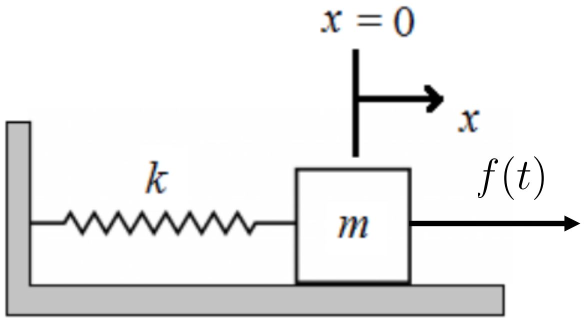
AM 114 (Fall 2025)

Introduction to Dynamical Systems

Qi Gong, Professor
Department of Applied Mathematics
Baskin School of Engineering
University of California, Santa Cruz

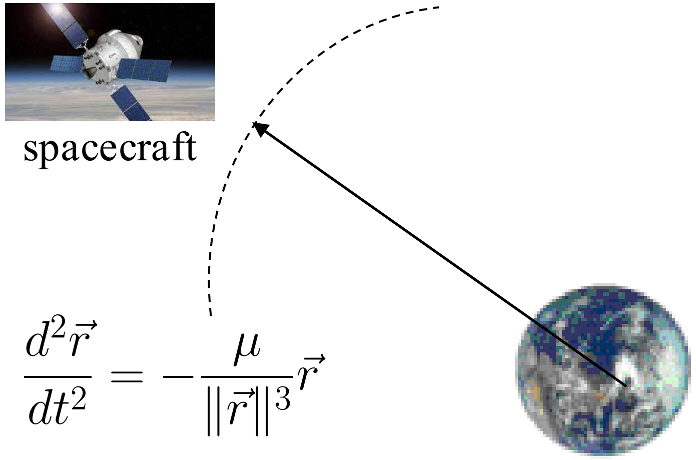
A simple spring-mass system:



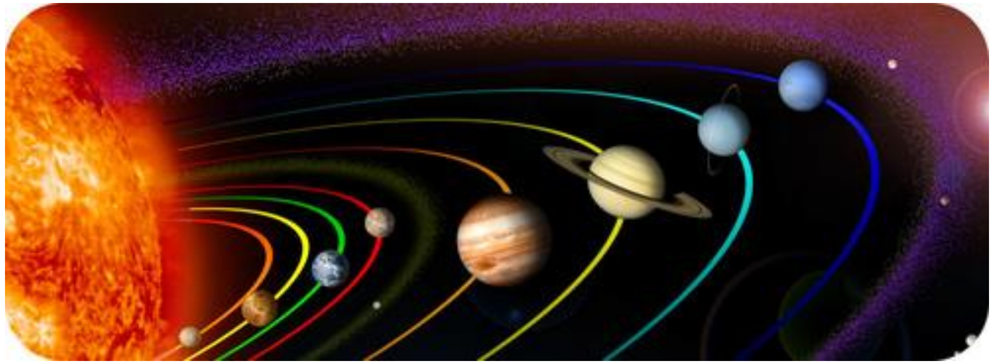
$x(t)$: position, k_s: spring constant, k_f: friction coefficient, $f(t)$: external force

$$F = ma \implies mx'' = f(t) - k_f x' - k_s x \implies mx'' + k_f x' + k_s x = f(t)$$

Newton's law of gravitation



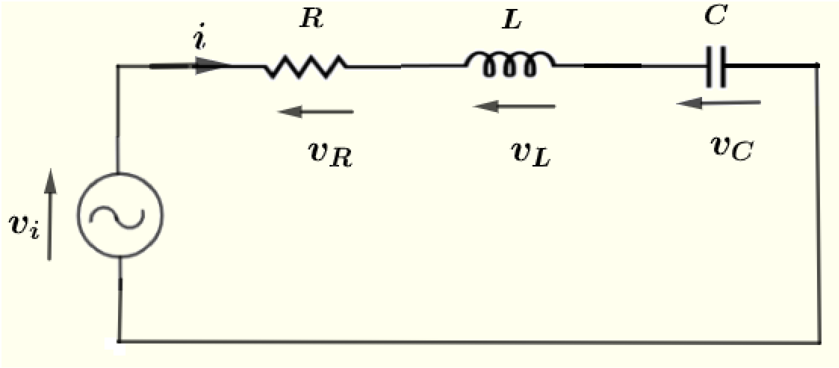
$$\frac{d^2\vec{r}}{dt^2} = -\frac{\mu}{\|\vec{r}\|^3}\vec{r}$$



