Welcome!
#pod-031
Week2, Day2

(Reviewed by Deepak)

neuromatch academy

facebook Reality Labs

Agenda

- **Day-#7**
  - Tutorial part 1 (Deterministic Linear Dynamic Systems)
    - 3 Exercises
  - Tutorial part 2 (Probabilistic Linear Dynamic Systems)
    - 3 Exercises
  - Tutorial part 3 (Dynamical systems with stochasticity)
    - 4 Exercises
  - Tutorial part 4 (Autoregressive models)
    - 2 Exercises

W2D2_pod 031

Tutorial #1 Explanations

Objective

Express behavior of dynamical systems -- systems that evolve in time -- where the rules by which they evolve in time are described precisely by a differential Equation.

Differential equations are equations that express the **rate of change** of the state variable x.

W2D2_pod 031

Linear Dynamics

rate of change of state variable x

$$\frac{dx}{dt} = f(x) = \dot{x}$$

linear function of x (LINEAR DYNAMICS).

One dimensional differential equation.

$$\dot{x} = ax$$

scalar

how x evolves are governed by -

$$x(t) = x_0 \exp(at)$$

initial condition
$(x @ t=0)$.

Simulating ordinary diff. equation

→ by approximating/modeling time as discrete

list of time steps $t_0, t_1, t_2 \ldots$

$$t_{i+1} = t_i + dt$$

where $dx = \dot{x}\, dt$

computing x:

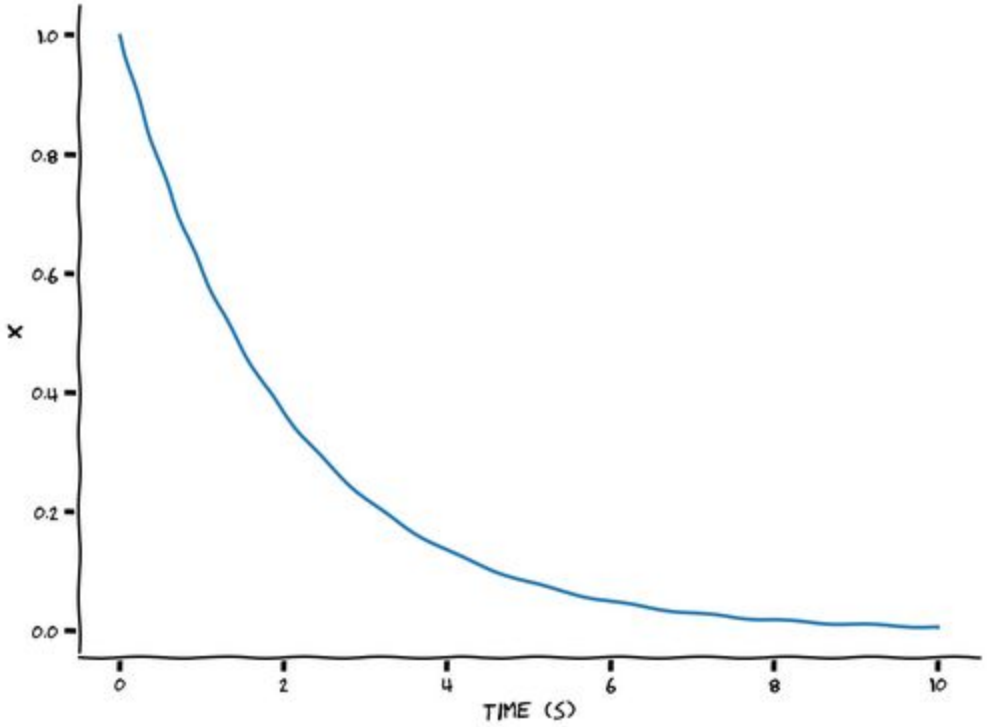
forward Euler integration { $x(t_i) = \sum x$ @ t_{i-1} & small change $dx = \dot{x}\, dt$

$$x(t_i) = x(t_{i-1}) + \dot{x}(t_{i-1})\, dt$$

issues – ① if ord. diff equation is noisy
② when dynamics result in sudden } sign changes of x.

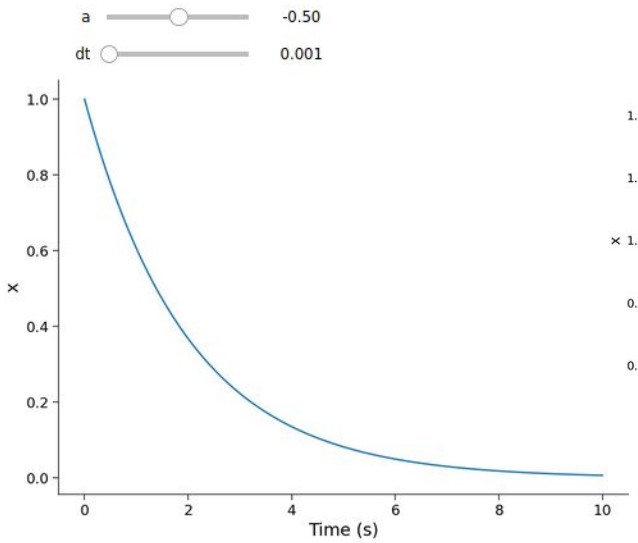
[Excitable neurons]

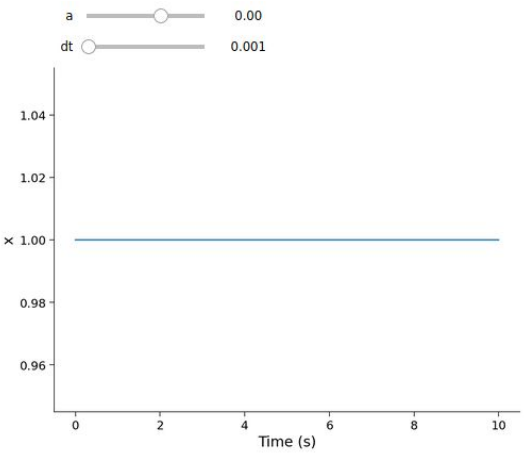
Evolution of x with respect to t

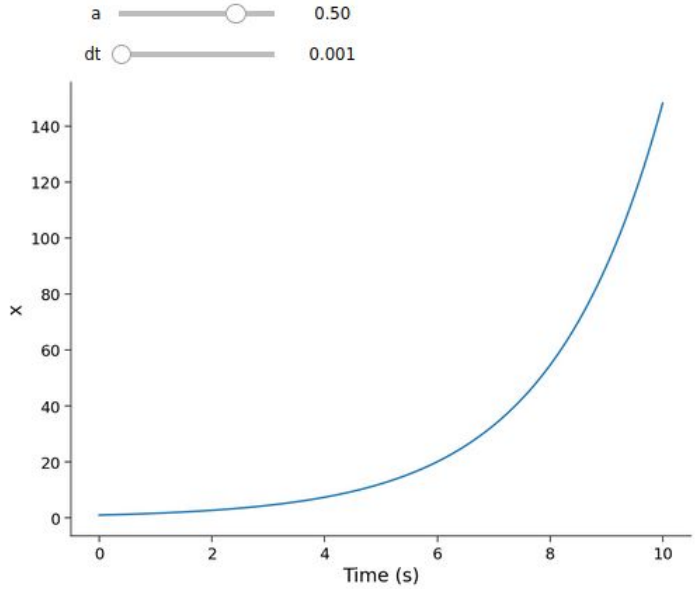


Effect of the scalar a

What happens when you change a? Try values where a<0 and a>0

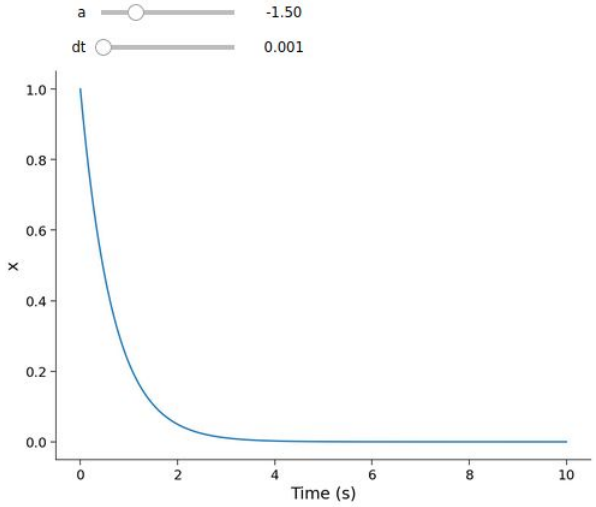


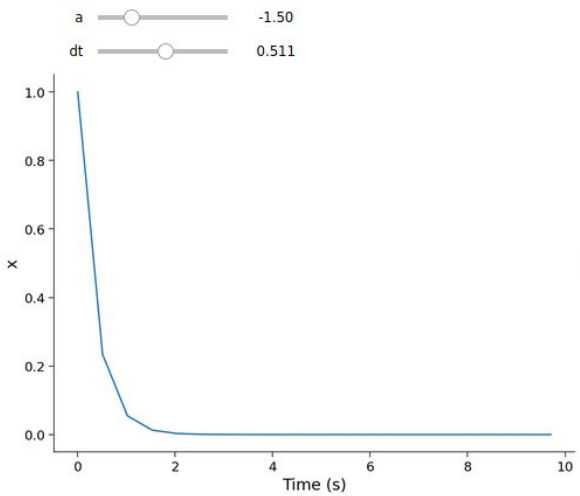


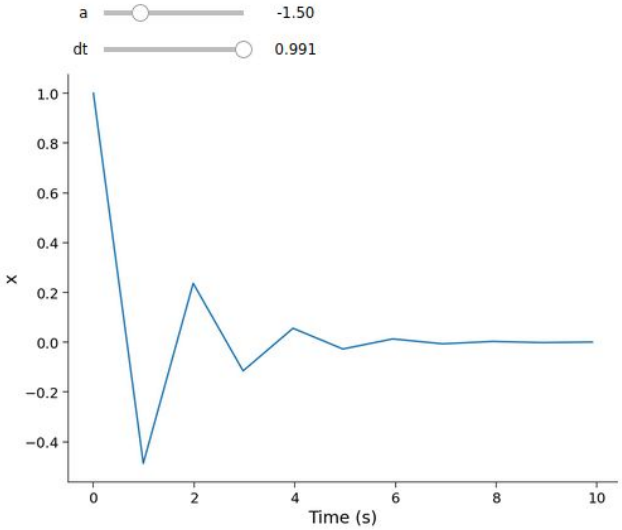


Effect of dt

The dt is the step size of the forward Euler integration. Try a=-1.5 and increase dt. What happens to the numerical solution when you increase dt?







Explanation

For a<0, the solution decays in time.
For a>0, the solution grows in time.
For a=0, the solution stays at 1 (and is stable).

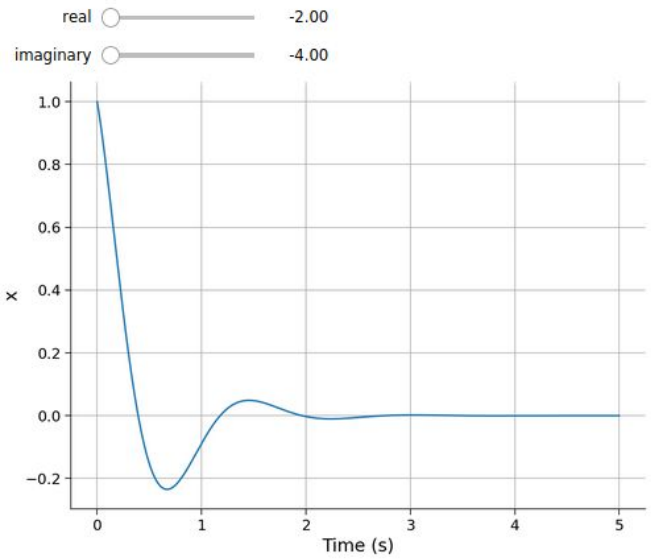
For small-ish dt, the solution still looks the same.
As dt gets bigger, the solution starts to look choppier and is no longer smooth, but still has mostly the right trends.
For a = 0.15, as dt gets above 0.7 or so, we start to see the forward Euler integration start to actually break down. Specifically, the solution is no longer decreasing monotonically and has developed an erroneous dip below zero.

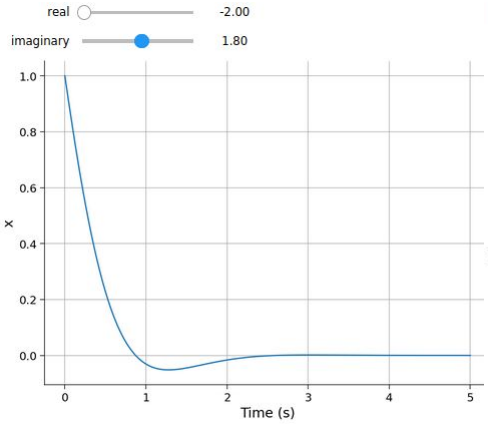
Generally for each system, there is a dt threshold above which the simulation introduces numerical artifacts and no longer behaves as an accurate estimate of the true underlying system. We may tolerate some choppiness in the solution, but eventually qualitatively wrong things creep in!

