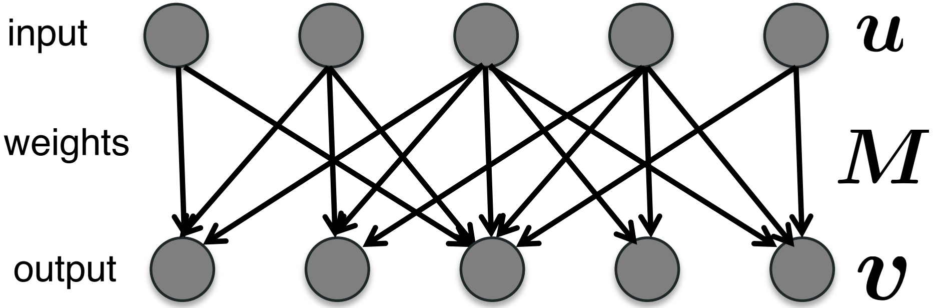
1. The activity of presynaptic neurons: $\boldsymbol{u}$
2. The activity of postsynaptic neurons: v
3. The strength of the synaptic weights between the pre- and postsynaptic neurons: $\boldsymbol{m}$



$$I = \sum_{j=1}^{N} m_j u_j$$

Firing rate equation: $\tau_r \frac{dv}{dt} = -v + F(I) = -v + F(\boldsymbol{m} \cdot \boldsymbol{u})$

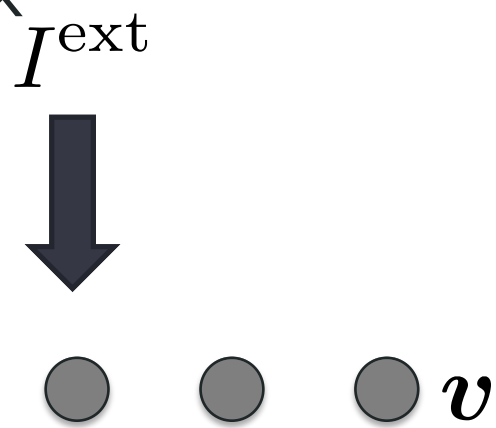
A feedforward network



$$I^{\text{ext}} = \boldsymbol{M}\boldsymbol{u}$$

$$\tau_E \frac{d\boldsymbol{v}}{dt} = -\boldsymbol{v} + F(I^{\text{ext}})$$

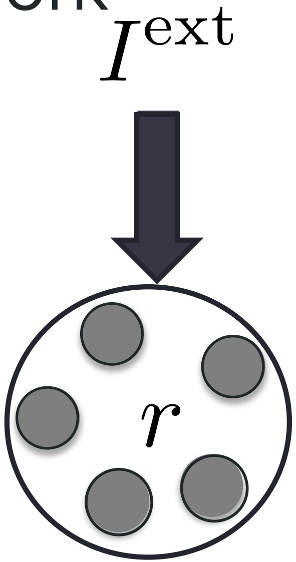
A feedforward network



A vector of firing rates

$$\tau_E \frac{d\boldsymbol{v}}{dt} = -\boldsymbol{v} + F(I^{\text{ext}})$$

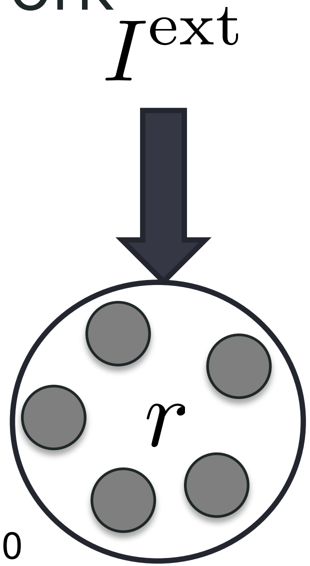
A feedforward network



$$\tau \frac{dr}{dt} = -r + F(I^{\text{ext}})$$

Firing rate of
the population
– a scalar

A feedforward network



$$\tau \cancelto{0}{\frac{dr}{dt}} = -r + F(I^{\text{ext}})$$

Steady state:

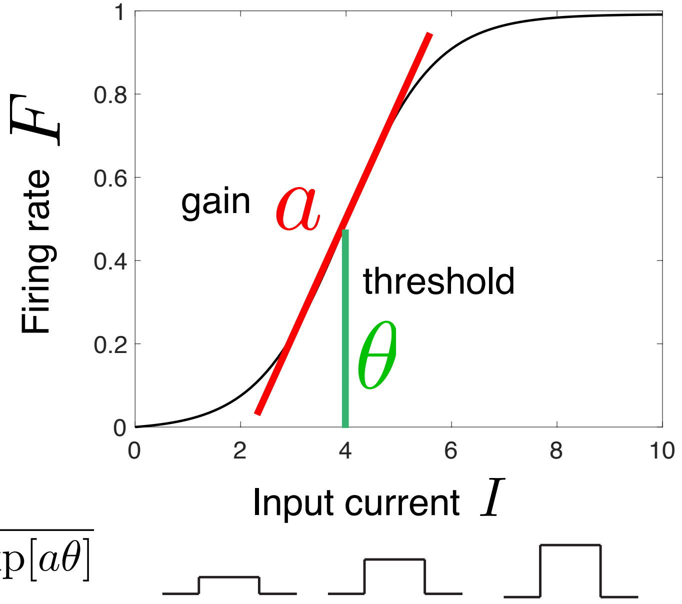
$$r = F(I^{\text{ext}})$$

Firing rate curves

Also known as:

- F-I curve
- Transfer function
- Stimulus-response function
- Tuning curve

$$r = F(I)$$



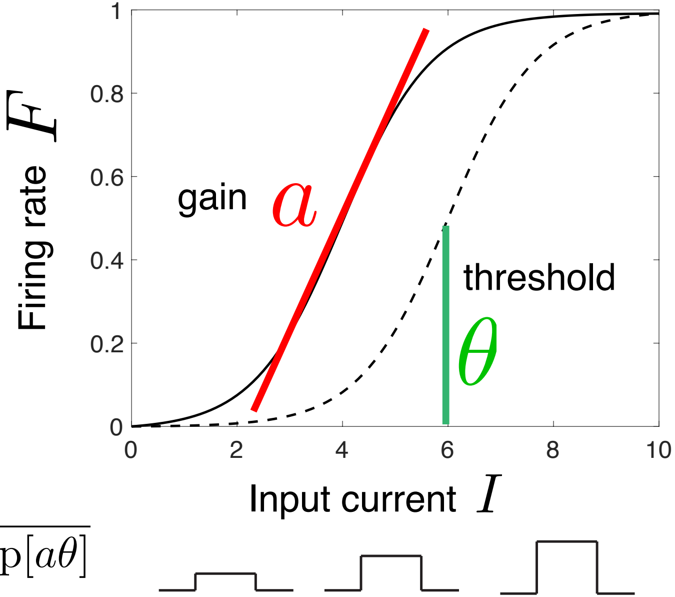
$$F(I; a, \theta) = \frac{1}{1 + \exp[-a(I - \theta)]} - \frac{1}{1 + \exp[a\theta]}$$

Firing rate curves

Also known as:

- F-I curve
- Transfer function
- Stimulus-response function
- Tuning curve

$$r = F(I)$$

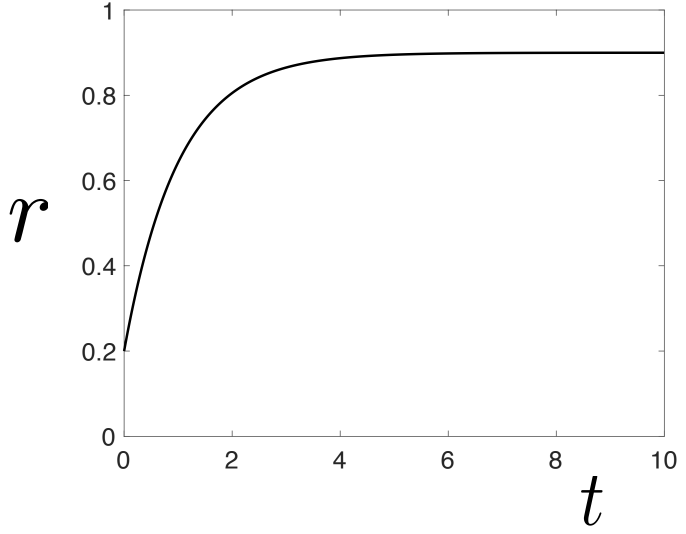


$$F(I; a, \theta) = \frac{1}{1 + \exp[-a(I - \theta)]} - \frac{1}{1 + \exp[a\theta]}$$

Solving E vs. time

We can solve how the excitatory population converges to its steady state.

$$r = F(I)$$

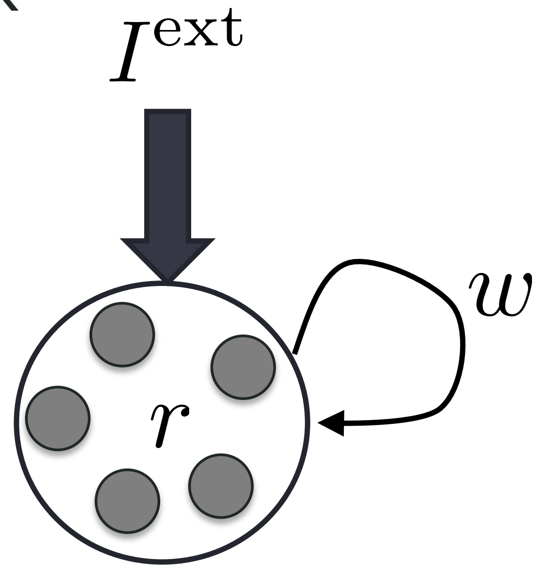


A recurrent network

This can be thought of as:

A one-dimensional (1D) dynamical system

Exhibits a diversity of behaviors depending on input and connectivity



$$\tau \frac{dr}{dt} = -r + F(w \cdot r + I^{\text{ext}})$$

Fixed points

A 1D dynamical system has fixed points,

or equilibria,

or steady-states

where the dynamics do not change.