Planetary motion

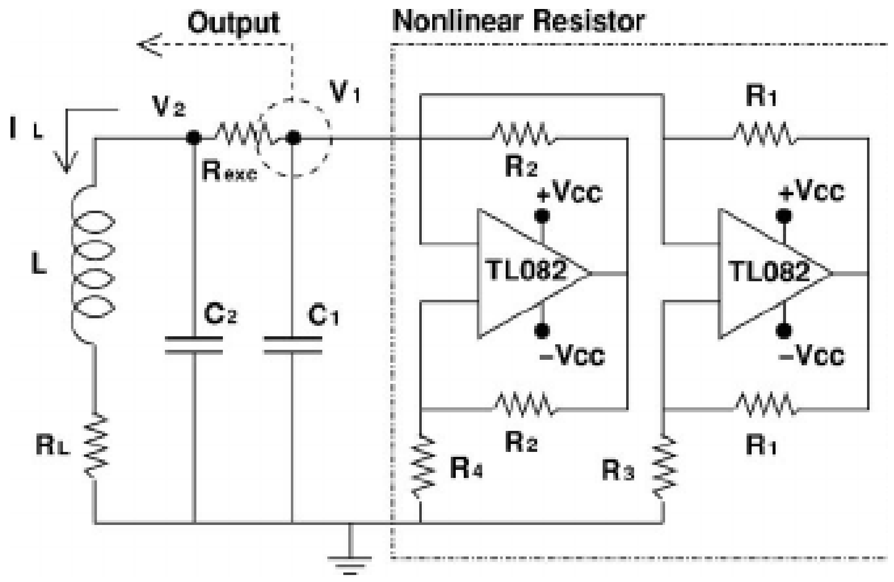
Examples of Dynamical Systems

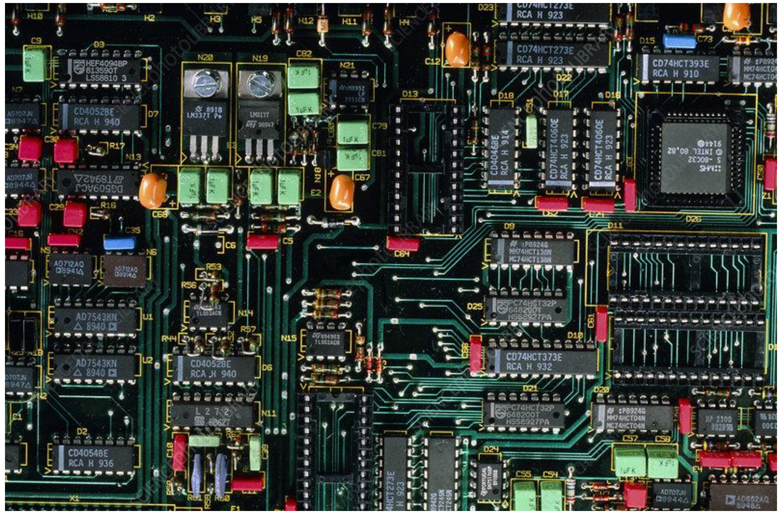
A RLC circuit:



- $V_R = iR$, $V_c = \frac{Q}{C}$, $V_L = L\frac{di}{dt}$, and $i = \frac{dQ}{dt}$

$$L\frac{di}{dt} + Ri + \frac{Q}{C} = v_i(t) \implies Li'' + Ri' + \frac{1}{C}i = v_i'(t)$$





Examples of Dynamical Systems

The **SIR model** is a classical mathematical model used to study how infectious diseases spread in a population. The population is divided into three compartments:

- S(t): **Susceptible** individuals who have not yet been infected but can catch the disease.
- I(t): **Infected** individuals who are currently infected and can spread the disease.
- R(t): **Recovered (or Removed)** individuals who were infected and have either recovered (and gained immunity) or died (removed from the population).

$$
\begin{aligned}
\frac{dS(t)}{dt} &= -\beta S(t)I(t) \\
\frac{dI(t)}{dt} &= \beta S(t)I(t) - \gamma I(t) \\
\frac{dR(t)}{dt} &= \gamma I(t)
\end{aligned}
$$

$0 \leq \beta \leq 1$: the rate of infection
$0 \leq \gamma \leq 1$: the rate of recovery

Examples of Dynamical Systems — Machine Learning

Generative Machine Learning with Diffusion Models

- Forward process: gradually adds noise to clean data to noise via a **stochastic differential equation** (SDE), e.g.,

$$dx_t = -\tfrac{1}{2}\,\beta(t)\,x_t\,dt + \sqrt{\beta(t)}\,dW_t$$

$$x(t_0) \longrightarrow x(t_f)$$

$$\longleftarrow$$

$$dx_t = \tfrac{1}{2}\,\beta(t)\,x_t\,dt + \beta(t)\nabla_{x_t}\log p_t(x_t)dt + \sqrt{\beta(t)}\,dW_t$$

- $x(t)$: data at time step t.
- W_t: Wiener process injecting noise.
- $\beta(t)$: Noise schedule.
- $\nabla_{x_t}\log p_t(x_t)$: the score function modeled by a neural network. It guilds the reverse dynamics so that $x(t)$ moves toward higher probability regions.
- Source: NVIDIA Developer

- Reverse process: generate realistic samples from pure noise by gradually removing noise using another SDE/ODE integrated backward in time.

Some Terminologies

- Ordinary Differential Equations (ODEs) and Partial Differential Equations (PDEs)
  - ODEs: unknown is a function of a single independent variable. Only ordinary derivatives are involved in the differential equation.
  - PDEs: unknown is a function of several independent variables. Partial derivatives are involved in the differential equation.

$$\frac{\partial^2 u(x,t)}{\partial t^2} = c\frac{\partial^2 u(x,t)}{\partial x^2} \qquad \text{(one-dimensional wave equation)}$$

  - This course focuses only on ODEs.

Some Terminologies

- **Continuous dynamics and discrete dynamics**

  — Continuous dynamical systems are governed by differential equations

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

  — Discrete dynamical systems are governed by difference equations

$$x_{k+1} = f(x_k, k), \quad x_k \in R^n$$

  — A continuous dynamical system can be approximated by a discrete dynamical system.

For example, let $t_{k+1} = t_k + \Delta t$, and $x_k \approx x(t_k)$. Then, $\frac{dx(t)}{dt} = f(x(t))$ can be approximated by

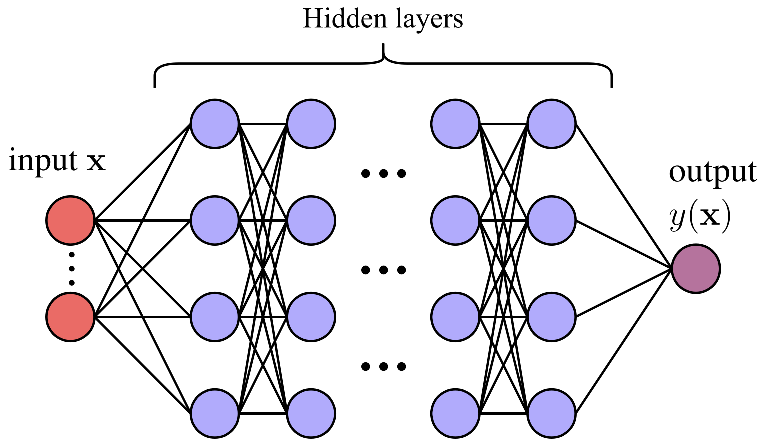
$$x_{k+1} = x_k + \Delta t f(x_k) \qquad \text{(Euler discretization)}$$

Examples of Discrete Dynamics — Deep Neural Networks

Many deep neural networks such as feedforward and residual networks can be mathematically represented as an iterative map

$$h_{k+1} = h_k + f(h_k, \theta_k)$$

where (h_k, h_{k+1}) are the input and output of a hidden layer, f is the activation function, and θ_k are neural network parameters (weights, bias).