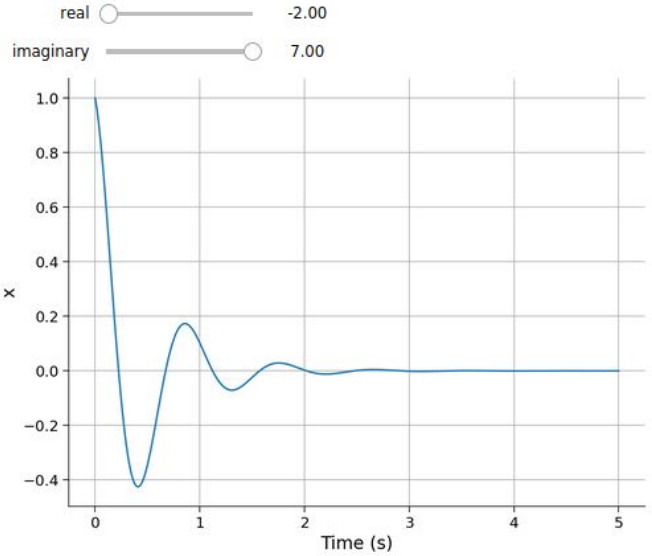
Oscillatory dynamics

explore what happens when a is a complex number and has a non-zero imaginary component.

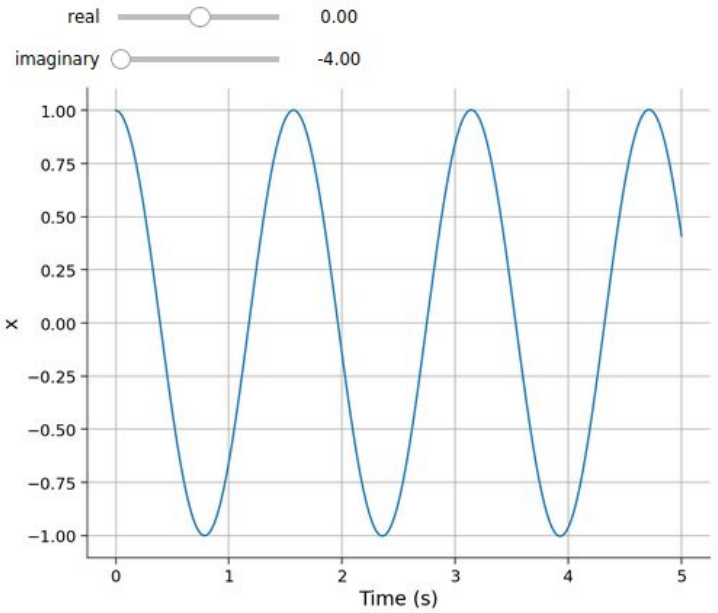
Influence of Imaginary part

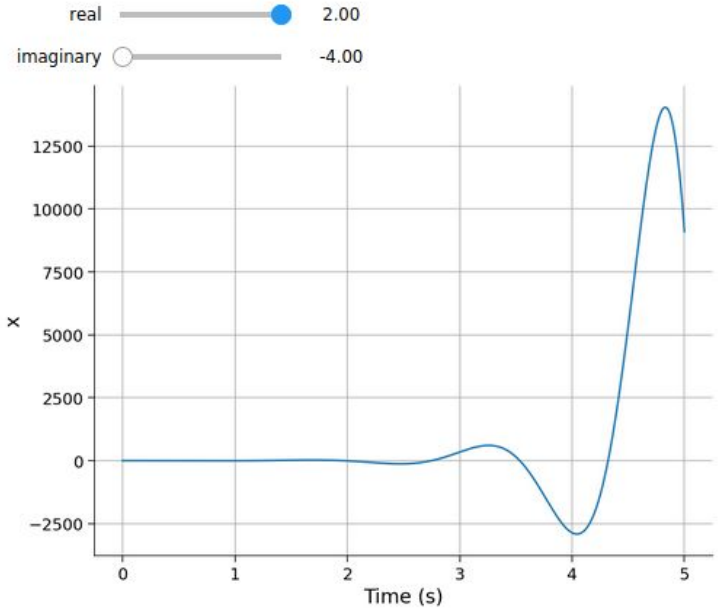




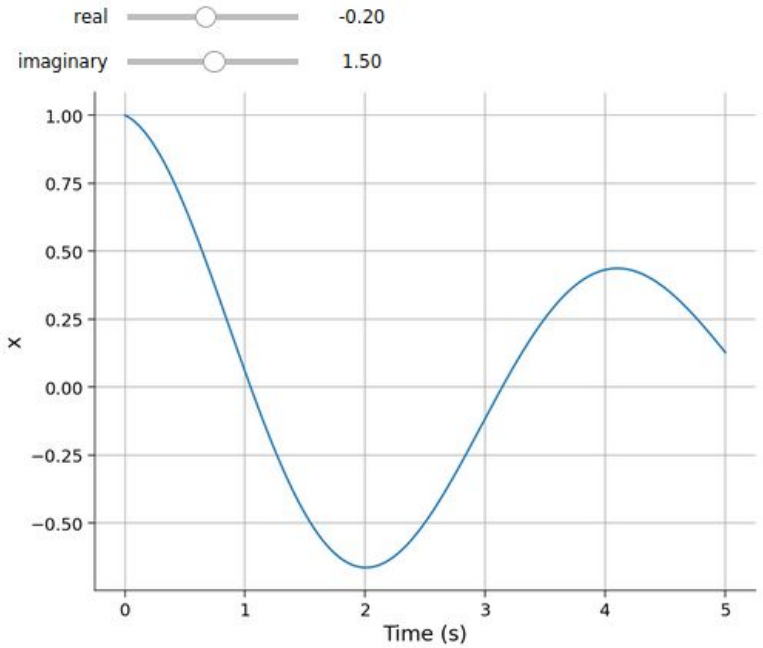


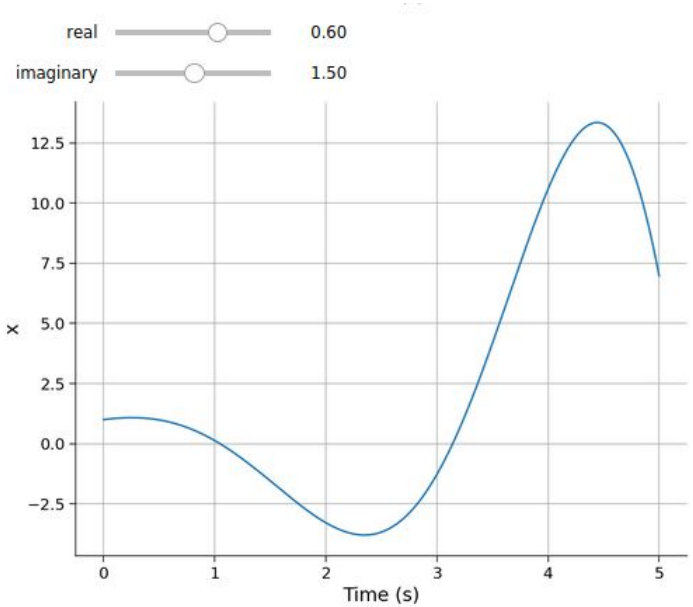
Influence of real part





Real and imaginary grow together





Oscillation, Stability and Growth

*What values of a produce dynamics that both **oscillate** and **grow**?*

To make the system both oscillate and grow, real has to be positive, and imaginary has to be not zero.

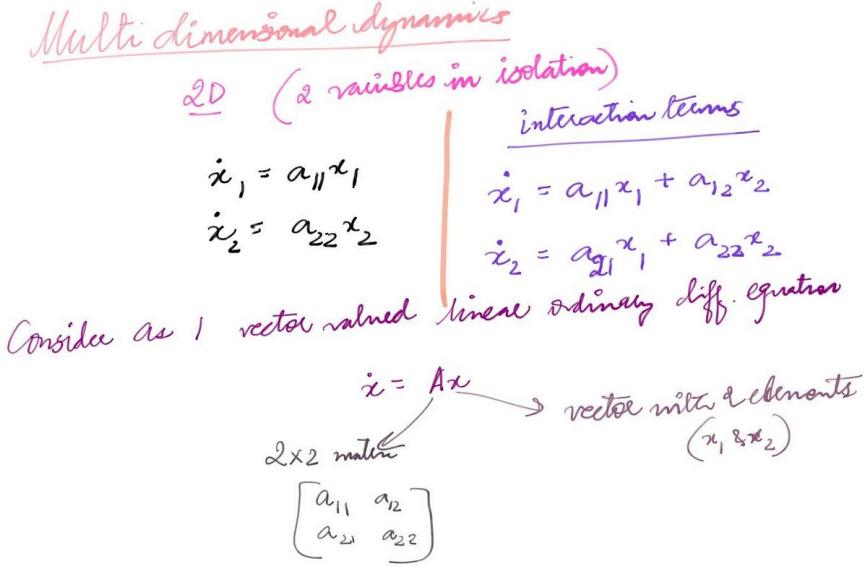
What value of a is needed to produce a stable oscillation of 0.5 Hertz (cycles/time units)?

Stable oscillation of 0.5 Hz (half a cycle per unit time, or one cycle per two unit time) is achieved with real = 0 and imaginary = +/- pi (approximately 3.1 or -3.1).

Note: For really large values of the imaginary component, the numerical integration scheme breaks down a bit, and we see non-stable oscillations even when real=0. This is a numerical artifact of the forward Euler scheme.

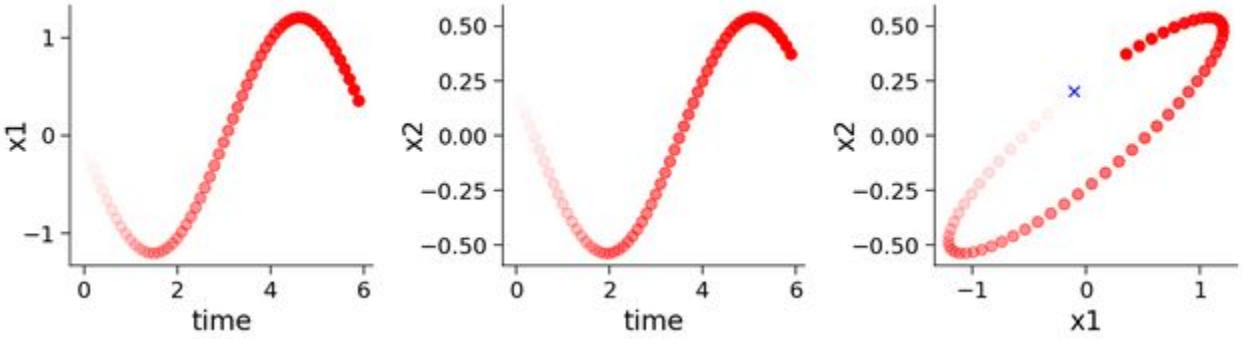
Multidimensional dynamics

Adding one additional variable (or dimension) adds more variety of behaviors. Additional variables are useful in modeling the dynamics of more complex systems with richer behaviors, such as systems of multiple neurons.



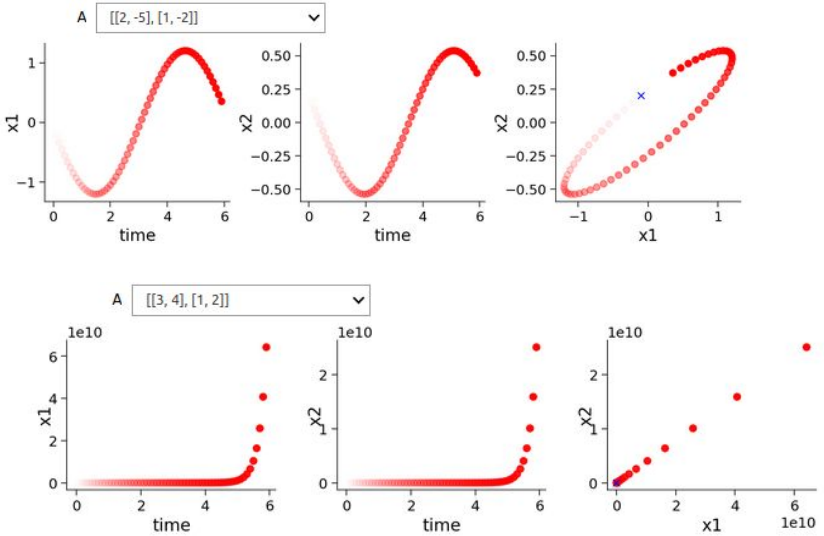
Trajectory simulation

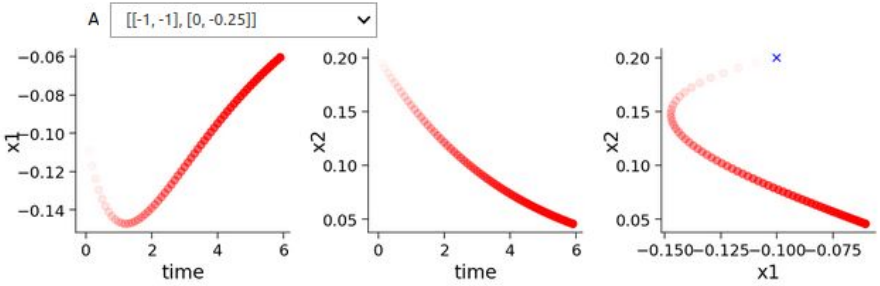
simulate some **trajectories** of a given system and plot how x1 and x2 evolve in time

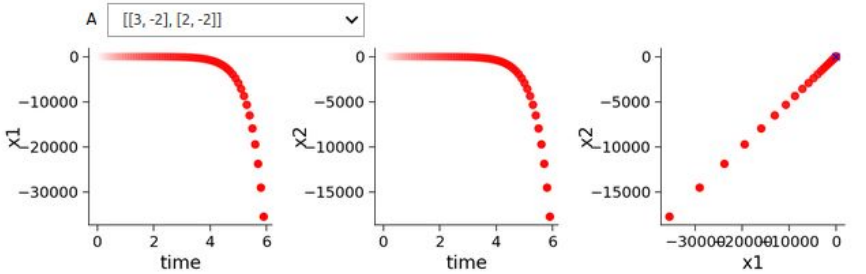


Varying A

What kinds of qualitatively different dynamics do you observe? Hint: Keep an eye on the x-axis and y-axis!





