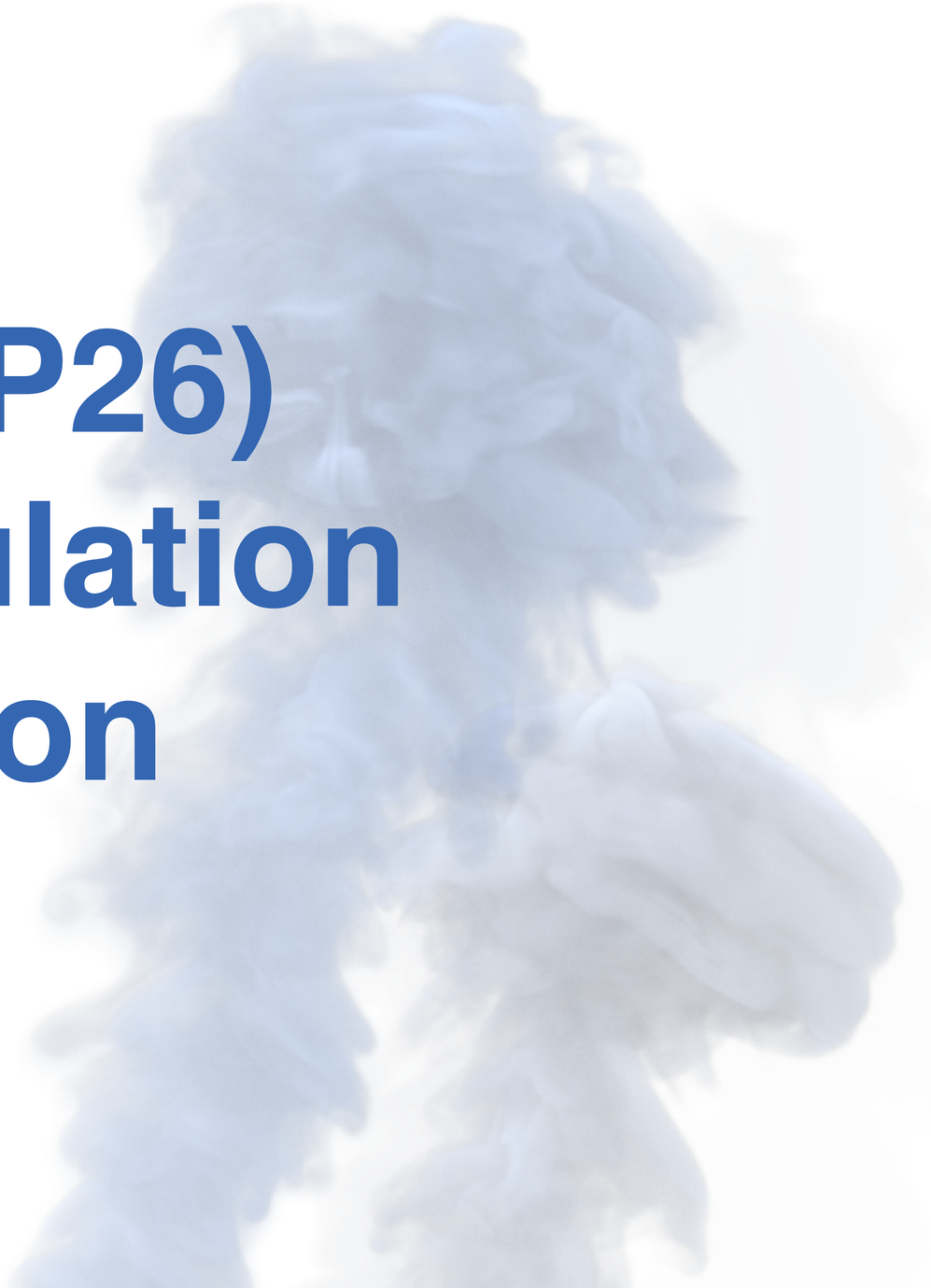
CSE 291 (SP26)
Physics Simulation
Introduction

Albert Chern

UC San Diego
JACOBS SCHOOL OF ENGINEERING



Course information

Physics Simulation

- **Instructor:** Albert Chern
- **TA:** Hesper Yin
- **Course website:** https://cseweb.ucsd.edu/~alchern/teaching/cse291_sp26/
- We use **Piazza** and **Gradescope**

Course information

Physics Simulation

- **Goal:** Mathematical principles behind simulation tasks
  Hands-on experience with physics-based animations
- **Applications:** Computer animation, scientific computing
  classical mechanics, theory abstraction
- **Grade:** HWs (HW0, mid and final project)
- **Collaboration:** Final submissions should be individual work, but we encourage you to study the math and solve the problems together!
- **ChatGPT, LLM:** Treat these AI as your colleague/classmate. You may ask them to proof read, but not write HW for you.