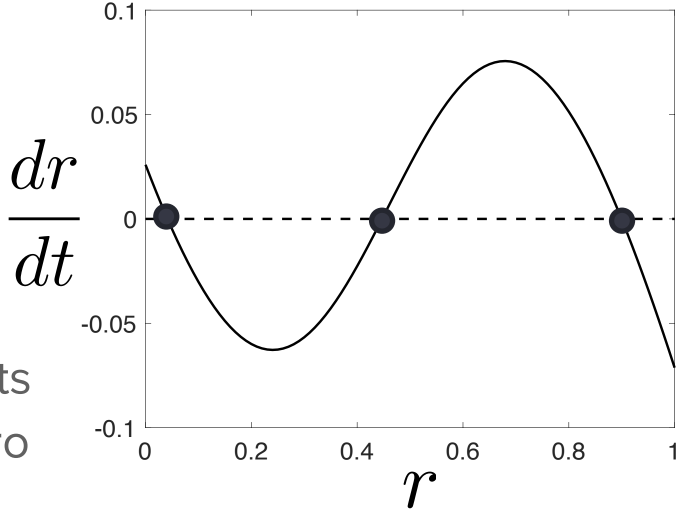
$$\tau \cancelto{0}{\frac{dr}{dt}} = -r + F(w \cdot r + I^{\text{ext}})$$

$$r = F(w \cdot r + I^{\text{ext}})$$

A vector field / phase plane and fixed points

$$\cancelto{0}{\frac{dr}{dt}} = \frac{1}{\tau}\left[-r + F(w \cdot r + I^{\text{ext}})\right]$$

At the fired points, the dynamics of the network do not change.



Fixed points are the zero crossings

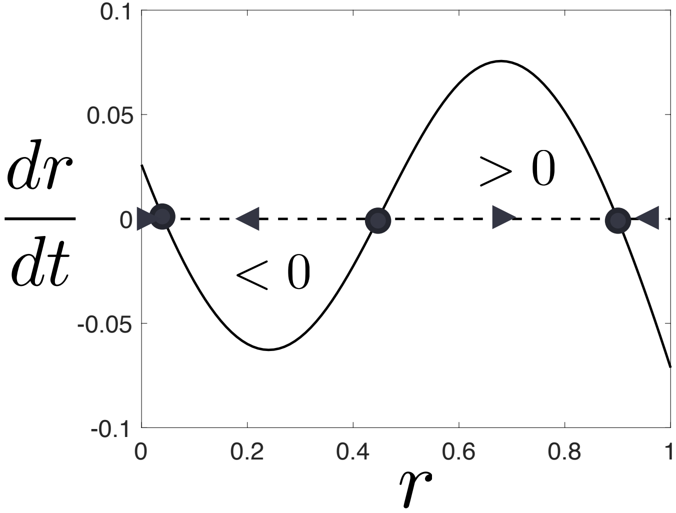
A vector field / phase plane

$$\frac{dr}{dt} = \frac{1}{\tau}\left[-r + F(w \cdot r + I^{\text{ext}})\right]$$

Flow along a vector field:

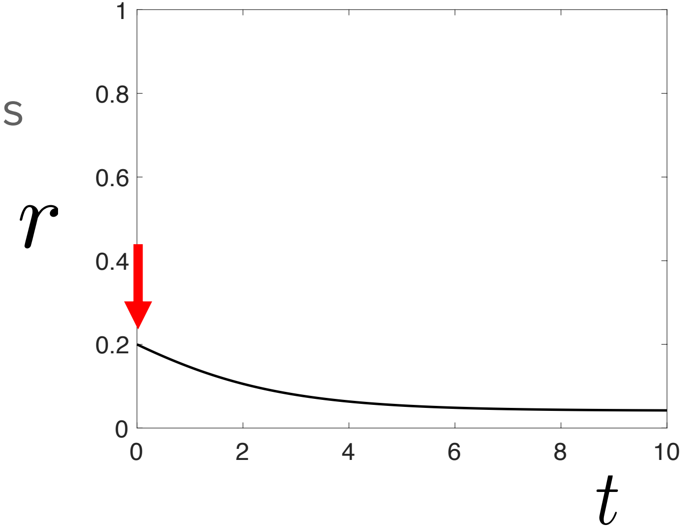
velocity vector $\dot{r} = \dfrac{dr}{dt}$

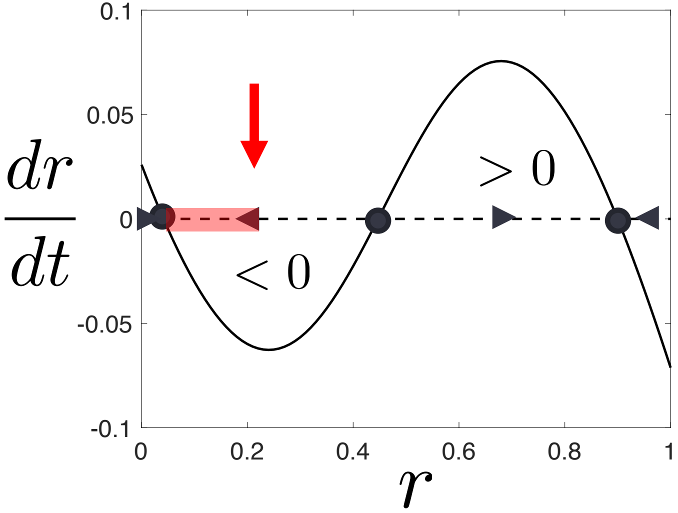
at each r



A vector field / phase plane

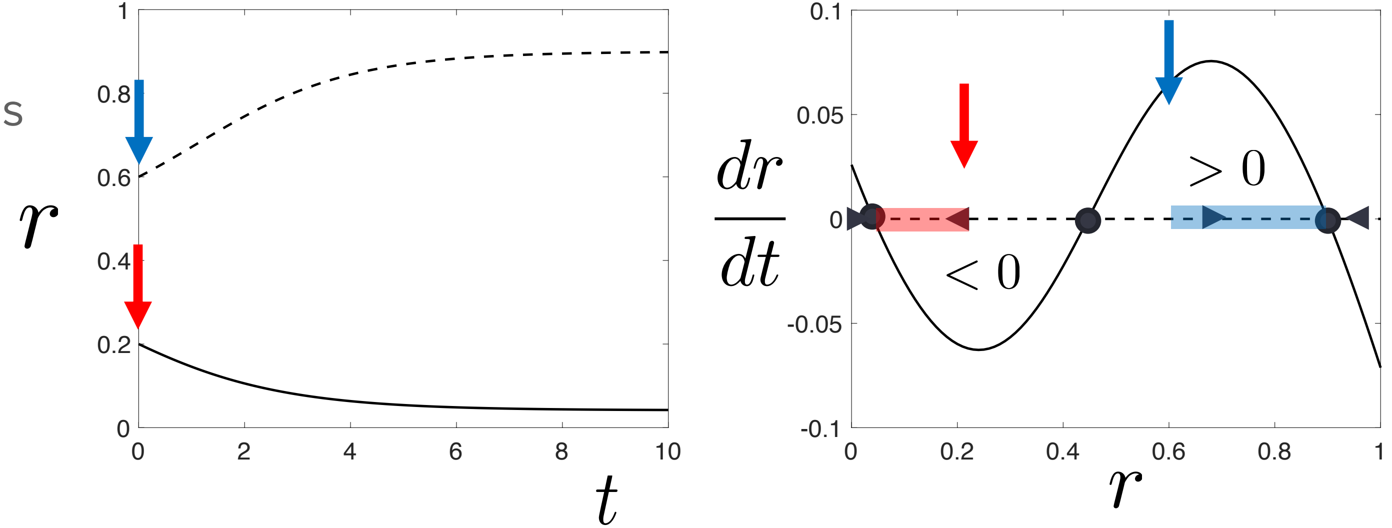
Flow along a vector field: determines where activity goes





A vector field / phase plane

Flow along a vector field: determines where activity goes



A vector field / phase plane

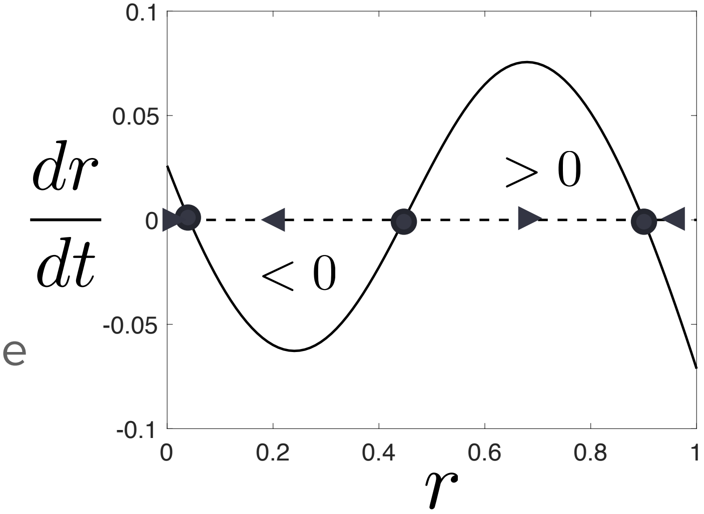
$$\frac{dr}{dt} = \frac{1}{\tau}\left[-r + F(w \cdot r + I^{\text{ext}})\right]$$