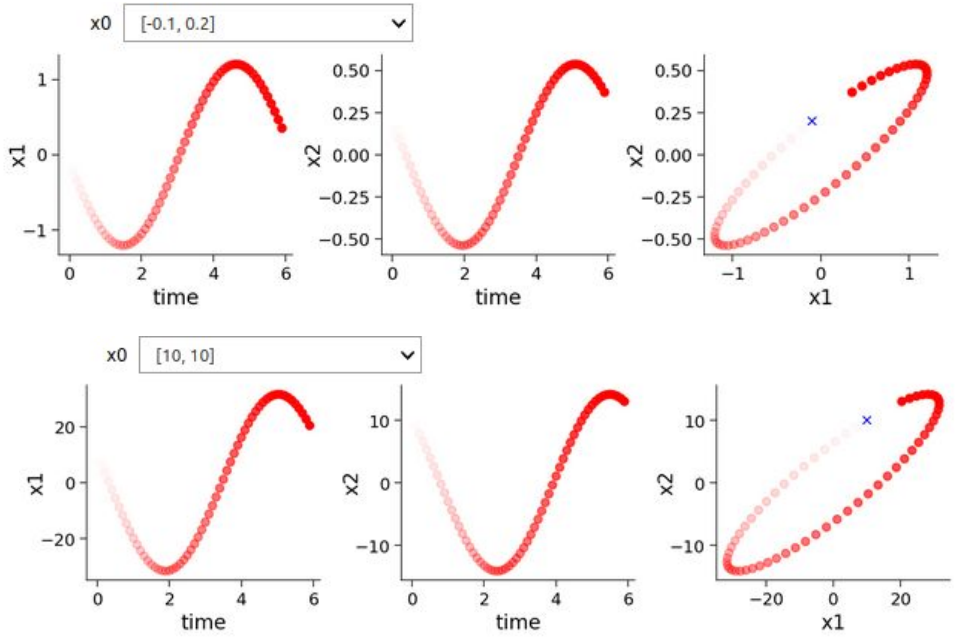


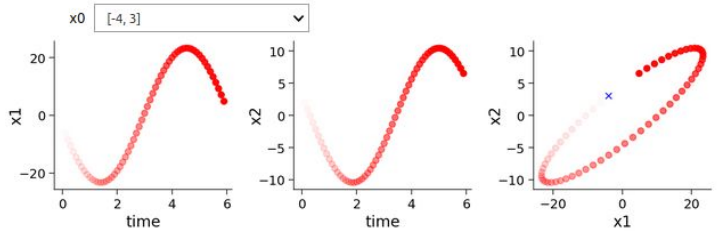
Observations

exponential growth to positive values, exponential growth to negative values, stable oscillations, and decay to the origin.

Changing the initial conditions for A always produces oscillatory dynamics. The only difference is the radii of the resulting elliptical trajectories.

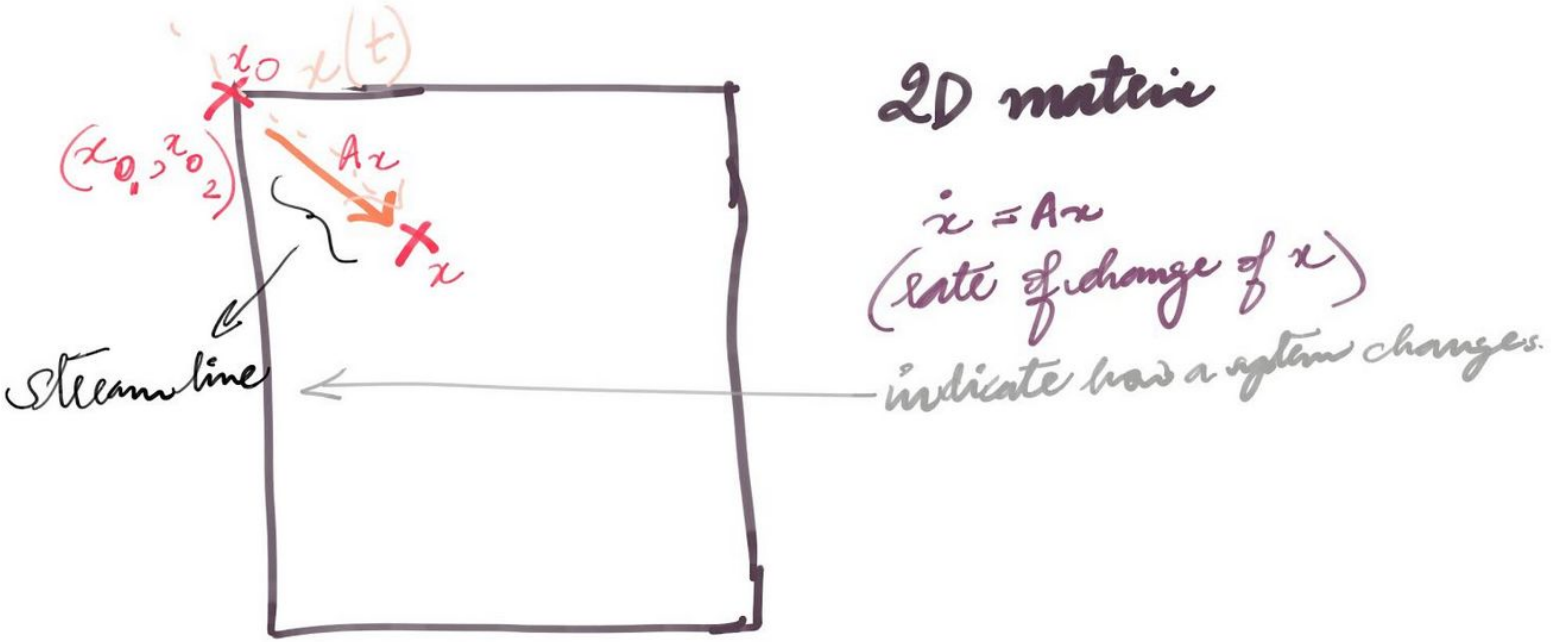
Varying initial conditions



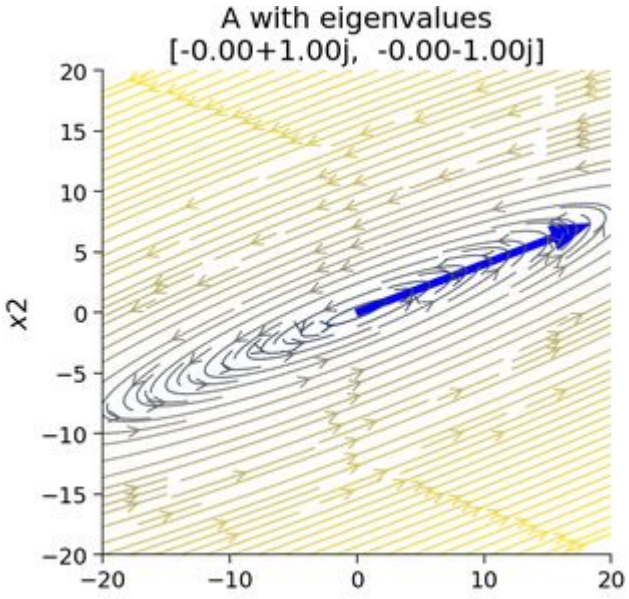


Stream Plots

Stream plot: grid of initial conditions that affect trajectories of a system,

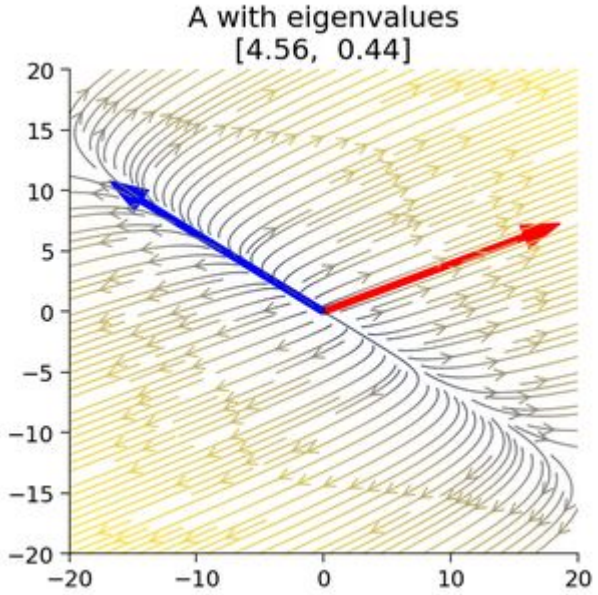


A with imaginary eigenvalues



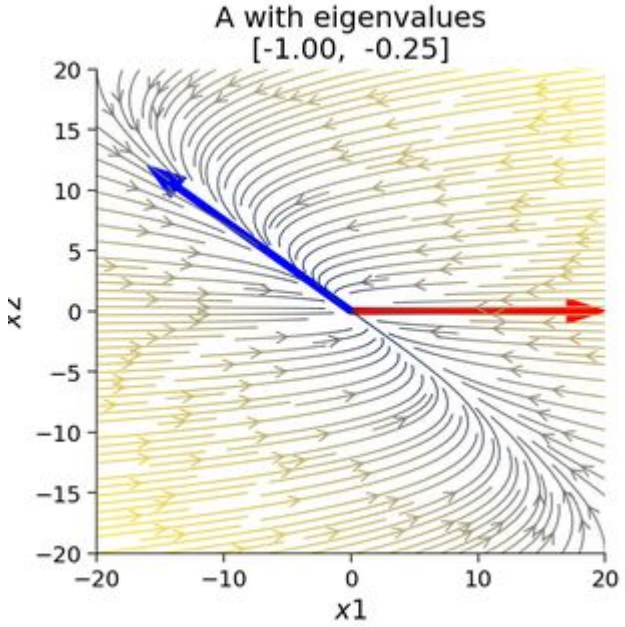
both eigenvalues are imaginary (no real component, the two eigenvalues are complex conjugate pairs), so the solutions are all stable oscillations. The eigenvectors are also complex conjugate pairs (plotted on top of each other). They point in the direction of the major axis of the elliptical trajectories.

A with positive eigenvalues



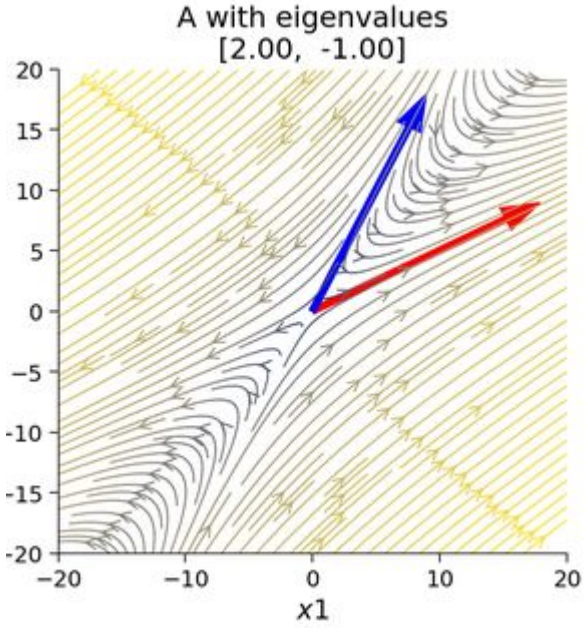
both eigenvalues are positive, so they are growing. The larger eigenvalue direction (red) grows faster than the other direction (blue), so trajectories all eventually follow the red eigenvector direction. Those thatstart close to the blue direction follow blue for a bit initially.