Feedforward deep NNs

Discrete dynamics, $h_{k+1} = h_k + f(h_k, \theta_k)$, can be viewed as a Euler discretization of some continuous differential equation: $\dfrac{dh(t)}{dt} = \bar{f}(h(t), \theta(t))$

> Such relation has been explored to develop more efficient machine learning algorithms, for example, Neural ODEs (Chen et al., NeurIPS 2018). Here is a nice introduction to this topic: https://www.youtube.com/watch?v=AD3K8j12EIE

Examples of Discrete Dynamics — Gradient Descent ML Algorithm

Gradient descent algorithm is widely used in machine learning. It's an efficient algorithm to find a minimum of a cost function (loss function). The algorithm finds an optimal solution by an iterative process:

$$x_{k+1} = x_k - \alpha \nabla J(x_k)$$

where J is the cost function (loss function) to be minimized, x_k is the current approximate solution, and α is a (small) constant called learning rate.

This discrete dynamics can be rewritten as $\dfrac{x_{k+1} - x_k}{\alpha} = -\nabla J(x_k)$.

As the learning rate α reducing to 0, the gradient descent algorithm converges to

$$\frac{dx(t)}{dt} = -\nabla J(x(t)) \longleftarrow$$ (differential equation called gradient dynamical system)

A very active current research area in ML is to analyze and design better gradient descent type of ML algorithms from dynamical system (differential equations) point of view.

Some Terminologies

- System of 1st order ODEs and higher order ODEs.
  - General form of system of 1st order ODEs.

$$\left\{\begin{array}{lcl} \frac{dx_1(t)}{dt} & = & f_1(x_1(t), x_2(t), \ldots, x_n(t), t) \\ \frac{dx_2(t)}{dt} & = & f_2(x_1(t), x_2(t), \ldots, x_n(t), t) \\ & \vdots & \\ \frac{dx_n(t)}{dt} & = & f_n(x_1(t), x_2(t), \ldots, x_n(t), t) \end{array}\right. \Longleftrightarrow$$

A compact form of system of 1st ODEs:

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

$$x(t) \in R^n \qquad f(x,t) : R^n \times R \to R^n$$

  - Higher-order ODEs: $y^{(n)} = g(y(t), y'(t), \ldots, y^{(n-1)}(t), t)$

  - A higher-order ODE can always be transformed into a system of 1st order ODEs.

Define:

$$\left\{\begin{array}{lcl} x_1(t) & = & y(t) \\ x_2(t) & = & y'(t) \\ & \vdots & \\ x_n(t) & = & y^{(n-1)}(t) \end{array}\right. \Longrightarrow \left\{\begin{array}{lcl} \frac{dx_1(t)}{dt} & = & x_2(t) \\ \frac{dx_2(t)}{dt} & = & x_3(t) \\ & \vdots & \\ \frac{dx_{n-1}(t)}{dt} & = & x_n(t) \\ \frac{dx_n(t)}{dt} & = & g(x_1(t), x_2(t), \ldots, x_n(t), t) \end{array}\right.$$

Some Terminologies

- **Time invariant / Time varying ODEs**

— A time varying equation depends explicitly on the independent variable.

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

— A time invariant equation does not explicitly depend on the independent variable.

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

— Any time varying ODE can be transferred into a time invariant one.

Define a new variable $z(t) = [x_1(t), \ldots, x_n(t), t] \in R^{n+1}$. Then,

$$\frac{dz(t)}{dt} = \begin{bmatrix} \frac{dx(t)}{dt} \\ \frac{dt}{dt} \end{bmatrix} = \begin{bmatrix} f(x(t), t) \\ 1 \end{bmatrix} = g(z(t))$$

Some Terminologies

- **Linear / nonlinear** ODES.
  - — If f depends linearly on x(t) and its derivatives, the equation is a linear ODE. Otherwise, the equation is nonlinear.
  - — A linear ODE has the form:

$$\dot{x} = Ax + b,$$

    where A is a n-by-n square matrix, and b is a n-by-1 column vector.
  - — Review linear dynamical systems (AM 20)
  - — This course focuses on nonlinear systems.