Flow along a vector field:

velocity vector $\dot{r} = \frac{dr}{dt}$

at each r

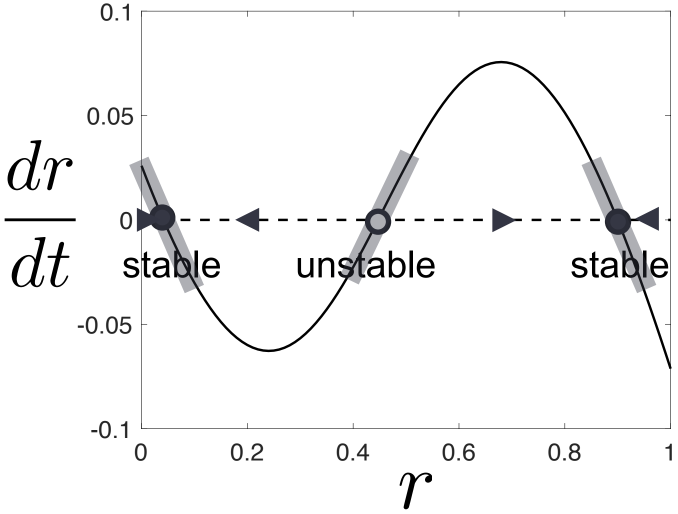
Fixed points are the zero crossings



Finding stability of fixed points

$$\frac{dr}{dt} = \frac{1}{\tau}\left[-r + F(w \cdot r + I^{\text{ext}})\right]$$

- Stability of fixed point refers to local perturbations around the fixed point.
- We have to linearize around the fixed point.
- How do small perturbations around the fixed point grow or decay?



Finding stability of fixed points

- Let r^* be a fixed point. Then

$$0 = \frac{dr^*}{dt} = \frac{1}{\tau}\left[-r^* + F(w \cdot r^* + I^{\text{ext}})\right]$$

- Consider a perturbation around the fixed point

$$r = r^* + \epsilon$$

- Plug into the main equation and obtain a first-order approximation for ϵ

$$\frac{d\epsilon}{dt} = \frac{1}{\tau}\left[-\epsilon + \frac{dF}{dt}(w \cdot r^* + I^{\text{ext}})\ w \cdot \epsilon\right]$$

- Rewrite

$$\frac{d\epsilon}{dt} = \frac{\epsilon}{\tau}\left[-1 + w\frac{dF}{dt}(w \cdot r^* + I^{\text{ext}})\right]$$

- This equation has as a solution an exponential $\epsilon(t) \sim e^{\lambda t}$ with the exponent

$$\lambda = \frac{1}{\tau}\left[-1 + w\frac{dF}{dt}(w \cdot r^* + I^{\text{ext}})\right]$$

- The sign of λ determines the stability!

Stability of fixed points

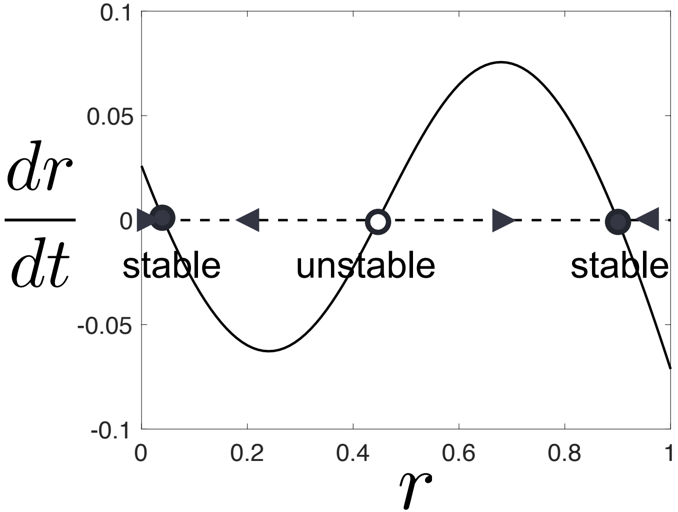
Called the eigenvalue of the dynamical system!

- The sign of λ determines the stability!

$$\lambda = \frac{1}{\tau}\left[-1 + w\frac{dF}{dt}(w \cdot r^* + I^{\text{ext}})\right]$$

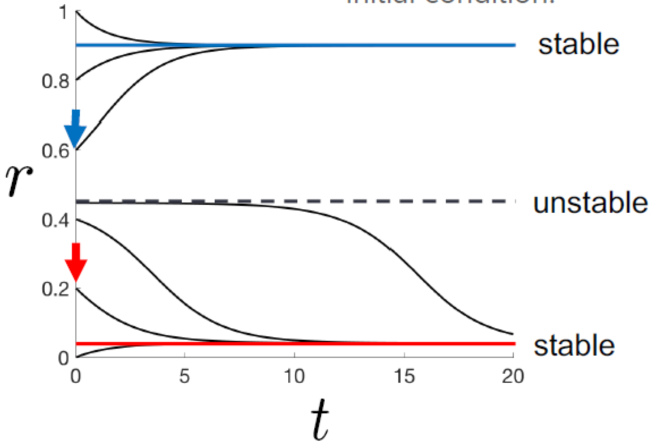
$\lambda < 0$ The fixed points is stable, Activity will converge to r^*

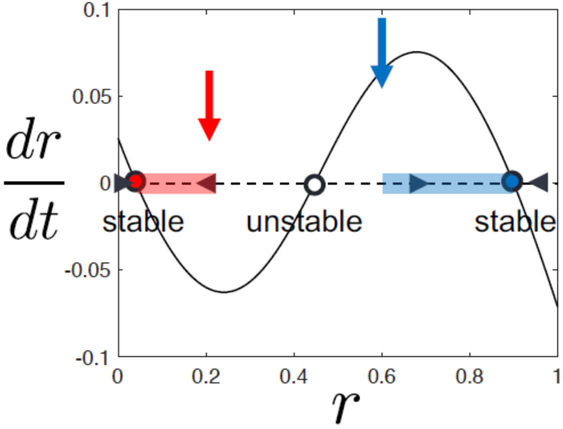
$\lambda > 0$ The fixed points is unstable, Activity will move away from r^*



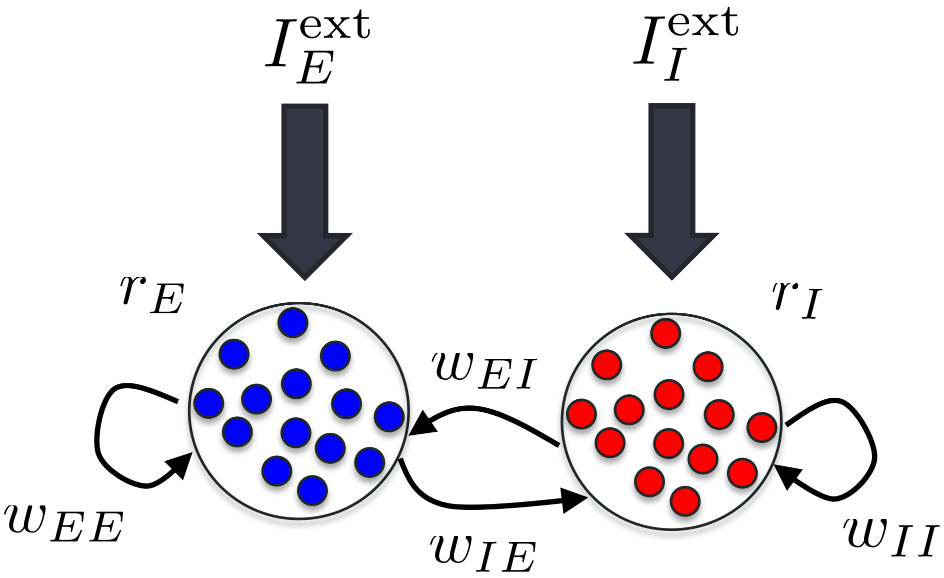
Neural activity

Activity converges to stable fixed points independent of initial condition!



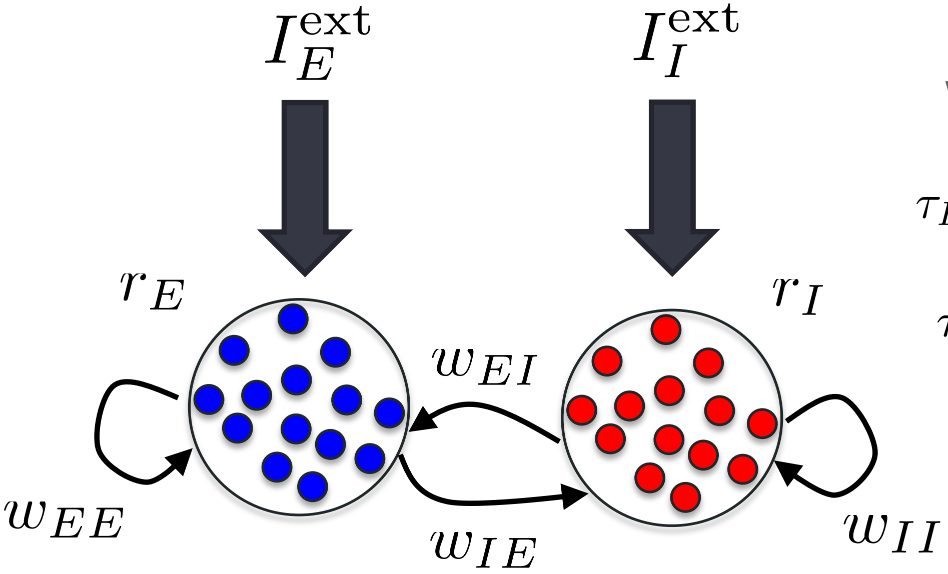


Excitatory and inhibitory networks



- We can model entire populations of excitatory and inhibitory neurons with the same rate-based equations
- The two populations are coupled to each other
- We will study their steady-states and dynamics
- A powerful model framework: the Wilson-Cowan rate model