A with negative eigenvalues



both eigenvalues are negative, so they are both decaying. All solutions decay towards the origin [0, 0]. The red eigenvalue is larger in magnitude, so decay is faster along the red eigenvector.

A with + and - eigenvalues



one eigenvalue is positive (red) and one eigenvalue is negative (blue). This makes the shape of the landscape the shape of a saddle (named after the saddle that one puts on a horse for a rider). Trajectories decay along the blue eigenvector but grow along the red eigenvector.

Principal eigenvector points - Discussion

What is special about the direction in which the principal eigenvector points? And how does the stability of the system relate to the corresponding eigenvalues?

for matrices with real eigenvalues, the eigenvectors indicate the lines on which Ax is parallel to x

and real eigenvalues indicate the factor by which Ax is streched or shrunk compared to x

Summary

- simulating the trajectory of a dynamical system specified by a differential equation $\dot{x}=f(x)$ using a forward Euler integration scheme.
- The behavior of a one-dimensional linear dynamical system $\dot{x}=ax$ is determined by a, which may be a complex valued number. Knowing a, we know about the stability and oscillatory dynamics of the system.
- The dynamics of high-dimensional linear dynamical systems $\dot{x}=Ax$ can be understood using the same intuitions, where we can summarize the behavior of the trajectories using the eigenvalues and eigenvectors of A

Tutorial #2 Explanations

probabilistic *dynamical systems (Stochasticity)*

In a probabilistic process, elements of randomness are involved. Every time you observe some probabilistic dynamical system, started from the same initial conditions, the outcome will likely be different. Aka dynamical systems that involve probability will incorporate random variations in their behavior.

Markov process

- present state entirely determines the transition to next state

Markovian

- differential equations express a relationship between $\dot{x}$ & x @ every time t

$\Rightarrow$ direction of x @ every time depends entirely on the value of x.

$\Rightarrow$ knowledge of value of state vaibles x @ time t is all the information needed to determine $\dot{x}$ & therefore x @ next time!

AGENDA

consider Markov process where the state transitions are probabilistic.

- Understand Markov processes and history dependence.
- Explore the behavior of a two-state telegraph process and understand how its equilibrium distribution is dependent on its parameters.

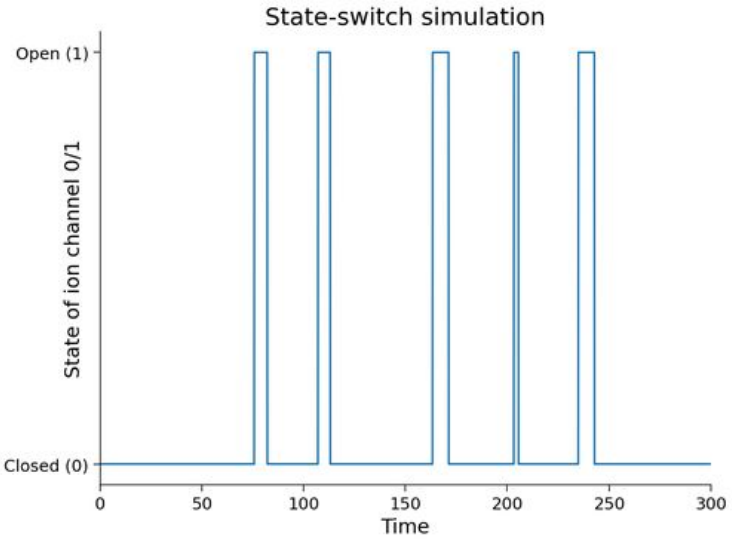
Poisson process

Simulate the process of changing states as a **Poisson process.** The Poisson process is a way to model discrete events where the average time between event occurrences is known but the exact time of some event is not known. Importantly, the Poisson process dictates:

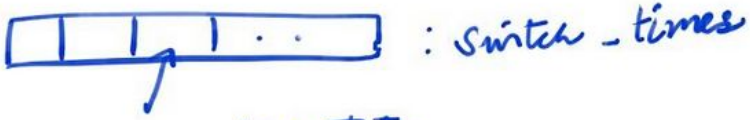
1. The probability of some event occurring is *independent from all other events*.
2. The average rate of events within a given time period is constant.
3. Two events cannot occur at the same moment. Our ion channel can either be in an open or closed state, but not both simultaneously.

Poisson process : models the state of our ion channel at all points t within the total simulation time T. *By* simulating the state change process, we also track at which times throughout the simulation the state makes a switch and use those times to measure the distribution of the time *intervals* between state switches.

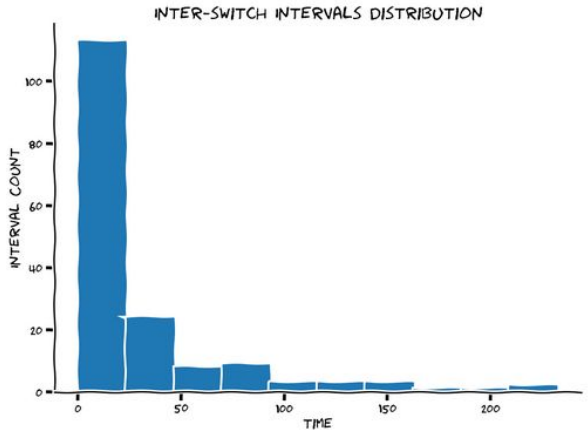
State change simulation



Inter-switch intervals distribution



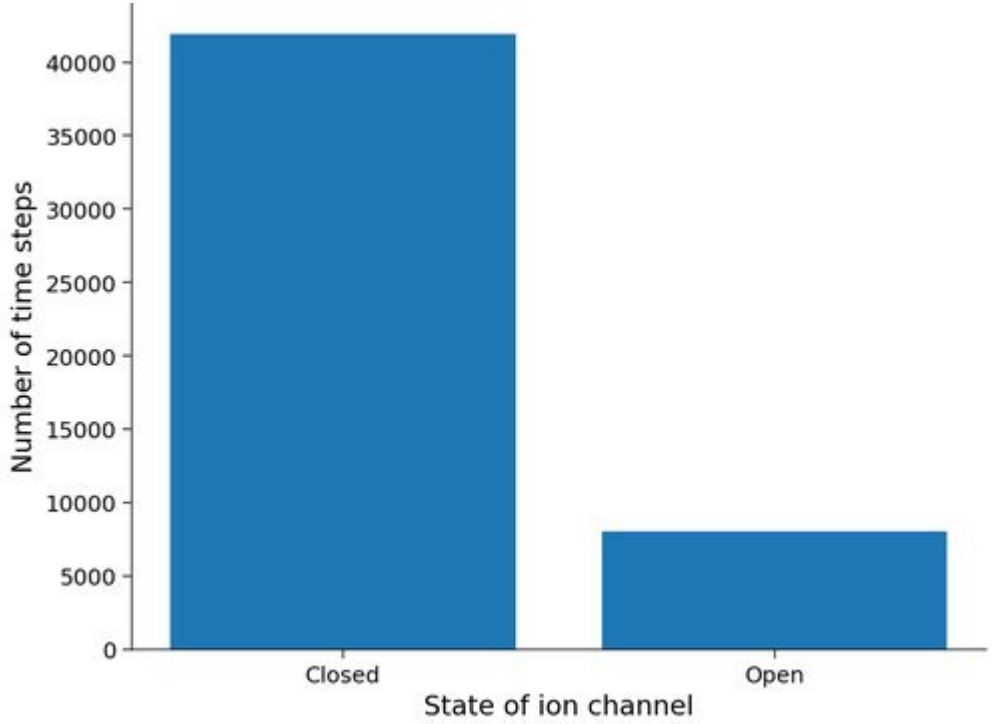
time intervals between each state switch = inter_switch_intervals



Distribution of time spent in each state.

Even though the state is discrete--the ion channel can only be either Closed or Open--we can still look at the ***mean state*** *of the system, averaged over some window of time.*

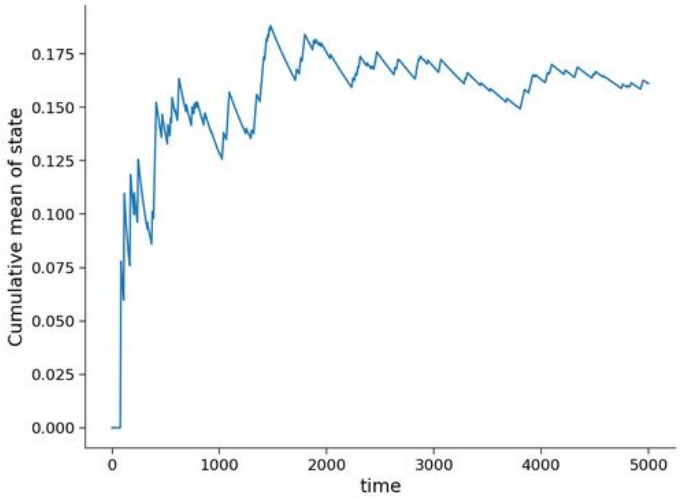
the mean of x over some window of time has the interpretation of ***fraction of time channel is Open****.*



Cumulative mean of state

fraction of Open states as a cumulative mean of the state x: The cumulative mean tells us the average number of state-changes that the system will have undergone after a certain amount of time

although the channel started in the Closed (x=0) state, gradually adopted some mean value after some time. This mean value is related to the transition probabilities μ_{c2o} and μ_{o2c}.



Varying transition probability values & T

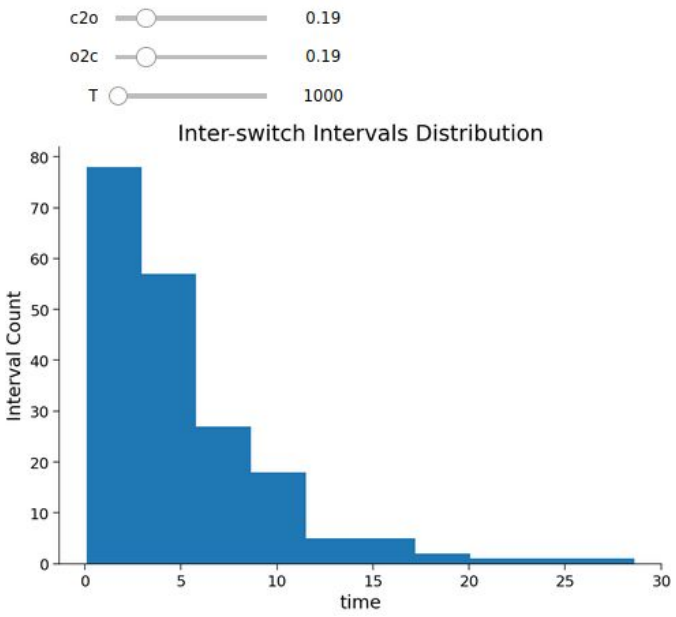
Explore the state-switch simulation for different transition probability values of states μc2o and μo2c. Also, try different values for total simulation time length T.

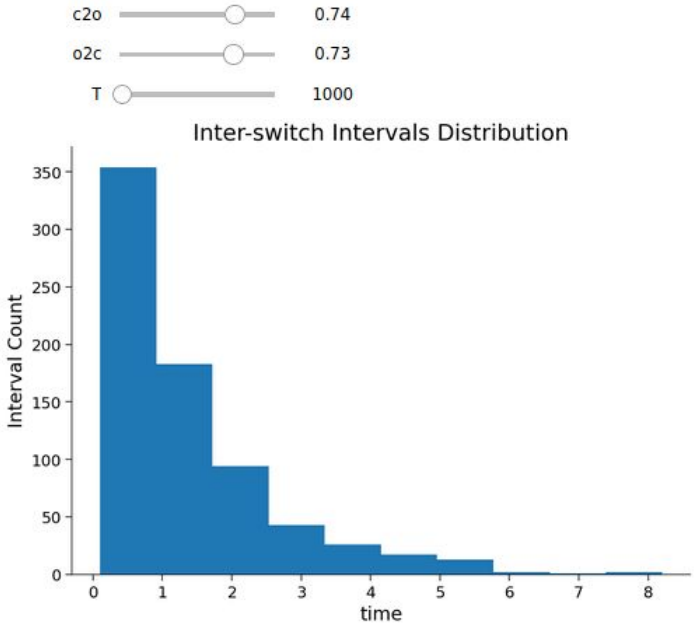
Does the general shape of the inter-switch interval distribution change or does it stay relatively the same? How does the bar graph of system states change based on these values?

The shape of the distribution remains the same, but larger values of either c2o or o2c shifts the distribution towards shorter intervals.