Course information

Physics Simulation

- **Prerequisites:**
  - Linear algebra, multivariable calculus, elementary physics
  - Using one programming platform with visualization that is capable of using/importing sparse matrix library
    - e.g. graphics software: Houdini, Blender, Unity
    - e.g. C++, Python, MATLAB, Javascript+WebGL
- **What tools can you use:**
  - Build your own solver from lower level (you can use built-in geometry processing functions) Don’t use a full-blown built-in simulation solver.

Course information

No lecture this Thursday

Simulation, Physics, Math

- Simulation, Physics, Math
- Getting started: F = ma
- Classification of ODEs
- Solve ODEs numerically

Physics simulation

- In computational physics, engineering, computer graphics,…
- Generate computer-generated data that mimic that we would observe in the physical world.
- Why?
  - Make predictions, conduct virtual experiments
  - Believable visual effects
- How?

Physics simulation

- How?
  - **Mathematical modeling**

    Turn physical phenomena into mathematical equations.

    (What are the variables? What are the laws of physics)
  - **Analysis**

    Get a general idea of how the solution should behave.

    (Is the problem well-posed?)
  - **Computation**

    Solve (approximate) solutions analytically or numerically.

Physics simulation

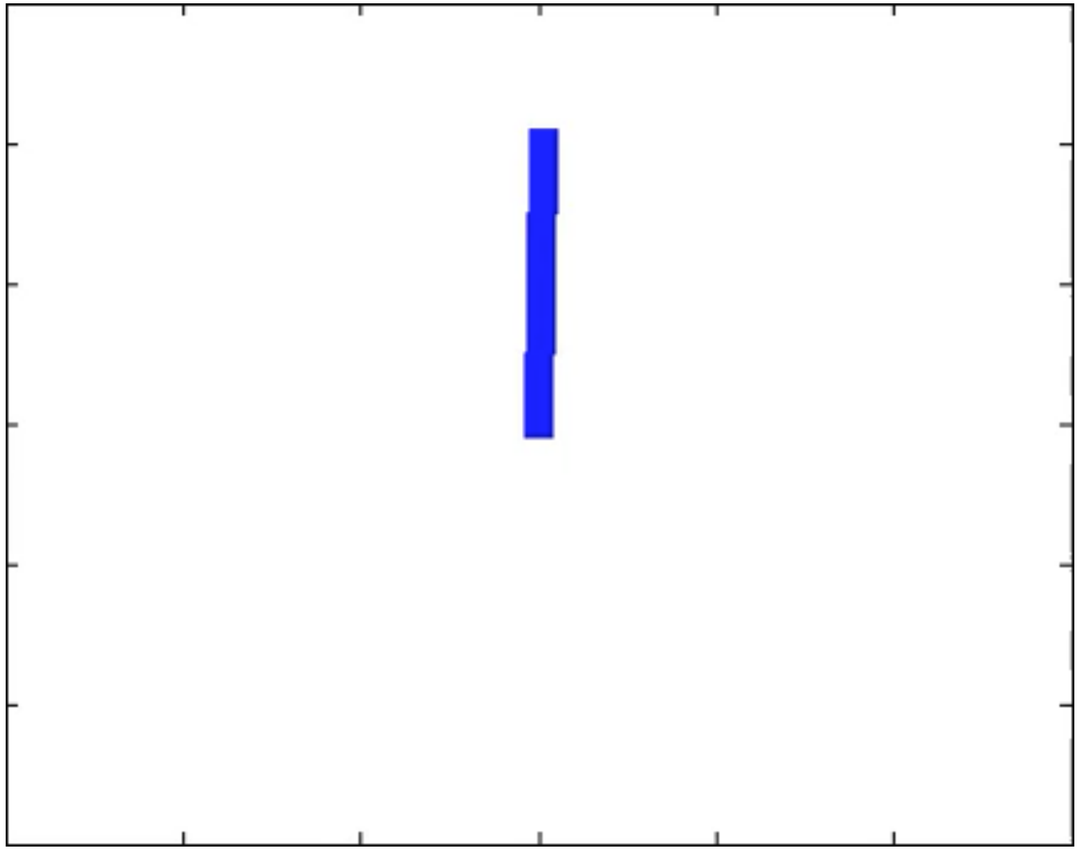
We will focus on **general principles**

- Dimensional analysis
- Least action principle
- Incremental potential formulation
- Constitutive modeling in continuum mechanics

Systems we will cover

- Small mechanical systems
- Rigid body
- Constrained system (linkage, robotics, collision and contact)
- Fluids
- Elasticity