Topics of AM 114

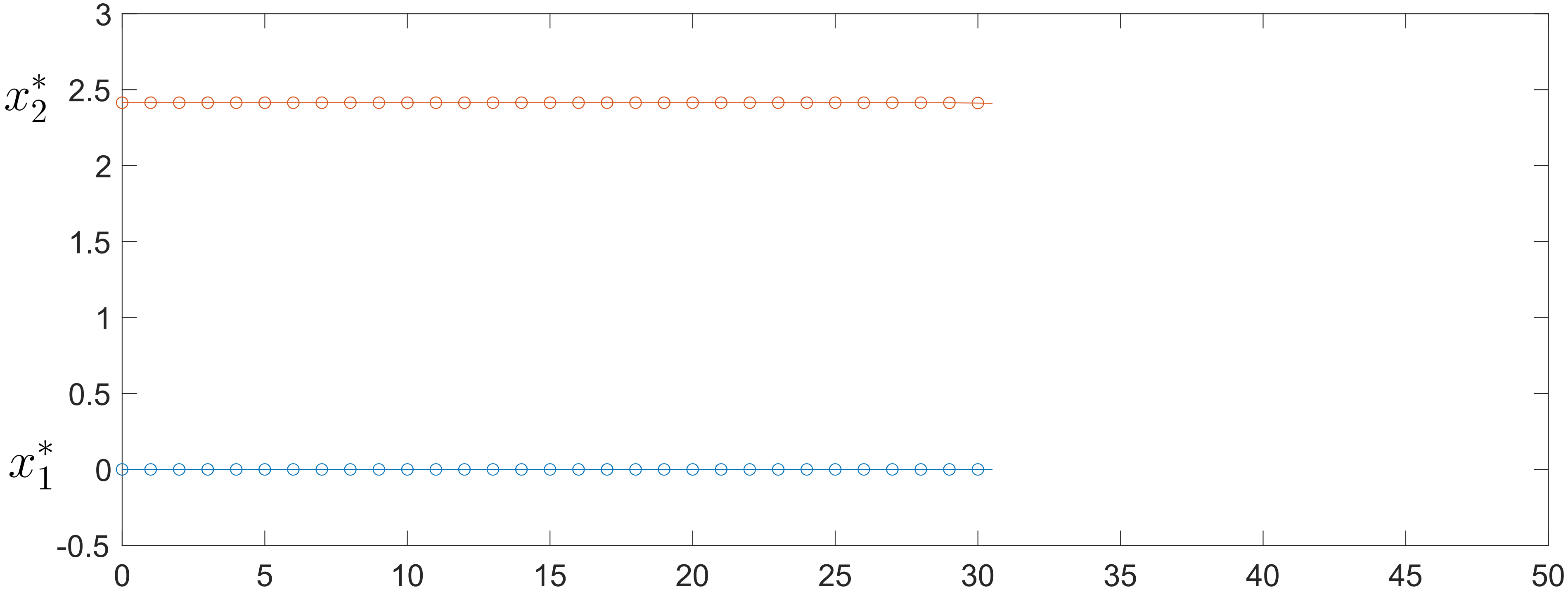
- **One-dimensional flow**
  - — Fixed points, stability …
  - — Bifurcation
- **Two-dimensional systems**
  - — Conservative systems
  - — reversible systems,
  - — Limits cycles
- **Higher-dimensional systems**
  - — Chaos
  - — …