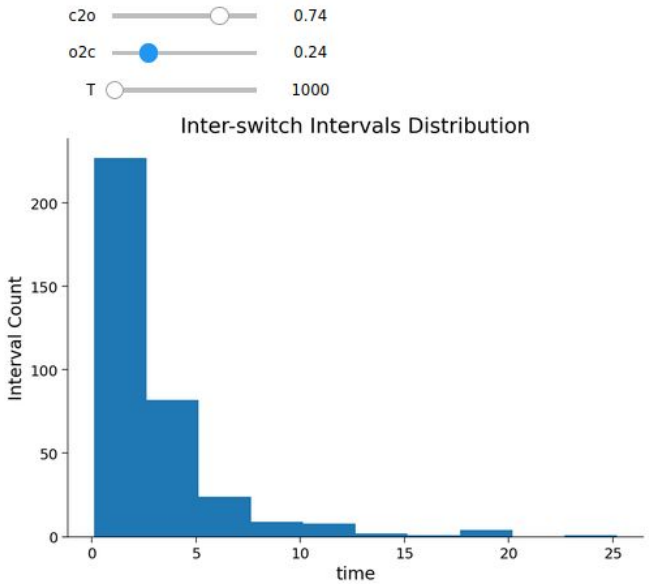
If c2o is larger than o2c, then the channel tends to be open a larger fraction of the time.

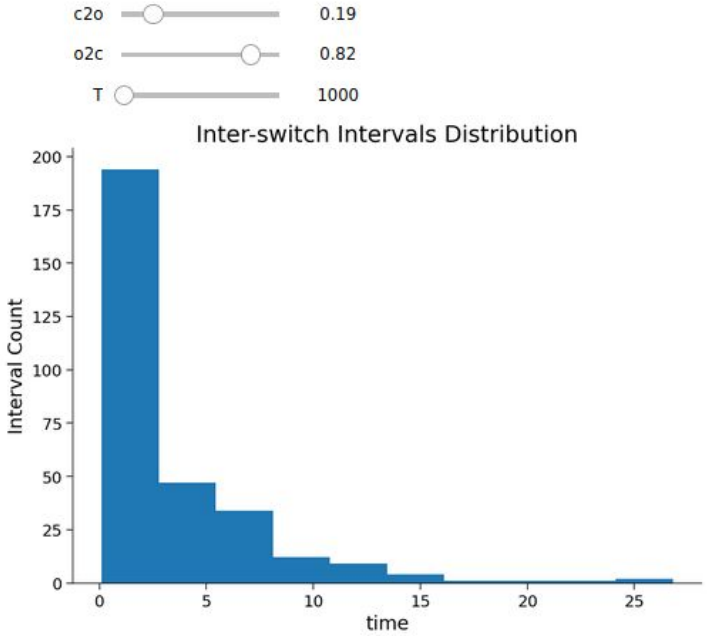
Interswitch interval distribution #1





Interswitch interval distribution # 2



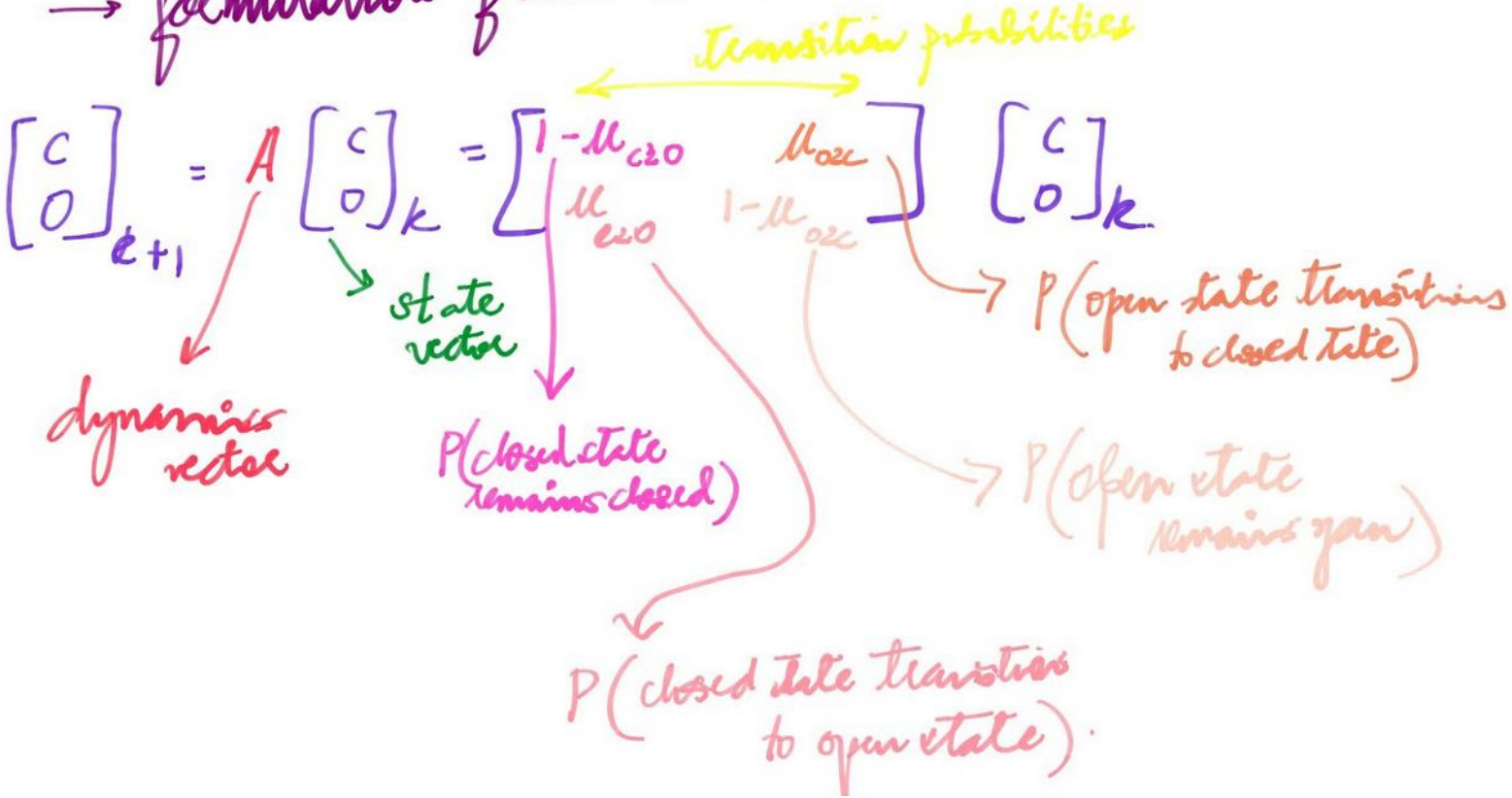


Distributional perspective

Gather empirical distributions of open/closed states/we can formulate the exact same system probabilistically, keeping track of the probability of being in each state written as a discrete step in time, and **A** describes the transition, mapping the state from step k to step $k+1$.

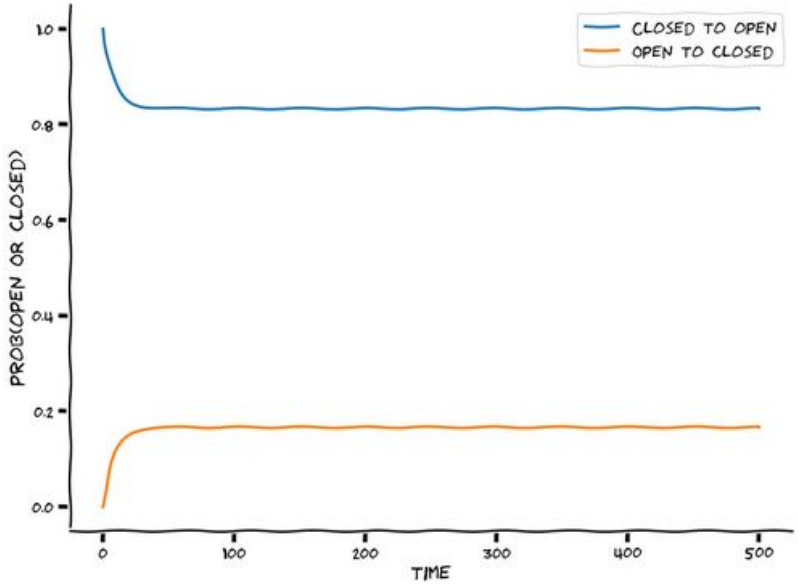
System of transitions

→ formulation of state transition matrix



Probability Propagation

simulate the propagation of probabilities of closed/open of the ion channel through time.



Observations

Simulate the propagation of probabilities of the ion channel's state changing through time.

Using this method is useful in that we can **run the simulation once** and see **how the probabilities propagate throughout time**, rather than re-running and empirically observing the telegraph simulation over and over again.

Although the system started initially in the Closed ($x=0$) state, over time, it settles into a equilibrium distribution where we can predict what fraction of time it is Open as a function of the μ parameters (relaxation towards equilibrium).

Equilibrium of the telegraph process

Connect the behavior of the system at equilibrium with the eigen decomposition of A. Eigenvalues of A tell us about the stability of the system, specifically in the directions of the corresponding eigenvectors.

Eigenvalues: [1. 0.988]

Eigenvector 1: [0.98058068 0.19611614]

Eigenvector 2: [-0.70710678 0.70710678]

Food for thought

Which of these eigenvalues corresponds to the **stable** (equilibrium) solution? What is the eigenvector of this eigenvalue? How does that explain the equilibrium solutions in simulation in Section 2 of this tutorial?

hint: Rescale the elements of the eigenvector such that they also sum to 1 so, these can then be directly compared with the probabilities of the states in the simulation.

Whichever eigenvalue is 1 is the stable solution. There should be another eigenvalue that is <1, which means it is decaying and goes away after the transient period.

The eigenvector corresponding to this eigenvalue is the stable solution. Normalize this eigenvector so that its 2 elements sum to one, then we would see that the two numbers correspond to [P(open), P(closed)) at equilibrium.

Results

```
# whichever eigenvalue is 1, the other one makes no sense
print(eigenvector1 / eigenvector1.sum()) #Result: [0.83333333
0.16666667]

print(eigenvector2 / eigenvector2.sum()) #Result:
[-1.06150861e+15  1.06150861e+15]
```

Summary

- The definition of a Markov process with history dependence.
- The behavior of a simple 2-state Markov process--the telegraph process--can be simulated either as a state-change simulation or as a propagation of probability distributions.
- The relationship between the stability analysis of a dynamical system expressed either in continuous or discrete time.
- The equilibrium behavior of a telegraph process is predictable and can be understood using the same strategy as for deterministic systems: by taking the eigen decomposition of the A matrix.

Tutorial #3
Explanations

Objectives

*Behavior of such systems when their trajectories are (1) entirely predictable and deterministic, or (2) governed by random processes, it's time to consider that neither is sufficient to describe neuroscience. Instead, we are often faced with processes for which we know some dynamics, but there is some random aspect as well (**dynamical systems with stochasticity**).*

deterministic and stochastic processes can both be a part of a dynamical system by:

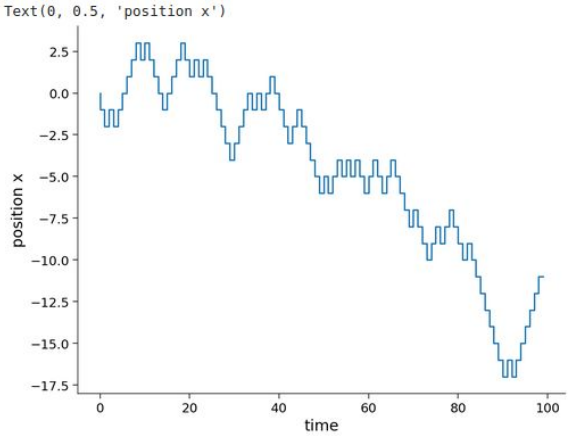
- *Simulating random walks*
- *Investigating the mean and variance of a Ornstein-Uhlenbeck (OU) process*
- *Quantifying the OU process's behavior at equilibrium.*

Random walk

Behavior of the E. coli bacterium: capable of navigating odor gradients on a substrate to seek a food source. Larger life (including flies, dogs, and blindfolded humans) sometimes use the same strategies to guide their decisions.