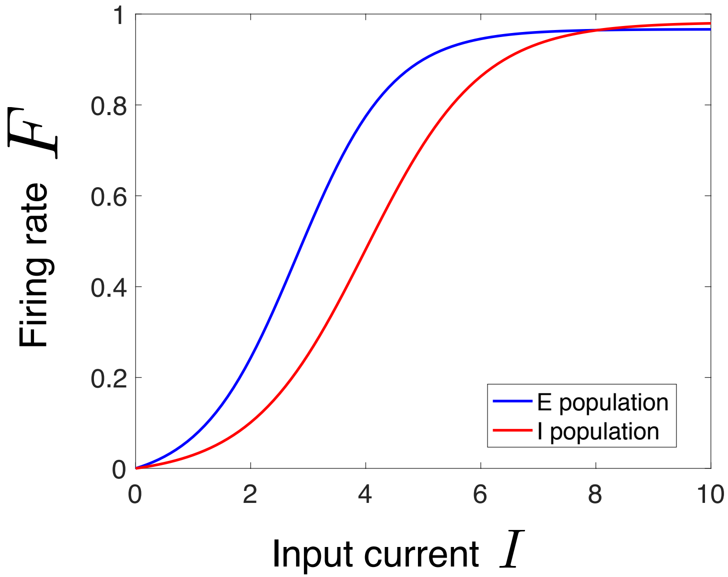
Excitatory and inhibitory networks



Wilson-Cowan equations:

$$\tau_E \frac{dr_E}{dt} = -r_E + F_E(w_{EE} r_E - w_{EI} r_I + I_E^{\text{ext}})$$
$$\tau_I \frac{dr_I}{dt} = -r_I + F_I(w_{IE} r_E - w_{II} r_I + I_I^{\text{ext}})$$

Excitatory and inhibitory networks



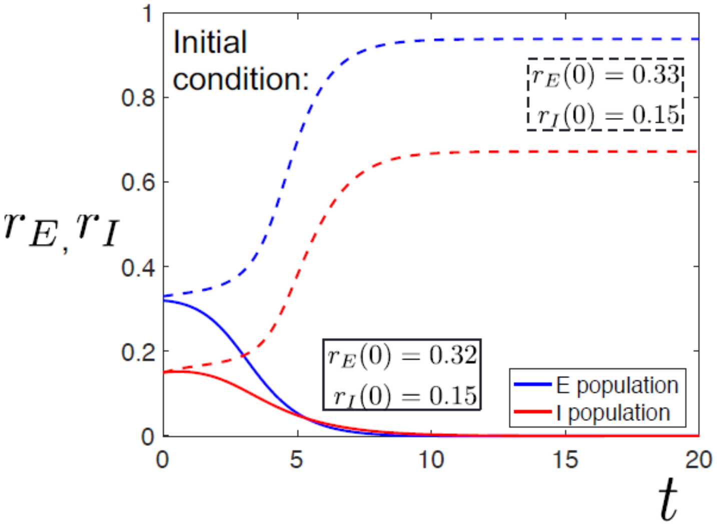
Recall: F-I curves of each population

Wilson-Cowan equations:

$$\tau_E \frac{dr_E}{dt} = -r_E + F_E(w_{EE}r_E - w_{EI}r_I + I_E^{\text{ext}})$$

$$\tau_I \frac{dr_I}{dt} = -r_I + F_I(w_{IE}r_E - w_{II}r_I + I_I^{\text{ext}})$$

Evolution of E and I dynamics

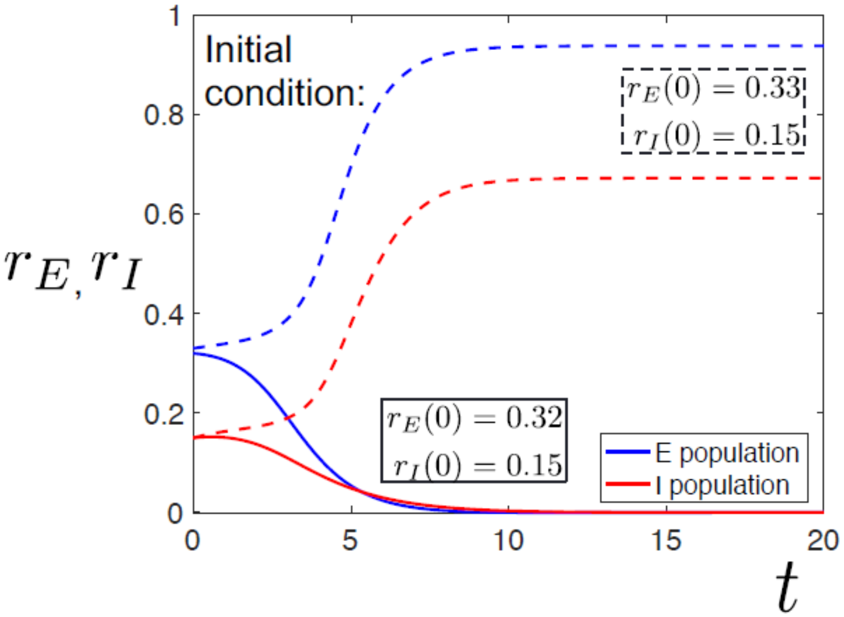


$$\tau_E \frac{dr_E}{dt} = -r_E + F_E(w_{EE}r_E - w_{EI}r_I + I_E^{\text{ext}})$$

$$\tau_I \frac{dr_I}{dt} = -r_I + F_I(w_{IE}r_E - w_{II}r_I + I_I^{\text{ext}})$$

Choose a slightly different initial condition: The two populations converge to very different steady states!

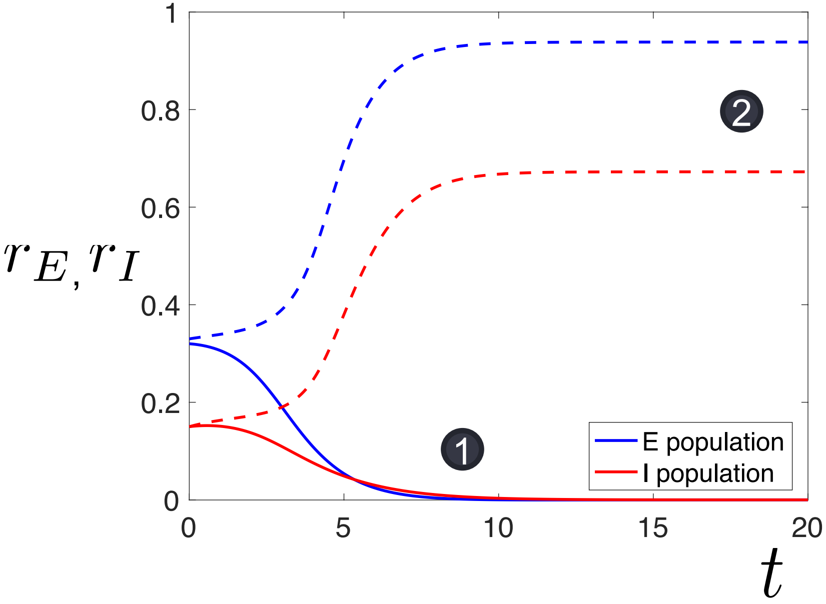
Activity vs. time plane



We showed that we can plot the activity of the two populations as a function of time.

Alternatively...
We can plot the activity of the two populations against each other.

Activity vs. time plane

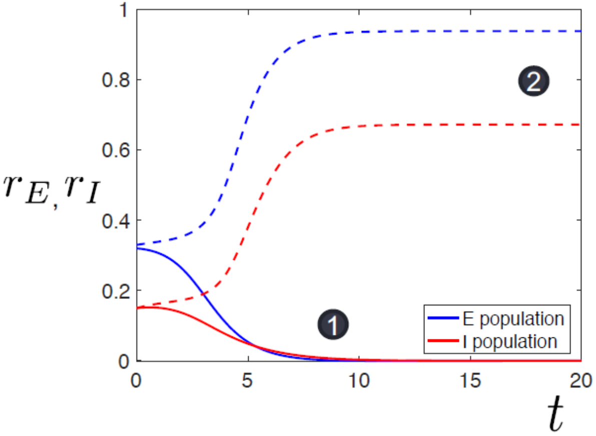


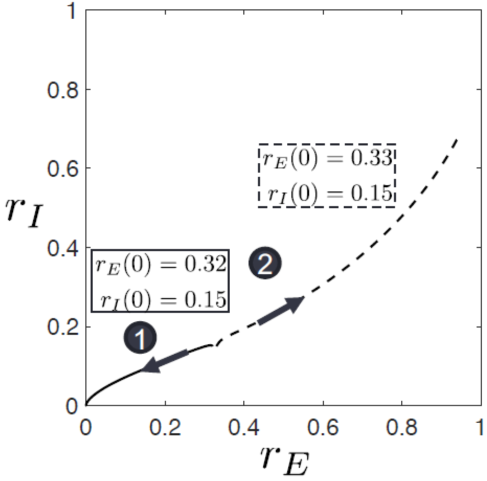
We showed that we can plot the activity of the two populations as a function of time.

Alternatively…

We can plot the activity of the two populations against each other in the **phase plane**.