Topics of AM 114

- **One-dimensional flow**
  — Fixed points, stability …

Discrete Logistic equation: $x_{n+1} = rx_n(1 - x_n)$

Two fixed points: $x_1^* = 0$ and $x_2^* = 1 - \dfrac{1}{r}$



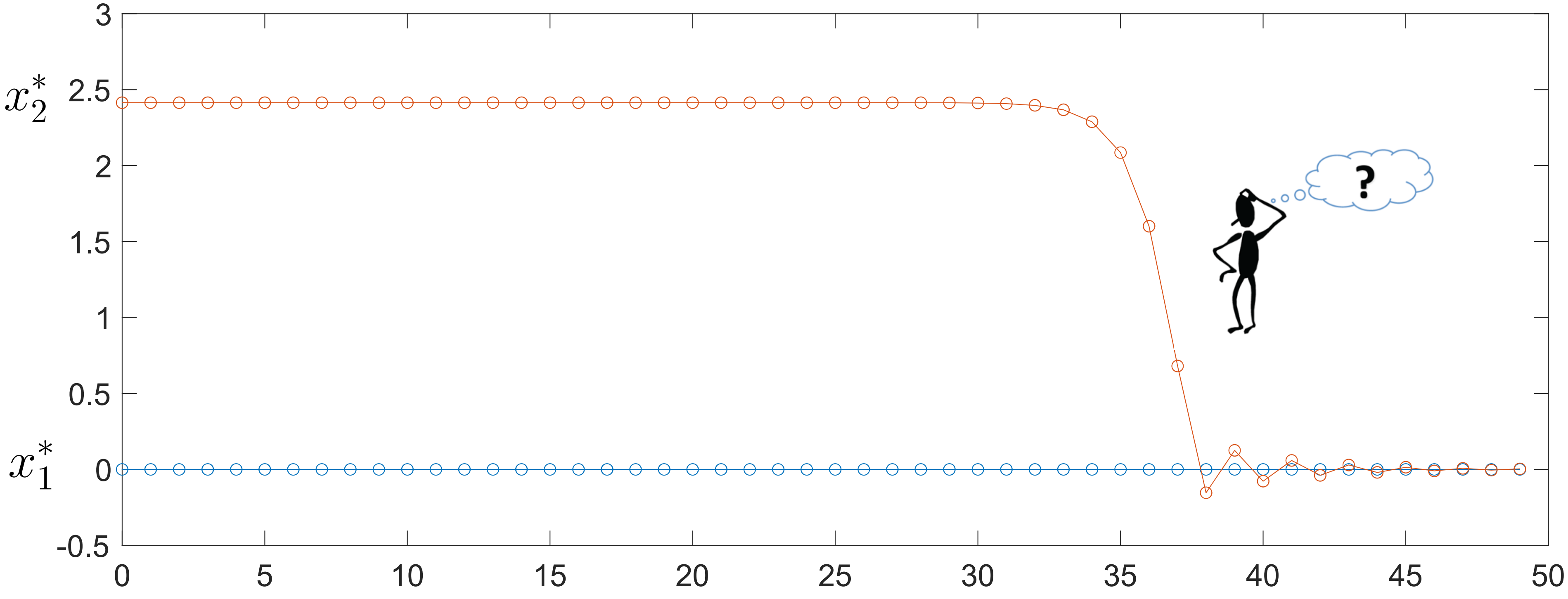
$r = -1/\sqrt{2}$

Topics of AM 114

- **One-dimensional flow**
  — Fixed points, stability …

Discrete Logistic equation: $x_{n+1} = r x_n (1 - x_n)$

Two fixed points: $x_1^* = 0$ and $x_2^* = 1 - \dfrac{1}{r}$



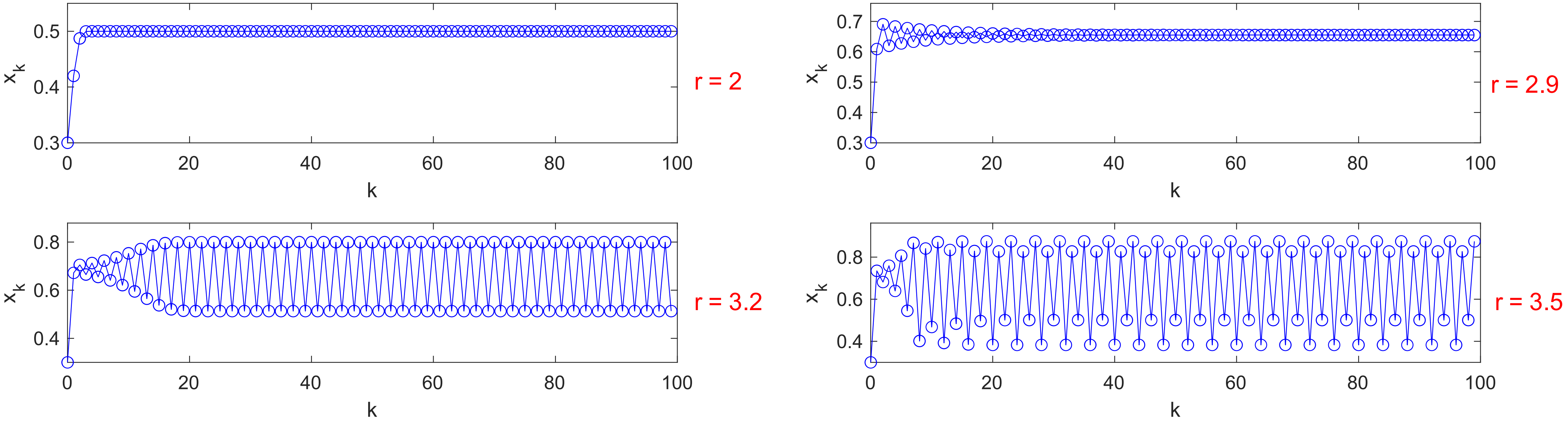
$r = -1/\sqrt{2}$

Topics of AM 114

- **One-dimensional flow**
  - — Fixed points, stability …
  - — **Bifurcation**

Discrete Logistic equation: $x_{n+1} = rx_n(1 - x_n)$

Bifurcation: change to the parameter of a dynamical system causes a sudden 'qualitative' or topological change in its behavior.