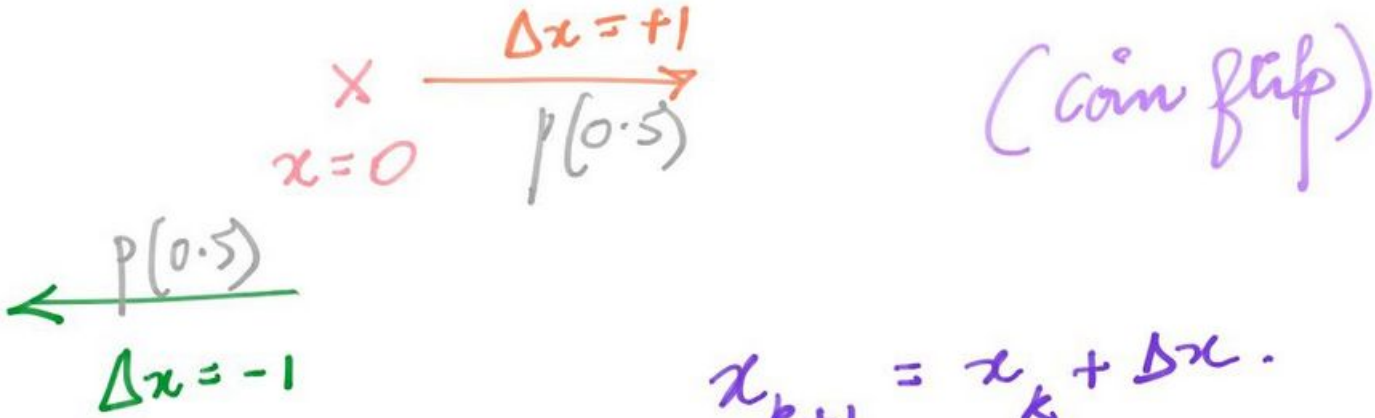
But what is the best strategy when one does not know where to head? Why, flail around randomly, of course!

The **random walk** is exactly that — at every time step, use a random process to change one's heading accordingly. Note that this process is closely related to Brownian motion.



One dimensional random Walk

time step



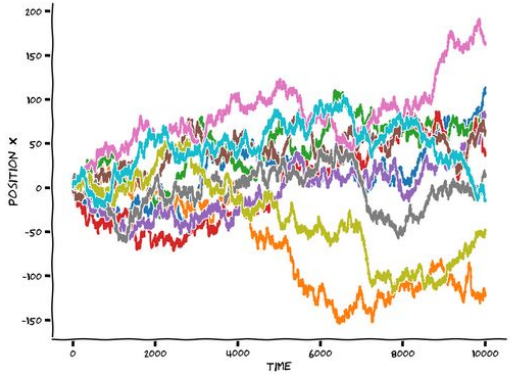
(coin flip)

$$x_{k+1} = x_k + \Delta x.$$

Improvised random walk

Assumed the bacterium takes a step of size 1 at every point in time. Let's let it take steps of different sizes!

Random walk where the steps have a standard normal distribution (with mean μ and standard deviation σ). Instead of running one trajectory at a time, we can simulate a large number of trajectories efficiently with random_walk_simulator generating N random walks each with T time points efficiently.

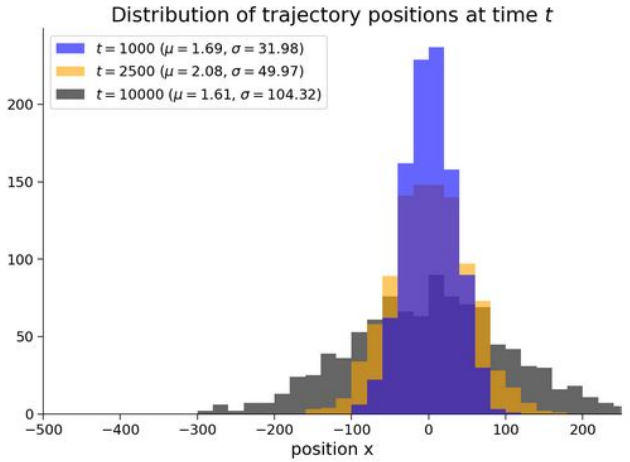


Trajectory analysis

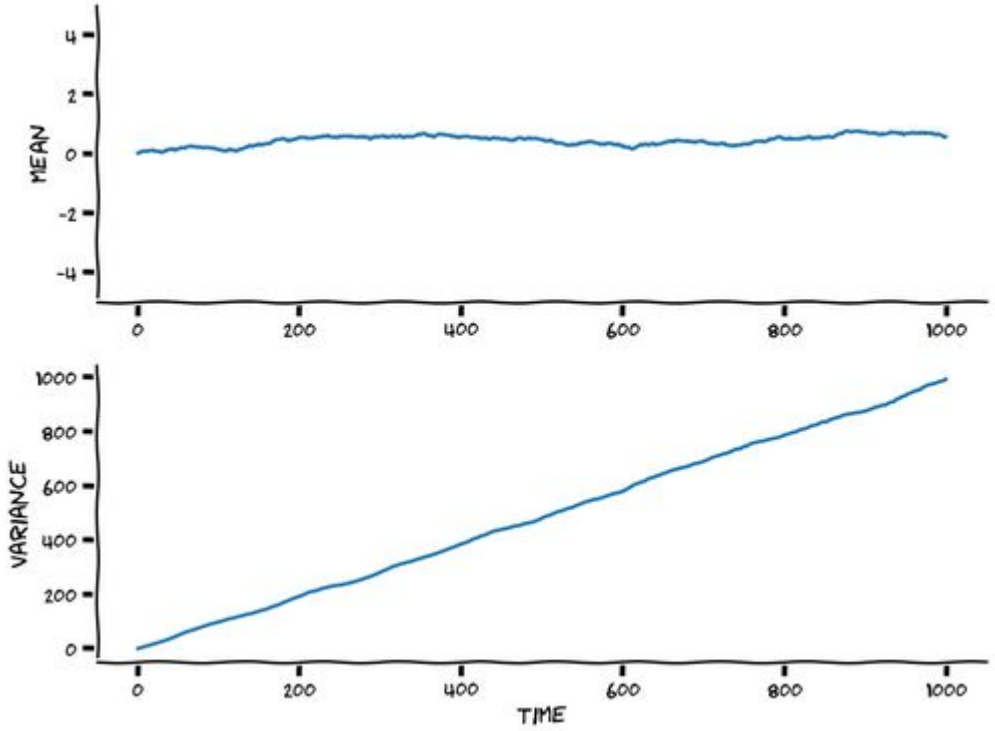
At the beginning almost all trajectories are very close to $x=0$ which is where our bacterium started. As time progresses, some trajectories move further and further away from the starting point. However, a lot of trajectories stay close to the starting point of $x=0$

Diffusive processes

At the beginning of the simulation, the distribution of positions is sharply peaked about 0. As time progresses, the distribution becomes wider but its center stays closer to 0. In other words, the mean of the distribution is independent of time, but the variance and standard deviation of the distribution scale with time. Such a process is called a **diffusive process.**



Random Walk mean and variance

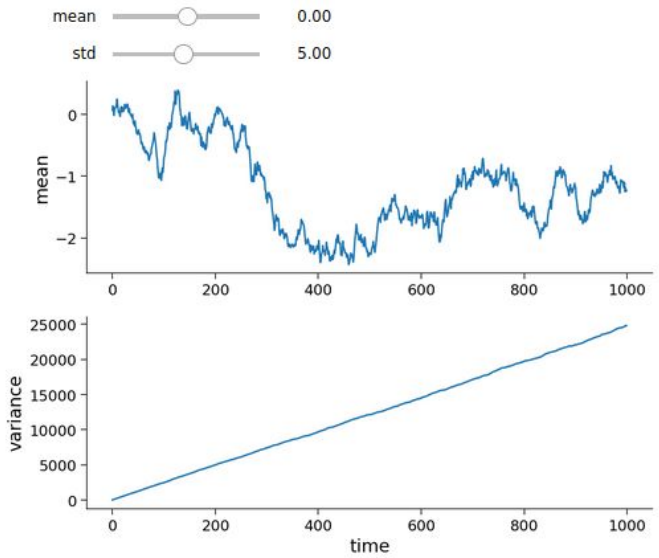


The expected value of x stays close to 0, even for random walks of very long time.

The variance, on the other hand, clearly increases with time. In fact, the variance seems to increase linearly with time!

Influence of Parameter Choice

How do the parameters μ and σ of the Gaussian distribution from which we choose the steps affect the mean and variance of the bacterium's random walk?



When mu is 0, the solutions remain around 0, regardless of sigma.

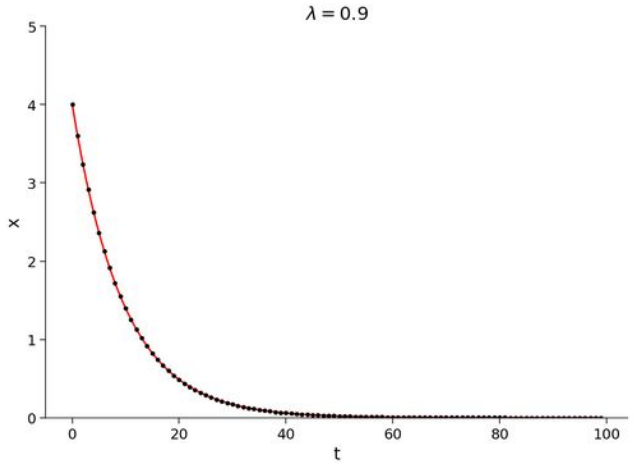
When mu is not 0, the mean of the solutions drift toward positive or negative, proportional to the magnitude of mu.

The variance of the solutions increases linearly in time. The slope of the increase is proportional to the std of the Gaussian distribution at each step.

drift-diffusion model (DDM)

The random walk process is diffusive, and the distribution of possible trajectories spreads, taking on increasing variance with time. Even so, at least in one dimension, the mean remains close to the initial value (0).

Our goal is to build on this model to construct a ***drift-diffusion*** *model (DDM).*



Random walk

Combine random walk with first diff. equation

Continuous time : $\dot{x} = ax$

discrete version : $x_{k+1} = \lambda x_k$.

Solution : $x_k = x_0 \lambda^k$.

x_0 → value of x @ $t = 0$

deterministic difference equation

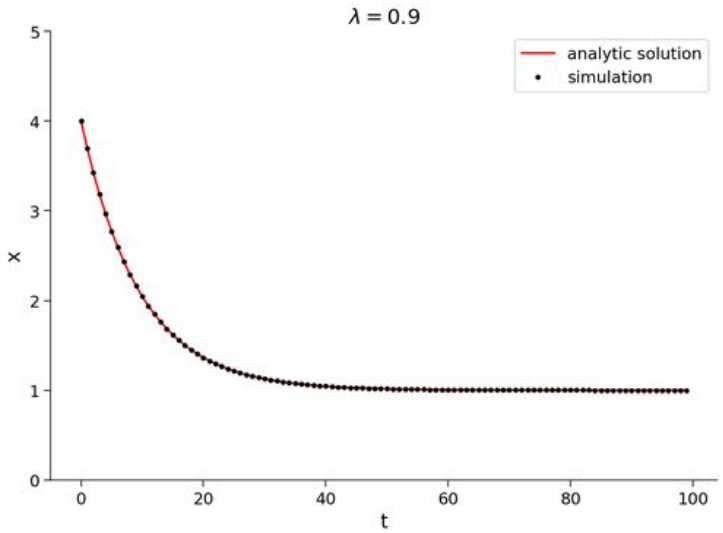
process decays

position $x=0$ — $+x_\infty$ → any other position

rate of decay $\propto x - x_\infty$

New system: $x_{k+1} = x_\infty + \lambda(x_k - x_\infty)$

modifying: $x_k = x_\infty(1-\lambda^k) + x_0\lambda^k$



Ornstein uhlenbeck process

The model is a combination of a drift term toward x_∞ and a diffusion term that walks randomly.

deterministic difference equation + diffusion process = drift diffusion model

Ornstein Uhlenbeck (ou) process

$$x_{k+1} = x_0 + \lambda(x_k - x_\infty) + \sigma\eta$$

η: sampled from std normal distribution $\mu = 0 / \sigma = 1$

Determinism +Stochasticity

each step through time has a *deterministic* part plus a *random, diffusive* part that is drawn from from a normal distribution with standard deviation of σ

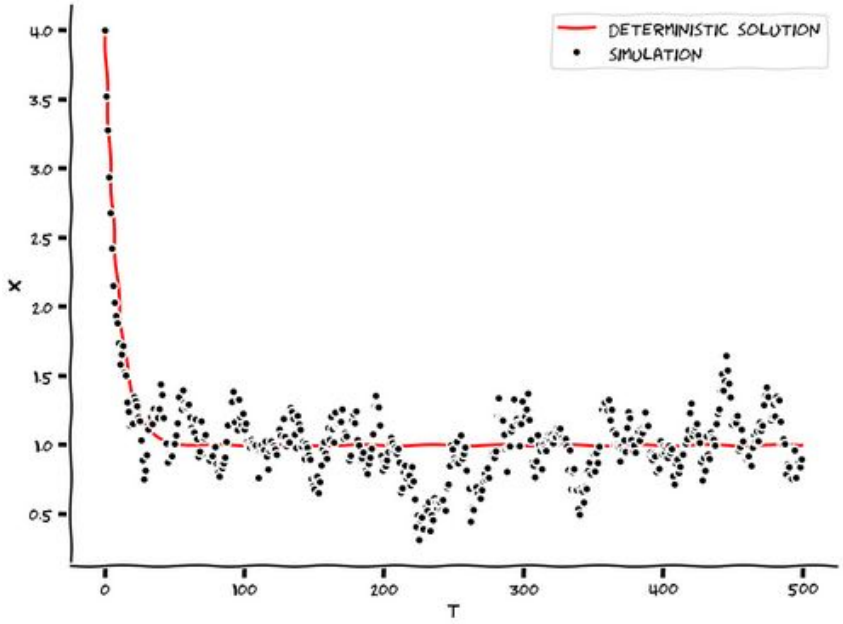
The solution follows the deterministic solution

on average, especially at the beginning.