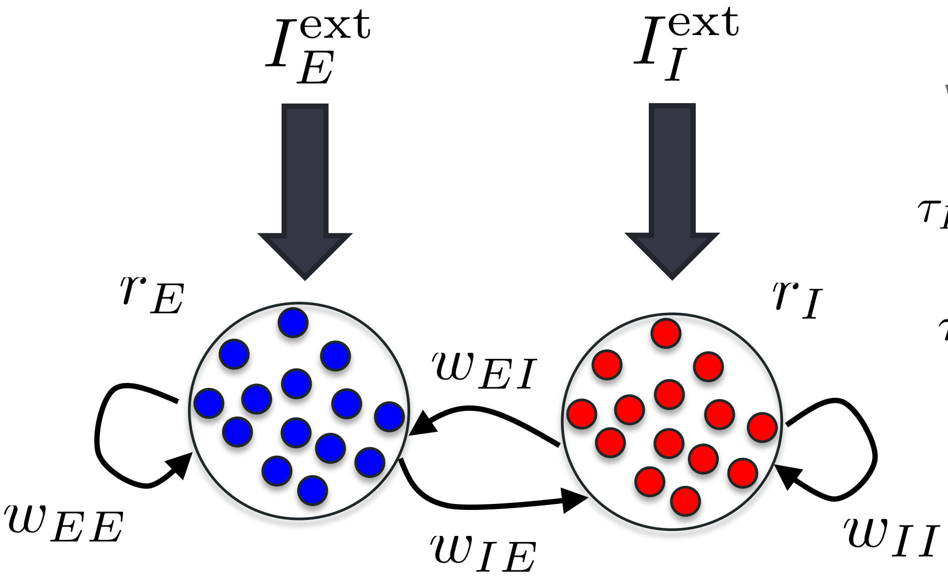
Phase plane





In the **phase plane**, each curve indicates how the two populations evolve in time.

Excitatory and inhibitory networks



Wilson-Cowan equations:

$$\tau_E \frac{dr_E}{dt} = -r_E + F_E(w_{EE} r_E - w_{EI} r_I + I_E^{\text{ext}})$$
$$\tau_I \frac{dr_I}{dt} = -r_I + F_I(w_{IE} r_E - w_{II} r_I + I_I^{\text{ext}})$$

Nullclines of the Wilson-Cowan model

To understand the dynamics of the Wilson-Cowan model, we return to the concept of phase planes and vector fields introduced for a 1D dynamical system.

$$\tau_E \frac{dr_E}{dt} = -r_E + F_E(w_{EE}r_E - w_{EI}r_I + I_E^{\text{ext}})$$
$$\tau_I \frac{dr_I}{dt} = -r_I + F_I(w_{IE}r_E - w_{II}r_I + I_I^{\text{ext}})$$

E nullcline $\quad \dfrac{dr_E}{dt} = 0$

The set of points in the phase plane where the excitatory population doesn’t change.

I nullcline $\quad \dfrac{dr_I}{dt} = 0$

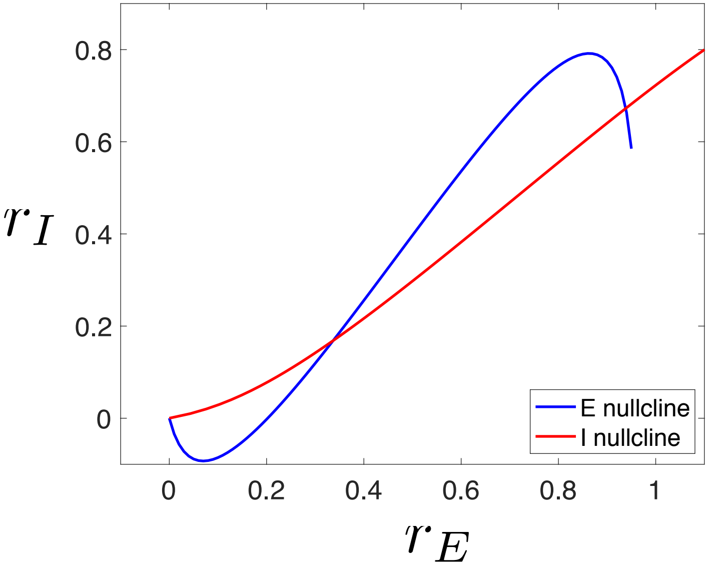
The set of points in the phase plane where the inhibitory population doesn’t change.

Nullclines of the Wilson-Cowan model

To understand the dynamics of the Wilson-Cowan model, we return to the concept of phase planes and vector fields introduced for a 1D dynamical system.

E nullcline $\qquad \dfrac{dr_E}{dt} = 0$

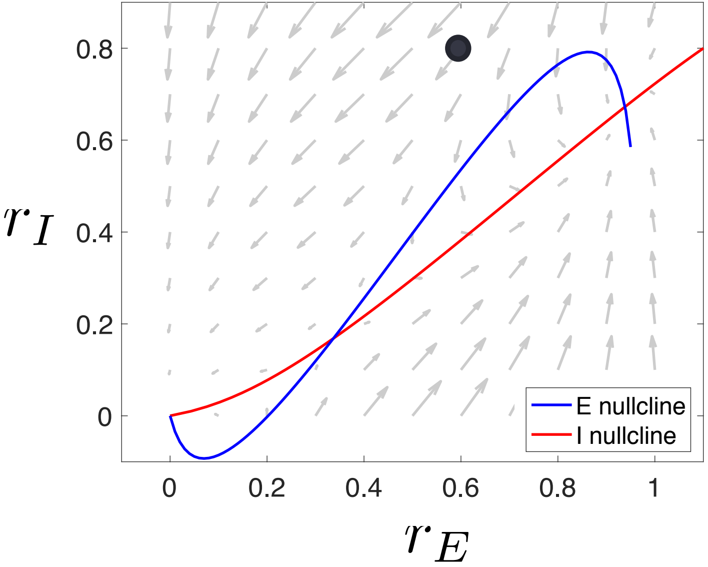
I nullcline $\qquad \dfrac{dr_I}{dt} = 0$



Vector field of the Wilson-Cowan model

We can think of the evolution of activity as a flow along a vector field: at each point, we have the activity of the neural population in a given time.

Let us a pick one E and one I in the plane: $r_E(0) = 0.6, r_I(0) = 0.8$

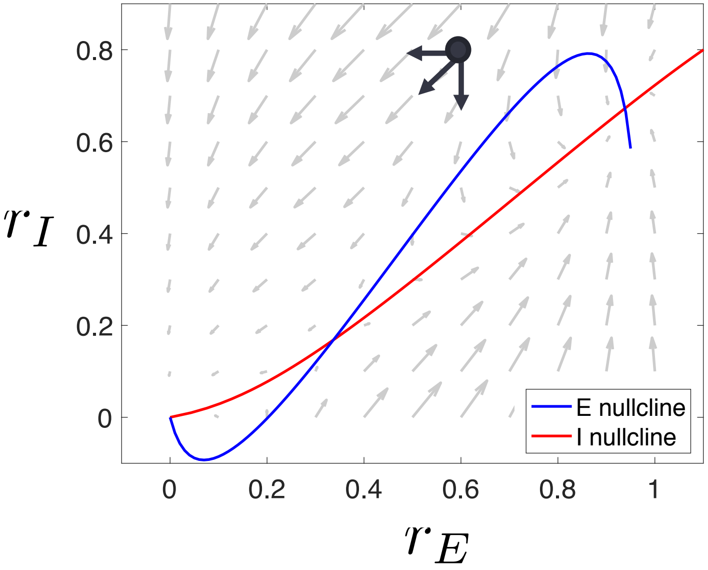


Vector field of the Wilson-Cowan model

We can think of the evolution of activity as a flow along a vector field: at each point, we have the activity of the neural population in a given time.

Let us a pick one E and one I in the plane: $r_E(0) = 0.6, r_I(0) = 0.8$

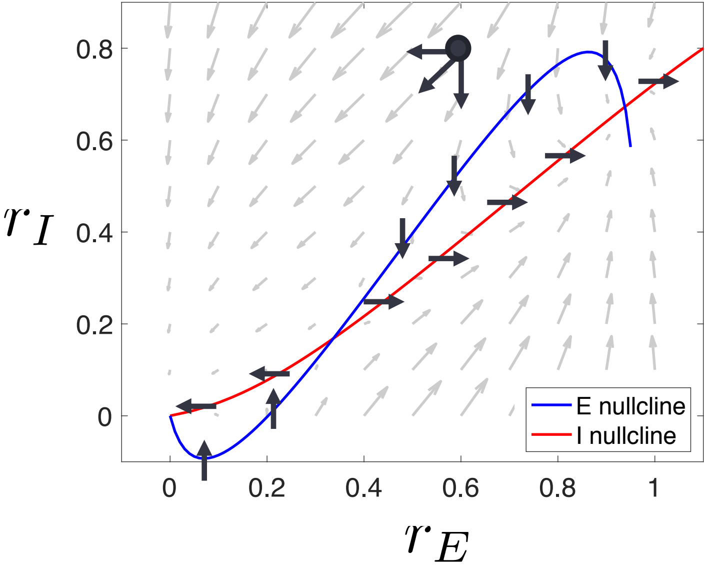
$$\frac{dr_E}{dt} < 0 \qquad \frac{dr_I}{dt} < 0$$



Vector field of the Wilson-Cowan model

The flow along the excitatory nullcline is zero in the horizontal direction, meaning the excitatory population does not change.

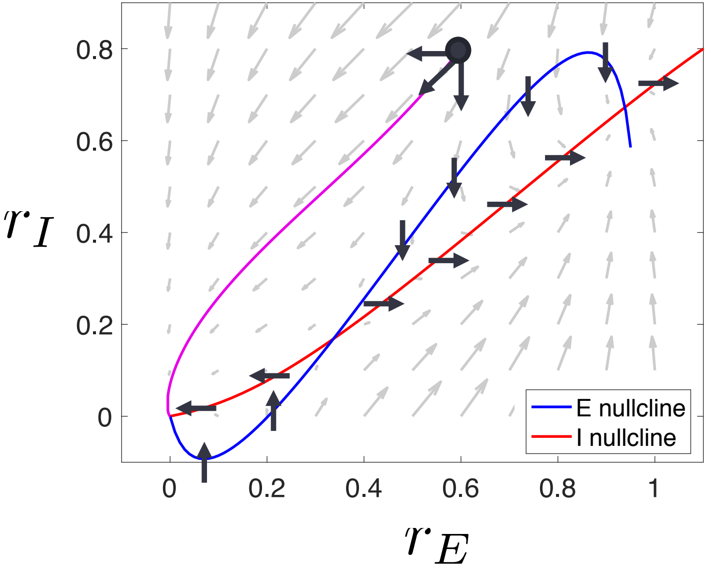
The flow along the inhibitory nullcline is zero in the vertical direction, meaning the inhibitory population does not change.