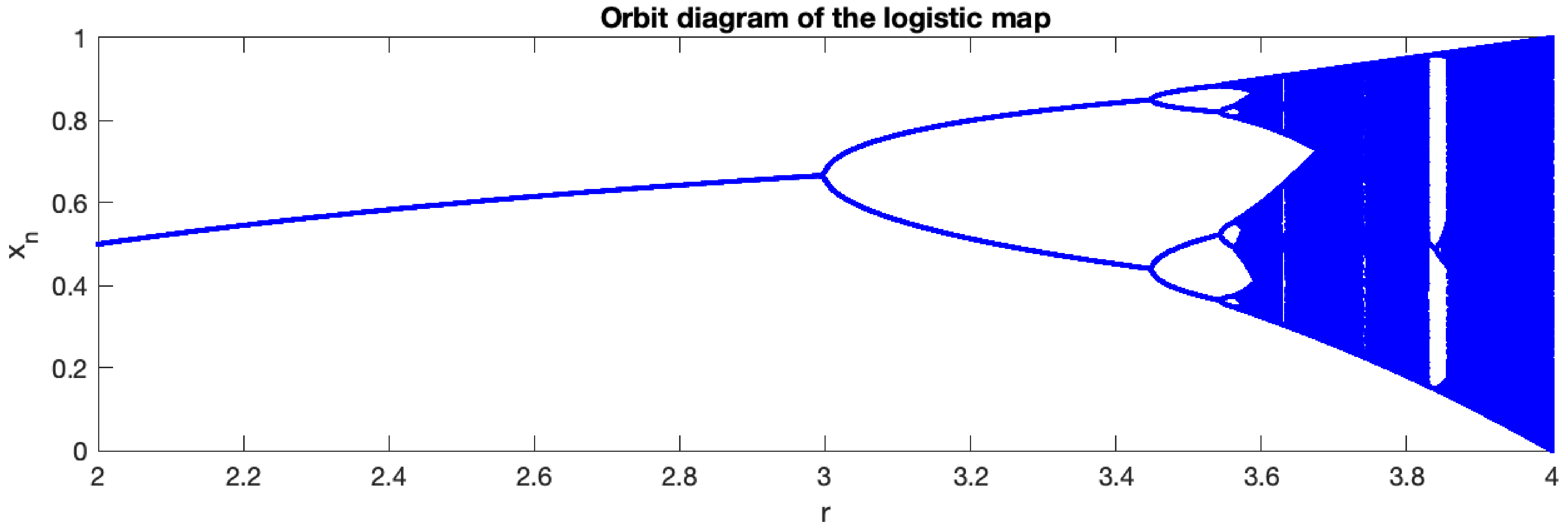


Topics of AM 114

- **One-dimensional flow**
  — Fixed points, stability …
  — **Bifurcation**

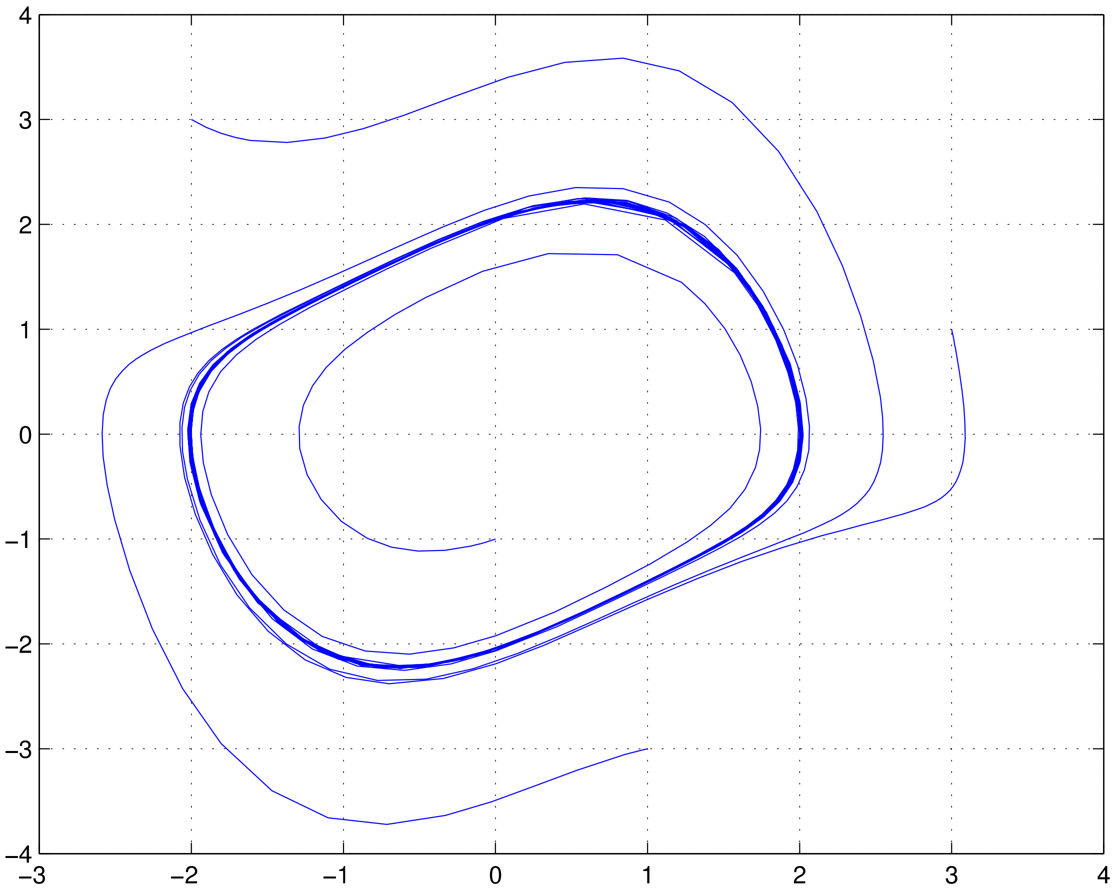
Discrete Logistic equation: $x_{n+1} = r x_n (1 - x_n)$

Bifurcation: change to the parameter of a dynamical system causes a sudden 'qualitative' or topological change in its behavior.



Topics of AM 114

- **One-dimensional flow**
  - — Fixed points, stability …
  - — Bifurcation
- **Two-dimensional systems**
  - — Conservative systems
  - — Reversible systems,
  - — Limit cycles



Limit cycle — Van Der Pol equation

Topics of AM 114

- **One-dimensional flow**
  - — Fixed points, stability …
  - — Bifurcation
- **Two-dimensional systems**
  - — Conservative systems
  - — reversible systems,
  - — Limits cycles
- **Higher-dimensional systems**
  - — Chaos
  - — …