At the end, it follows the analytic solution

on average, but there's

non-zero variation around this solution.



Variance of OU process

mean of the process follows the solution to the deterministic part of the governing equation

Unlike the random walk, because there's a decay process that "pulls" x back towards x^∞, the variance does not grow without bound with large t. Instead, when it gets far from x^∞, the position of x is restored, until an equilibrium is reached.

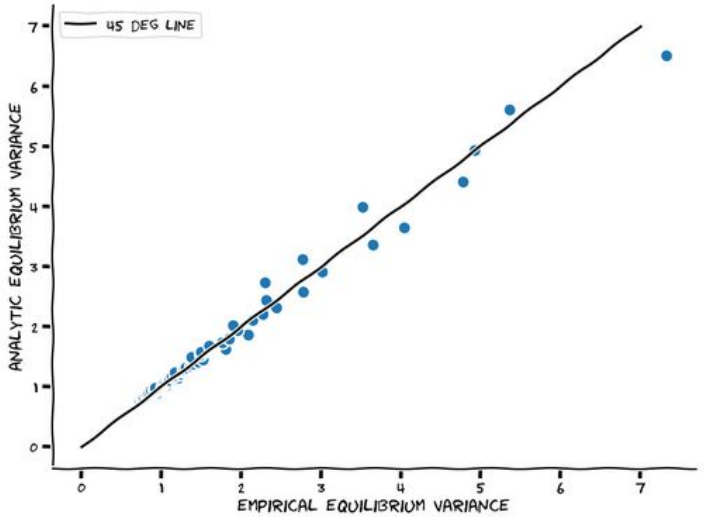
Equilibrium variance for drift diffusion system $\Big\}$ $\text{var} = \dfrac{\sigma^2}{1-\lambda^2}$

value of this equilibrium variance depends on λ and σ. It does not depend on x_0 and x^∞.

Analytical vs empirical variance

compute the analytical variance, and compare against the empirical variances

Observations: They should be about equal to each other and lie close to the 45 degree (y= x) line.



Summary

Importantly, **the interplay between the deterministic and stochastic parts** serve to *balance* the tendency of purely stochastic processes (like the random walk) to increase in variance over time. This behavior is one of the properties of OU systems that make them popular choices for modeling cognitive functions, including short-term memory and decision-making.

Tutorial #4
Explanations

Autoregressive models - Objective

Use the modeling tools and intuitions to fit data.

Instead of generating synthetic data points from a known underlying process, what if we are given data points measured in time and have to learn the underlying process?

drift diffusion (OU) process.

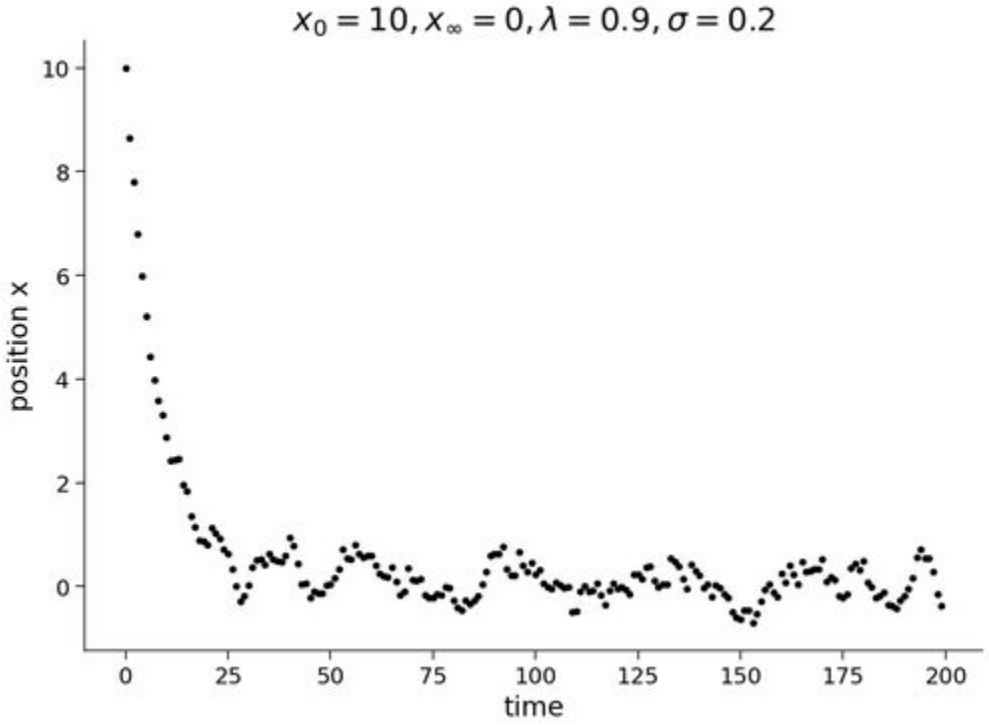
$$x_{k+1} = x_\infty + \lambda(x_k - x_\infty) + \sigma\eta$$

for simplicity $x_\infty = 0$

$$x_{k+1} = \lambda x_k + \eta$$

noise from
normal distribution

Simulating OU process

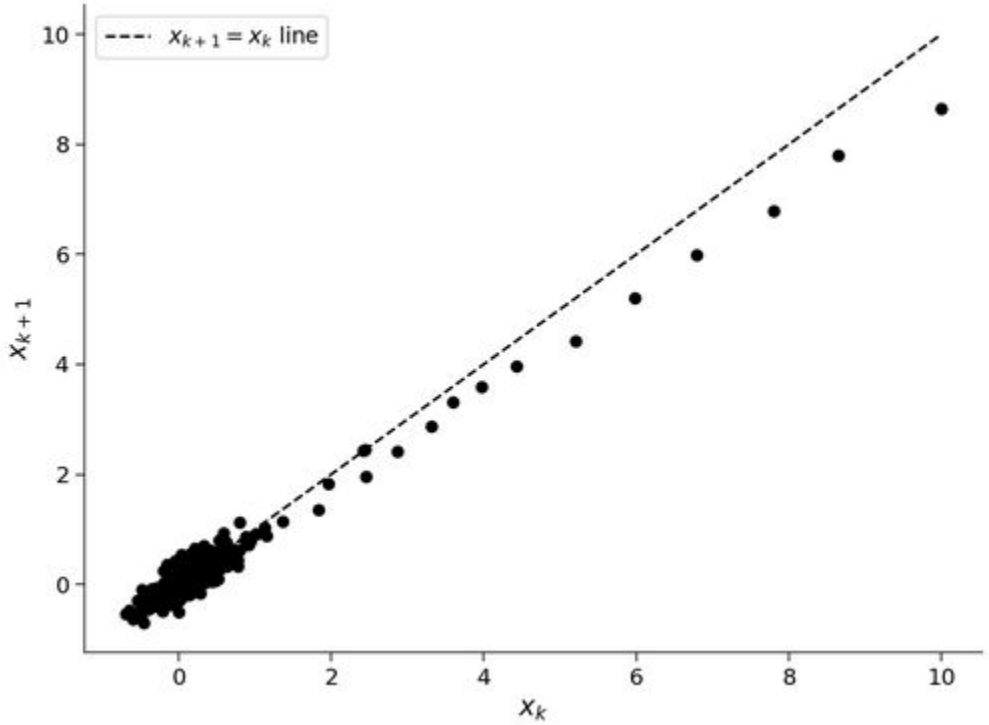


What if we were given these positions x as they evolve in time as data, how would we get back out the dynamics of the system $\boldsymbol{\lambda}$?

Our approach is to solve for $\boldsymbol{\lambda}$ as a regression problem.

X(k) vs. X(k+1)

dynamics that generated the data is ***linear***.



Reformulate task as regression problem

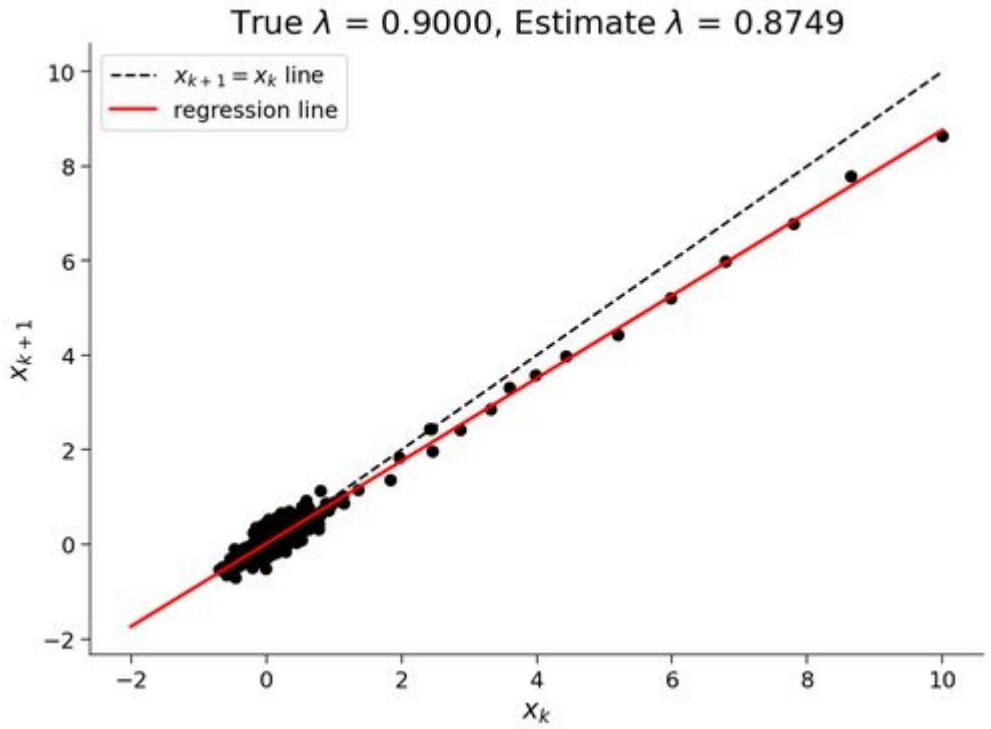
$x_1 = x_{0:T-1}$; $x_2 = x_{1:T}$ } vectors of data indexed (shifted in time by one)

Regression problem

$x_2 = \lambda x_1$ } Autoregressive (Regression of time series on itself from the past)

One step in the past (first order autoregressive model)

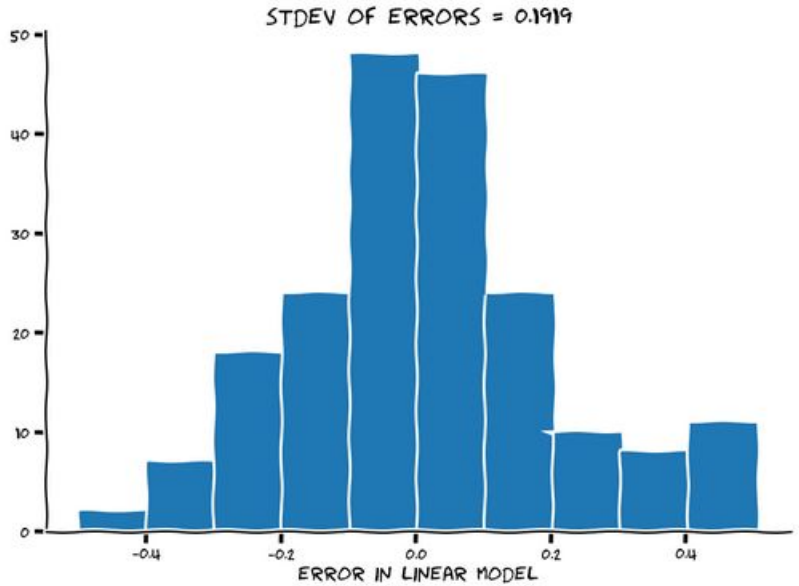
True vs estimate



Residuals of the autoregressive model

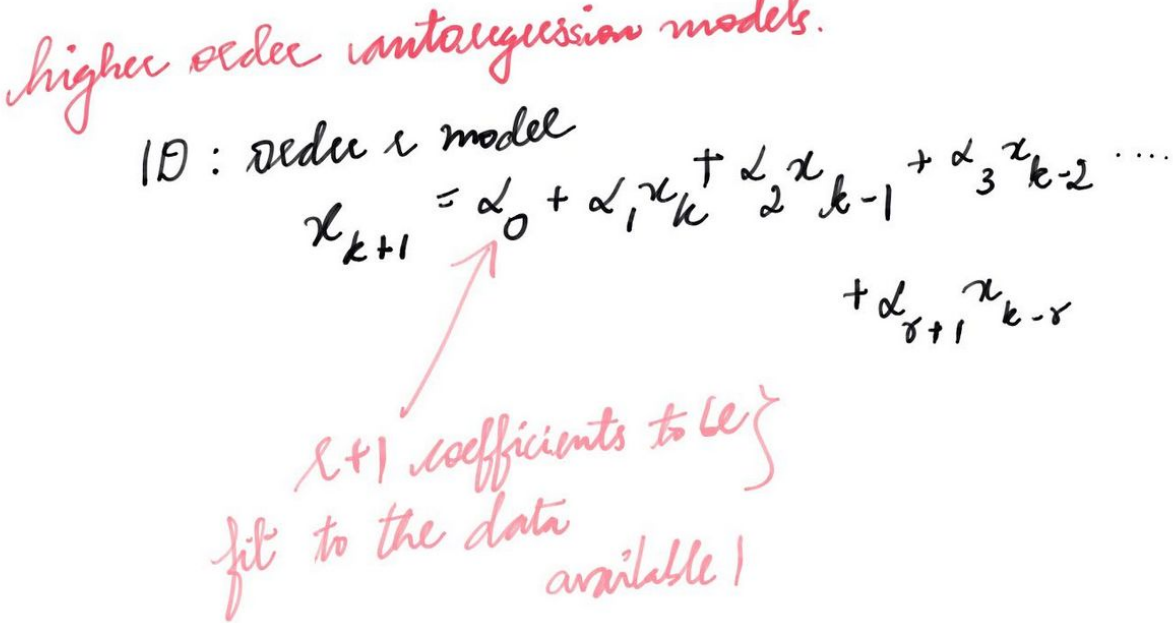
how accurate this one-step prediction might be by plotting the residuals.