Vector field of the Wilson-Cowan model

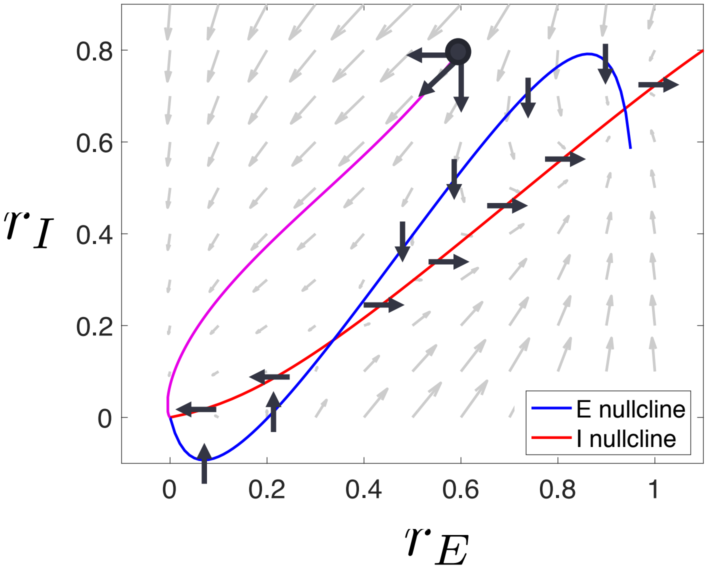
Recall: in the phase plane each curve indicates how the two populations evolve in time.

Initial condition: $r_E(0) = 0.6, r_I(0) = 0.8$



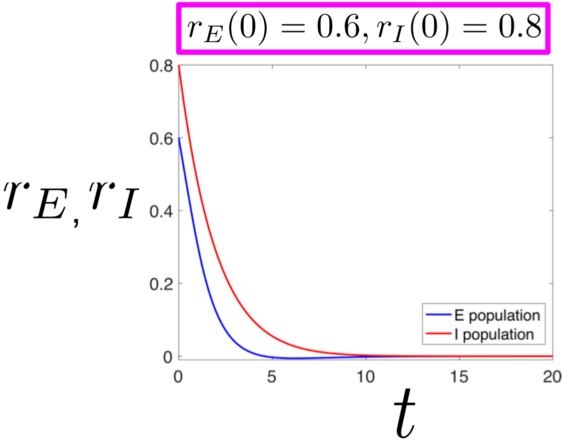
Vector field of the Wilson-Cowan model

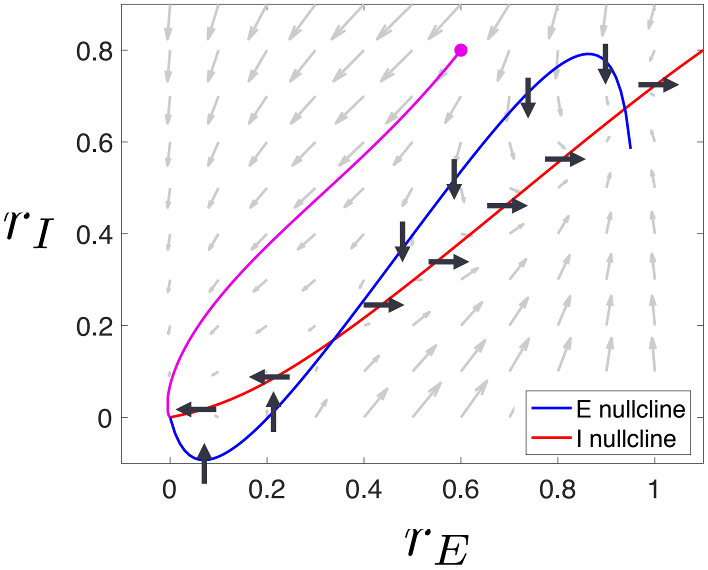
The tangent corresponds to how fast the activity changes (for both populations).



Evolution of population activity over time

Can also plot the population vs. time for the same initial condition.

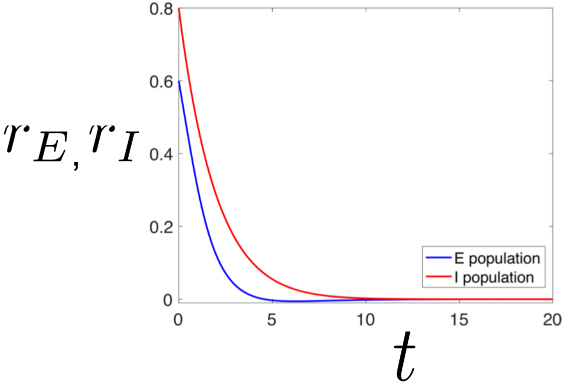


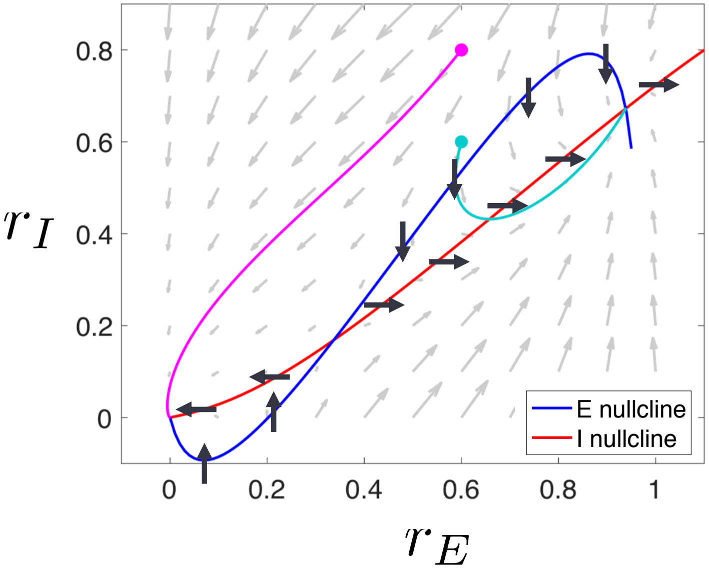


Evolution of population activity over time

Let us choose a second (different) initial condition.

$r_E(0) = 0.6, r_I(0) = 0.8$



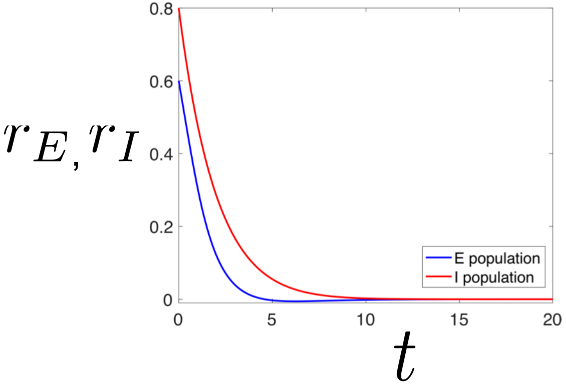


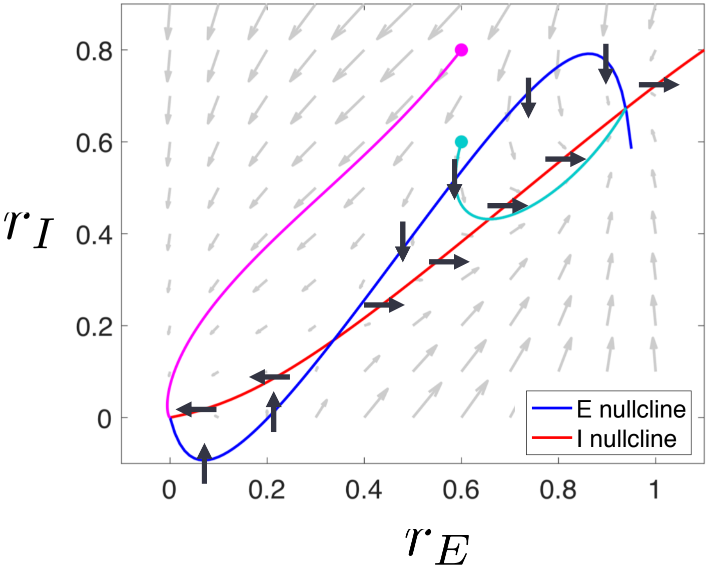
Evolution of population activity over time

Again we plot the population vs. time for that initial condition.

$r_E(0) = 0.6, r_I(0) = 0.8$

$r_E(0) = 0.6, r_I(0) = 0.6$

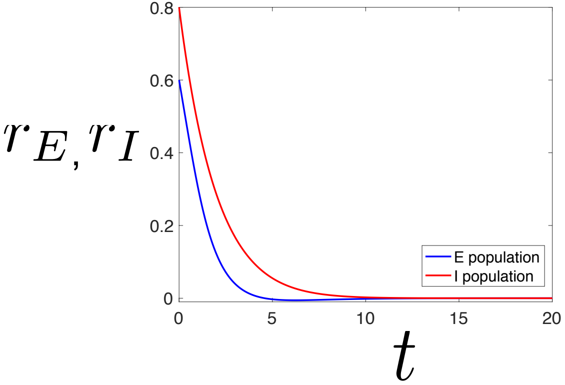




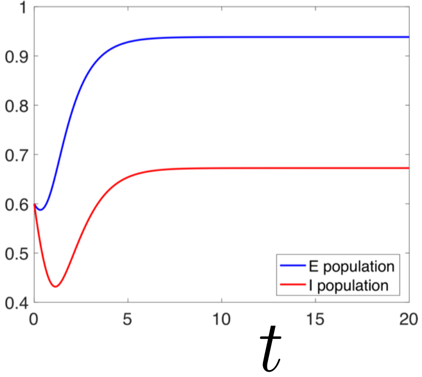
Evolution of population activity over time

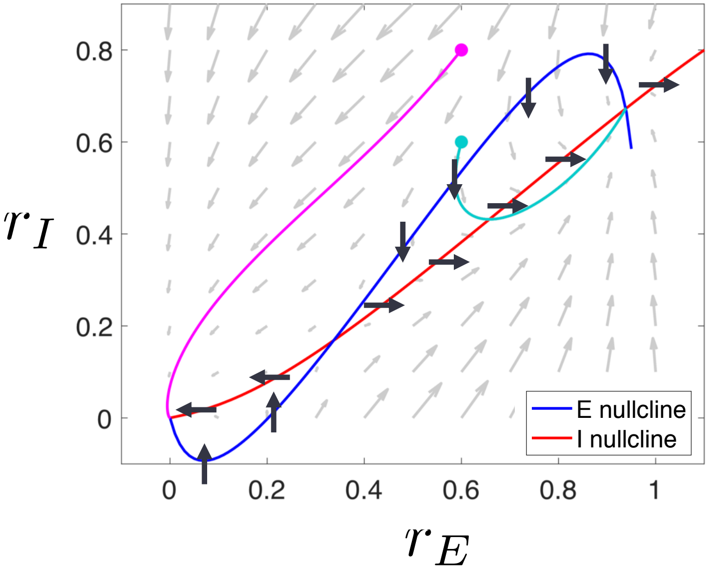
Again we plot the population vs. time for that initial condition.

$r_E(0) = 0.6, r_I(0) = 0.8$



$r_E(0) = 0.6, r_I(0) = 0.6$





Fixed points of the Wilson-Cowan model

Just like in 1D, a dynamical system in 2D
has fixed points,

or equilibria,

or steady-states

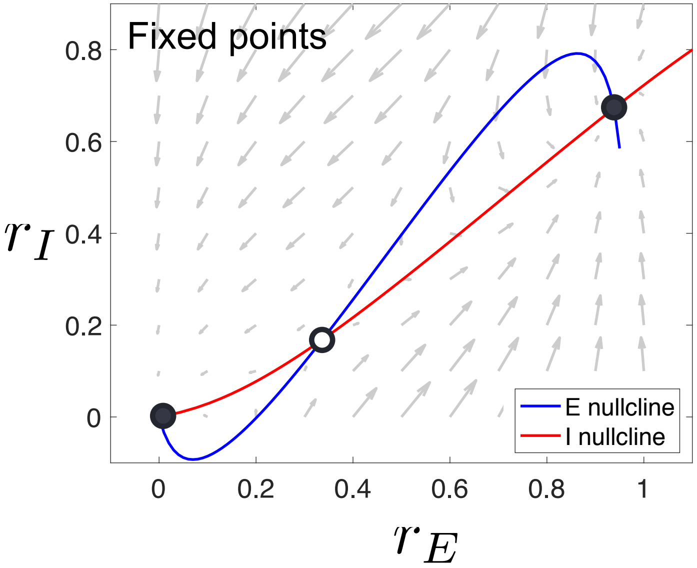
where the dynamics do not change.

$$\frac{dr_E}{dt} = 0 \quad \text{and} \quad \frac{dr_I}{dt} = 0$$

Fixed points of the Wilson-Cowan model

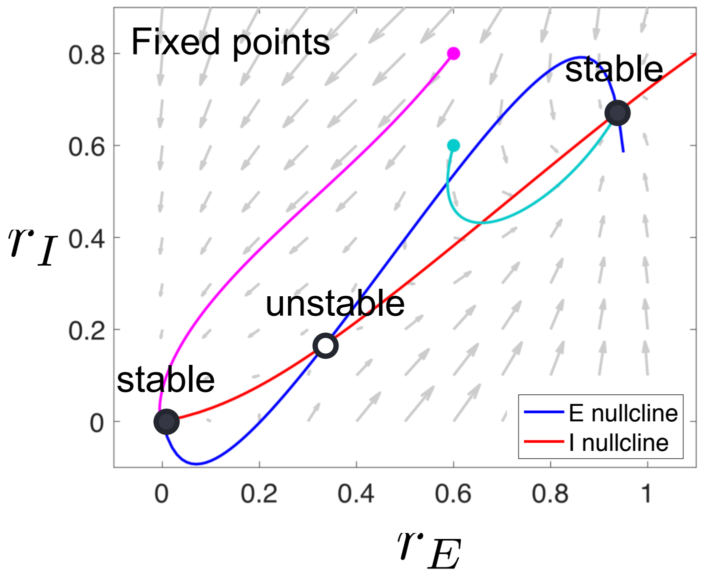
You can find the fixed points where the nullclines intersect.

$$\frac{dr_E}{dt} = 0 \quad \text{and} \quad \frac{dr_I}{dt} = 0$$



Fixed points of the Wilson-Cowan model

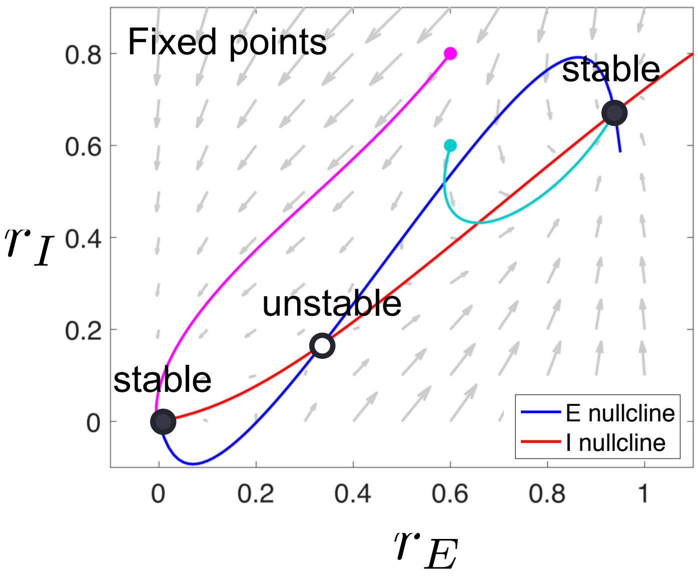
Some fixed points are stable, attracting all population trajectories.



Fixed points of the Wilson-Cowan model

Some fixed points are stable, attracting all population trajectories.

Other fixed points are unstable, repelling all population trajectories.



Stability of fixed points

How do we determine fixed point stability?

Linearize the dynamical system around a fixed point:

Recall:

$$\tau_E \frac{dr_E}{dt} = -r_E + F_E(w_{EE} r_E - w_{EI} r_I + I_E^{\text{ext}})$$
$$\tau_I \frac{dr_I}{dt} = -r_I + F_I(w_{IE} r_E - w_{II} r_I + I_I^{\text{ext}})$$

Rearrange:

$$\frac{dr_E}{dt} = \frac{1}{\tau_E}\left[-r_E + F_E(w_{EE} r_E - w_{EI} r_I + I_E^{\text{ext}})\right]$$
$$\frac{dr_I}{dt} = \frac{1}{\tau_I}\left[-r_I + F_I(w_{IE} r_E - w_{II} r_I + I_I^{\text{ext}})\right]$$

Stability of fixed points

Rename:

$$\frac{dr_E}{dt} = \underbrace{\frac{1}{\tau_E}\left[-r_E + F_E(w_{EE}r_E - w_{EI}r_I + I_E^{\text{ext}})\right]}_{G_E(r_E,r_I)}$$

$$\frac{dr_I}{dt} = \underbrace{\frac{1}{\tau_I}\left[-r_I + F_I(w_{IE}r_E - w_{II}r_I + I_I^{\text{ext}})\right]}_{G_I(r_E,r_I)}$$

⇒

$$\frac{dr_E}{dt} = G_E(r_E, r_I)$$

$$\frac{dr_I}{dt} = G_I(r_E, r_I)$$

Linearize the dynamical system around a fixed point: (r_E^*, r_I^*)

Stability of fixed points

Compute the Jacobian matrix at the fixed point:

$$J = \begin{pmatrix} \frac{\partial G_E}{\partial r_E} & \frac{\partial G_E}{\partial r_I} \\ \\ \frac{\partial G_I}{\partial r_E} & \frac{\partial G_I}{\partial r_I} \end{pmatrix}_{|(r_E^*, r_I^*)}$$

Compute eigenvalues of Jacobian at fixed point:

Case 1: Real eigenvalues

Case 2: Complex eigenvalues

Stability of fixed points

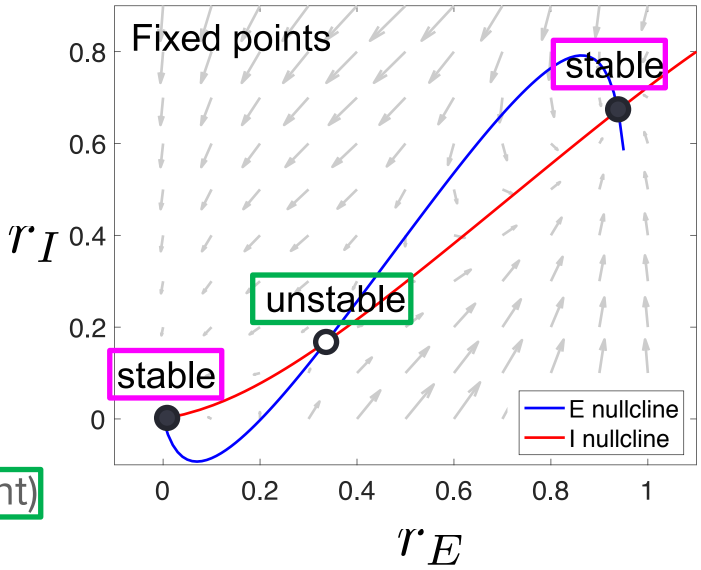
Compute eigenvalues of Jacobian at fixed point:

Case 1: Real eigenvalues λ_1, λ_2

If $\lambda_1 < 0, \lambda_2 < 0$ stable fixed point (node)

If $\lambda_1 > 0, \lambda_2 > 0$ unstable fixed point (node)

If $\lambda_1 > 0, \lambda_2 < 0$ unstable fixed point (saddle point)



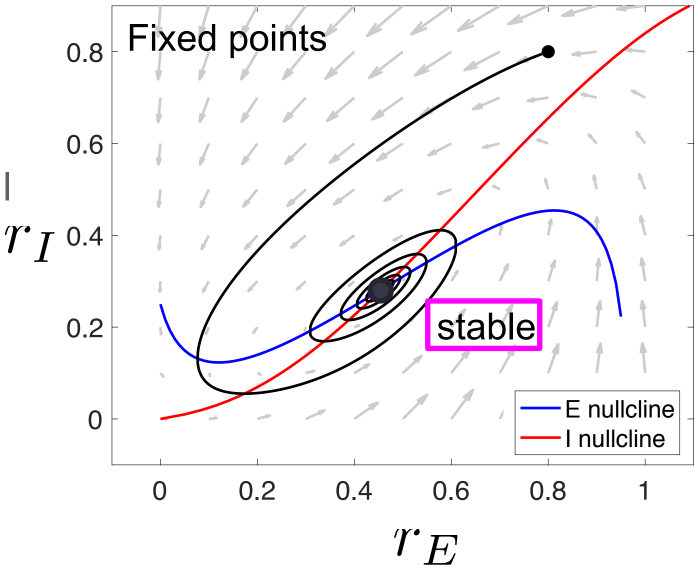
Stability of fixed points

Compute eigenvalues of Jacobian at fixed point:

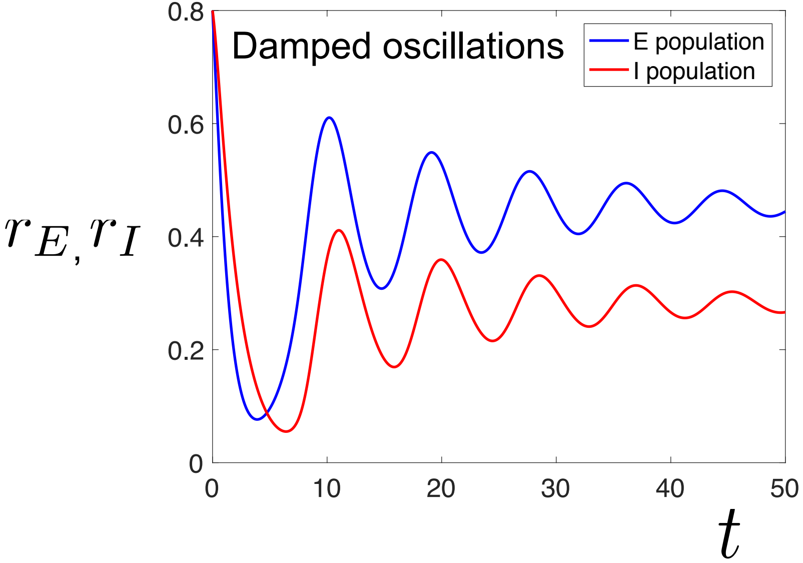
Case 2: Imaginary eigenvalues λ_1, λ_2 (same real part, opposite-in-sign imaginary part)

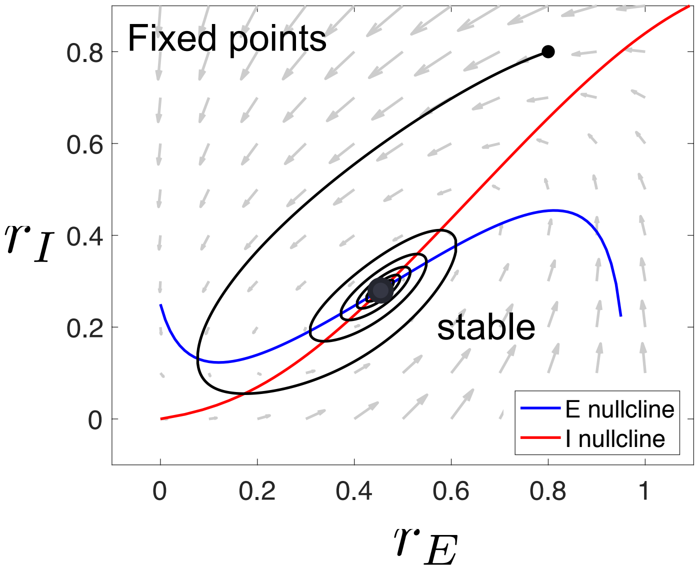
If real part is negative: stable fixed point (focus)

If real part is positive: unstable fixed point (focus)



Stability of fixed points



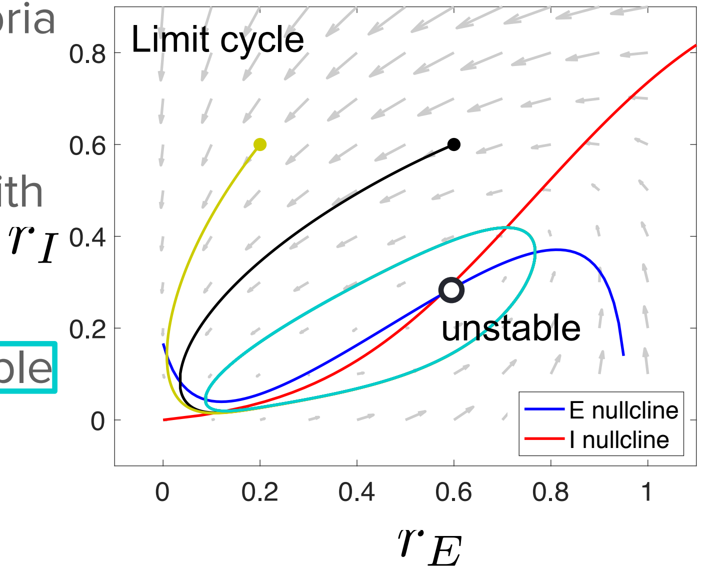


Stability of fixed points

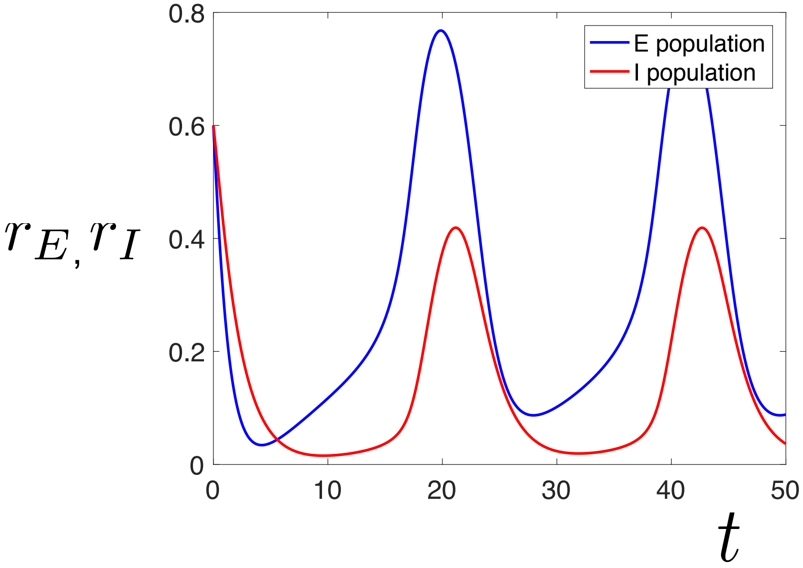
In 2D systems, we can also get other equilibria (steady states) that are not fixed points:

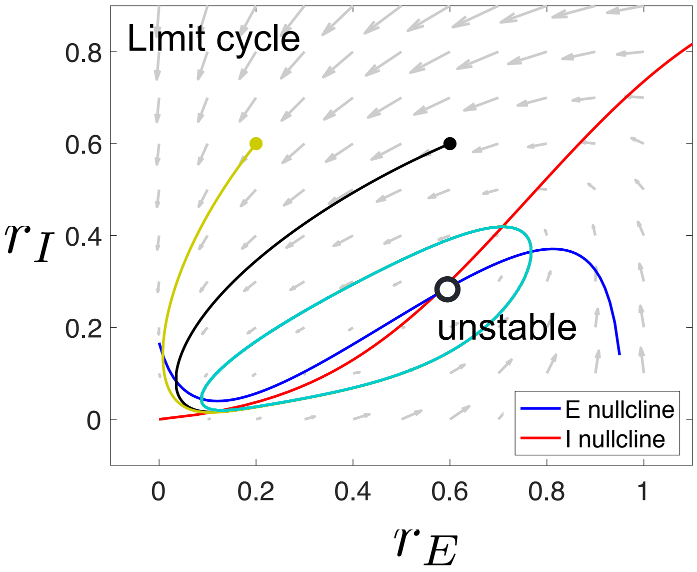
Limit cycles – which generate oscillations with fixed amplitude

Just like fixed points, limit cycles can be stable (attracting) or unstable (repelling).



Stability of fixed points


E population
I population
r_E, r_I
t


Limit cycle
unstable
E nullcline
I nullcline
r_I
r_E