Plot a histogram of residuals of our autoregressive model, by taking the difference between the *data* and the *model* prediction



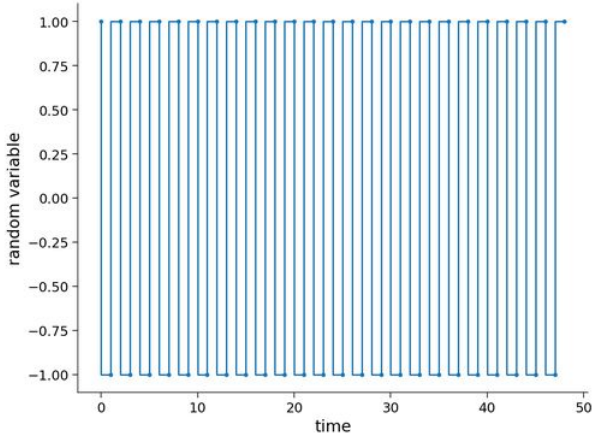
Higher order autoregressive models

generalizing for dependence on data points from the past is straightforward. **Higher order** autoregression models a future time point based on *more than one points in the past.*



entirely predictable sequence

Generate a random Bernoulli sequence as best as you can by typing in 0's and 1's. We will then build higher-order AR models to see if we can identify predictable patterns in the time-history of digits you generate.

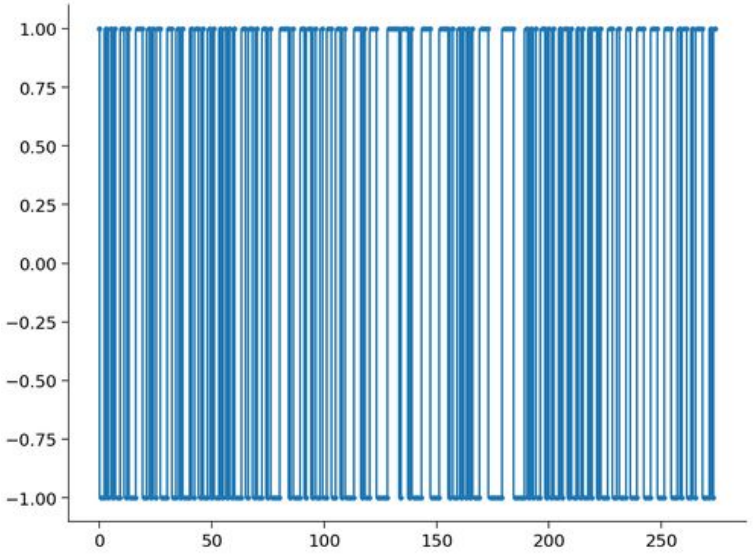


Modeling 0,1 sequence

alpha_0 = -0.00, alpha_1 = -1.00

Truly random sequences of numbers have no structure and should not be predictable by an AR or any other models.

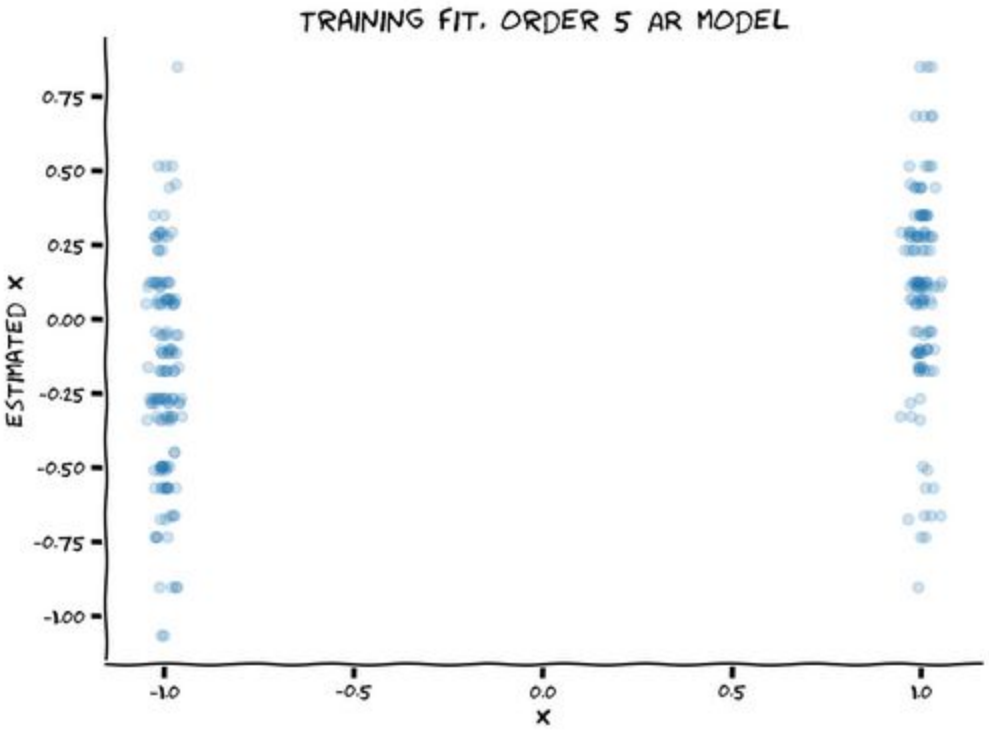
To test out an application of higher-order AR models, let's use them to model a sequence of 0's and 1's that a human tried to produce at random. If the digits really have no structure, then we expect our model to do about as well as guessing, producing an error rate of 0.5.



Fitting AR Sequence

Fit a order-5 (p=5) AR model to the data vector x. We will then plot the observations against the trained model.

Additionally, output from our regression model are continuous (real numbers) whereas our data are scalar (+1/-1). So, we will take the sign of our continuous outputs (+1 if positive and -1 if negative) as our predictions to make them comparable with data. Our error rate will simply be the number of mismatched predictions divided by the total number of predictions.

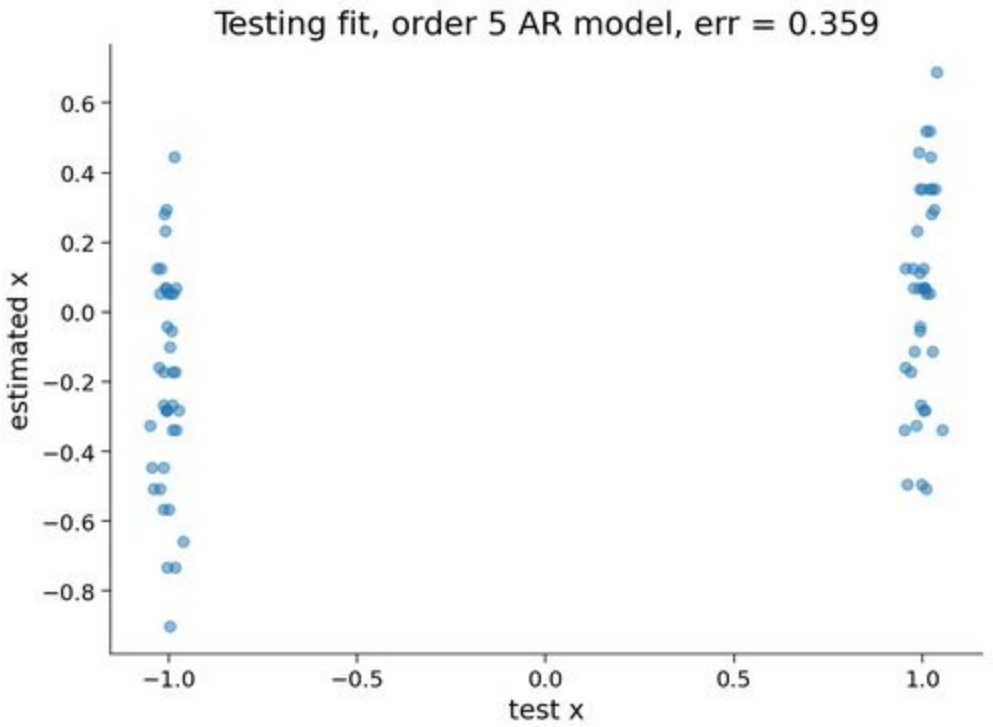

TRAINING FIT, ORDER 5 AR MODEL
ESTIMATED X
0.75
0.50
0.25
0.00
-0.25
-0.50
-0.75
-1.00
-1.0
-0.5
0.0
0.5
1.0
X

AR models

We're getting errors that are smaller than 0.5 (what we would have gotten by chance).

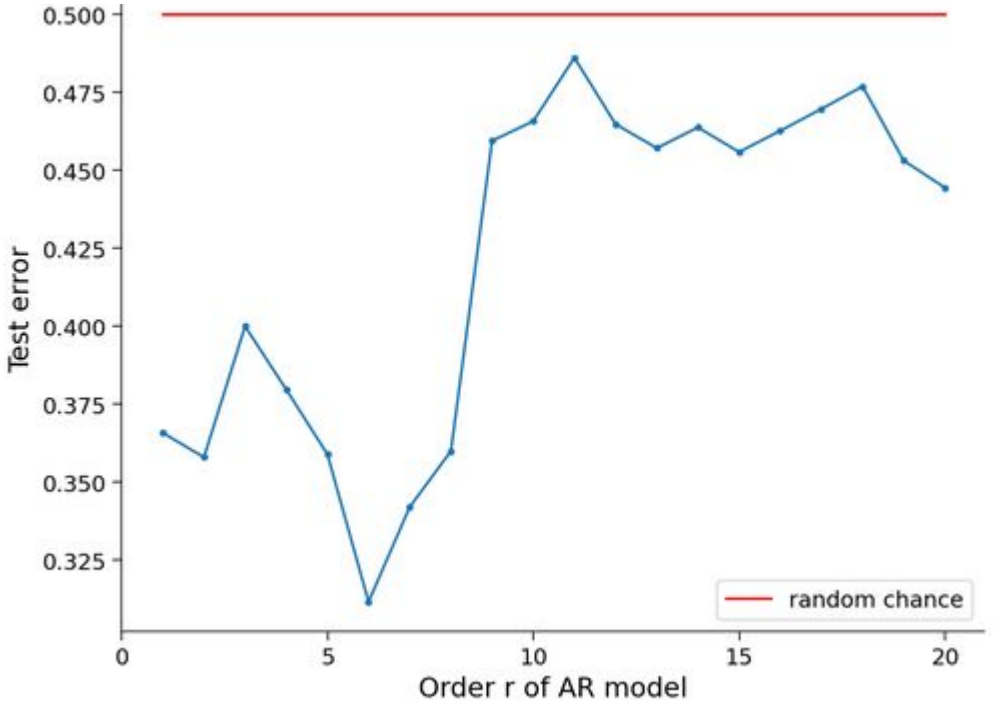
try **AR models of different orders** systematically, and plot the test error of each.

Remember: The model has never seen the test data before, and random guessing would produce an error of 0.5.



AR model - sweetspot

Notice that there's a sweet spot in the test error! The 6th order AR model does a really good job here, and for larger r's, the model starts to overfit the training data and does not do well on the test data.



Summary

- learning the parameters of a linear dynamical system can be formulated as a regression problem from data.
- Time-history dependence can be incorporated into the regression framework as a multiple regression problem.
- Humans are no good at generating random (not predictable) sequences.