Physics simulation



Physics simulation



Youtube “Soft Body Tetris [01]” by ImbaPixel https://youtu.be/rm44SV8xUDo



Nabizadeh, Wang, Ramamoorthi, C.
Covector Fluids
2022

Getting started: F=ma

- Simulation, Physics, Math
- Getting started: F = ma
- Classification of ODEs
- Solve ODEs numerically

Physics based motion

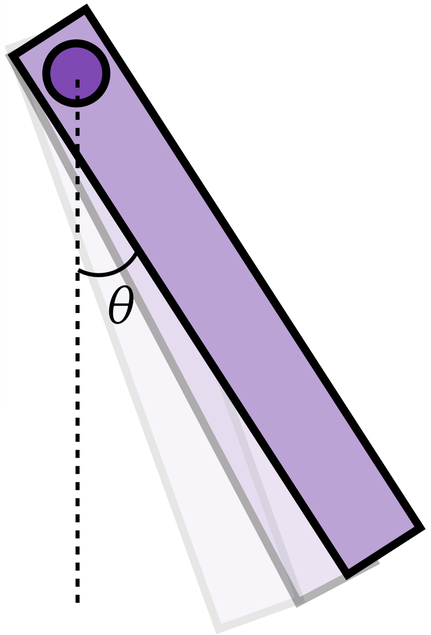
Exercise 0.1 — 5pt. Using your favorite program to produce an animation of a simple physical system. It could be a pendulum motion like demonstrated in the lecture, or other system.

(a) Upload a video of your result.

(b) Upload a written document that briefly explains the system. (Include the equation of motion, an explanation of what each variable in the equation means, and what the time stepping algorithm looks like.)

(c) Upload the source file(s) (for example .zip).

■



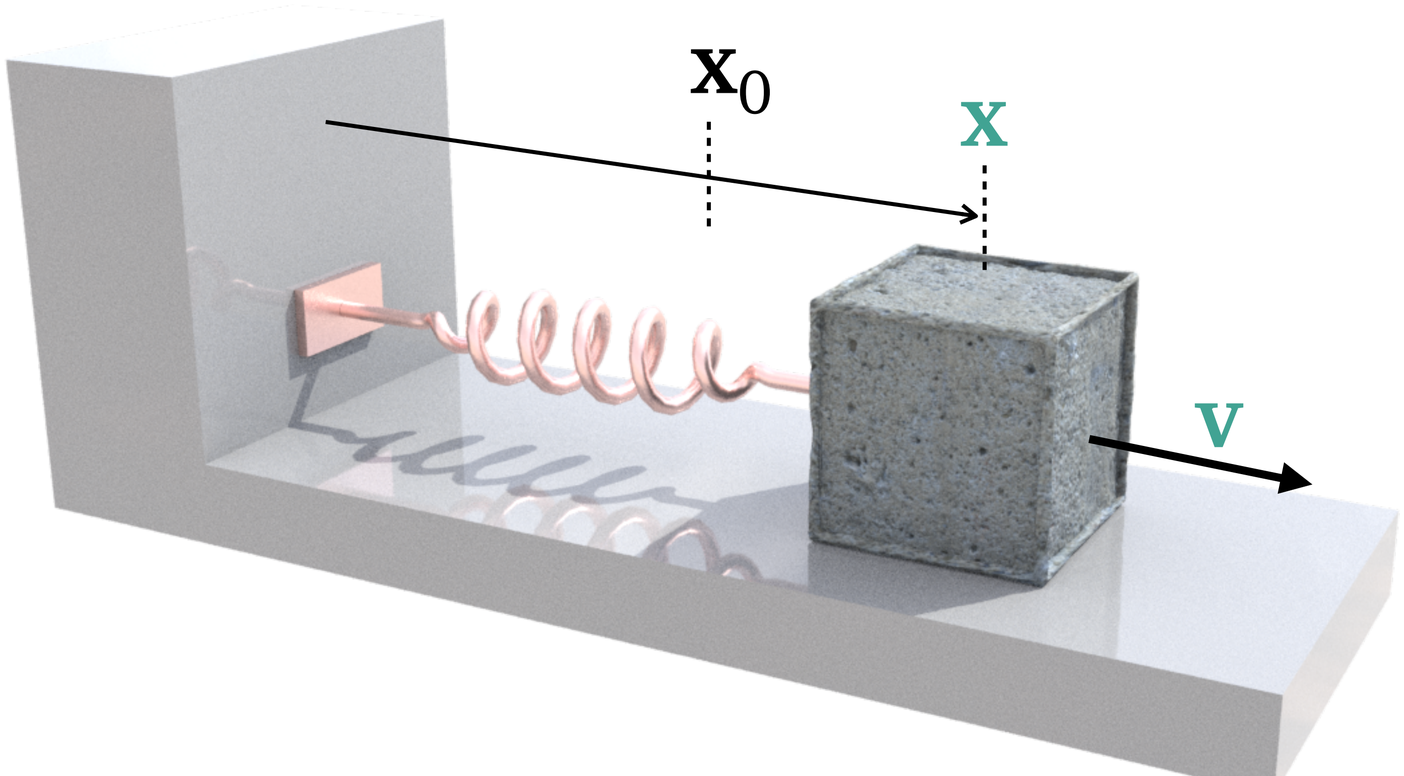
Physics based motion

Rough idea:

- Position of each object is governed by Newton's law of motion
- Rate of change of position is called velocity
- Rate of change of velocity is called acceleration
- Model "force" as a function of position and velocity
- Newton's law of motion: Mass x acceleration = force

Example

- Animate an object attached to a spring
- Identify the moving position: **x**
- Associated velocity **v**

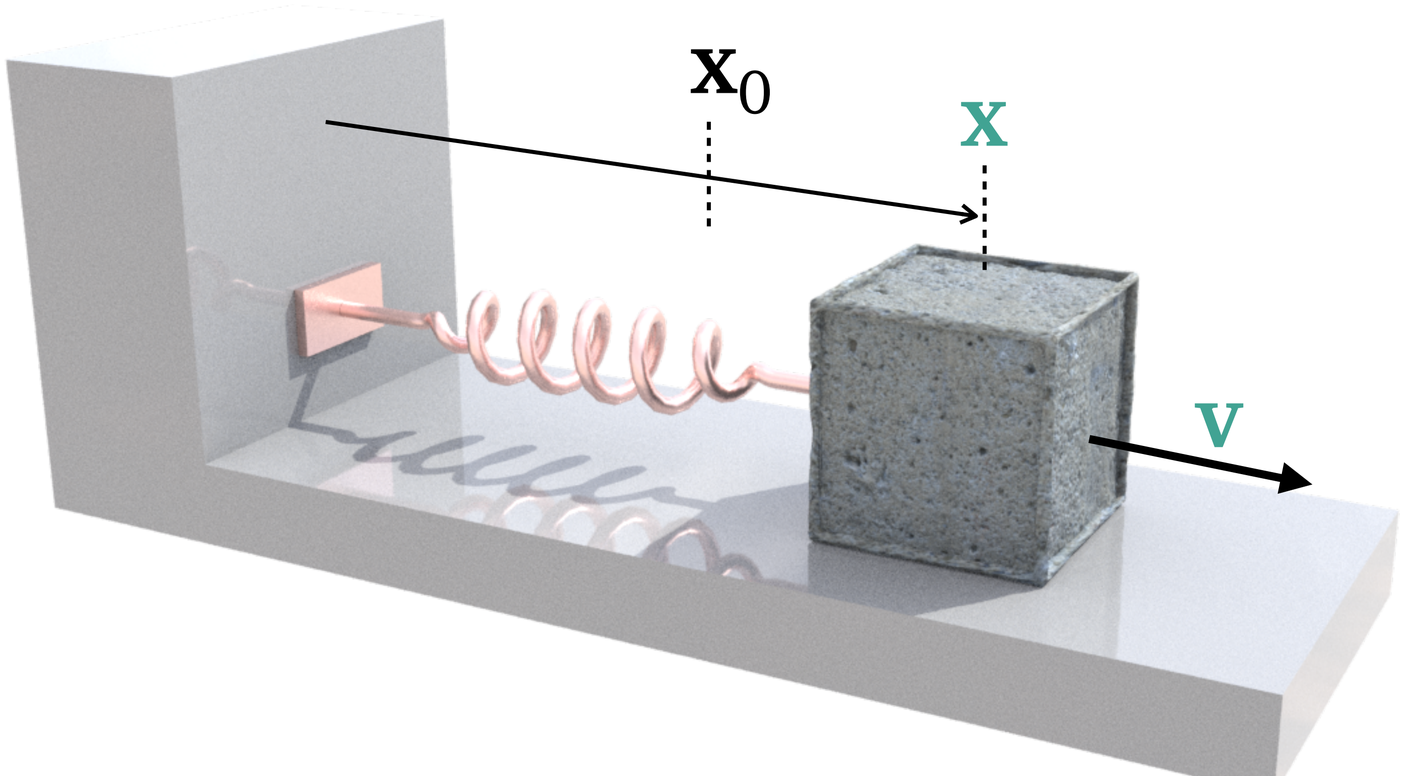


Example

- Force

$$\mathbf{f}(\mathbf{x}, \mathbf{v}) = -k(\mathbf{x} - \mathbf{x}_0) - \mu\mathbf{v}$$

rest position

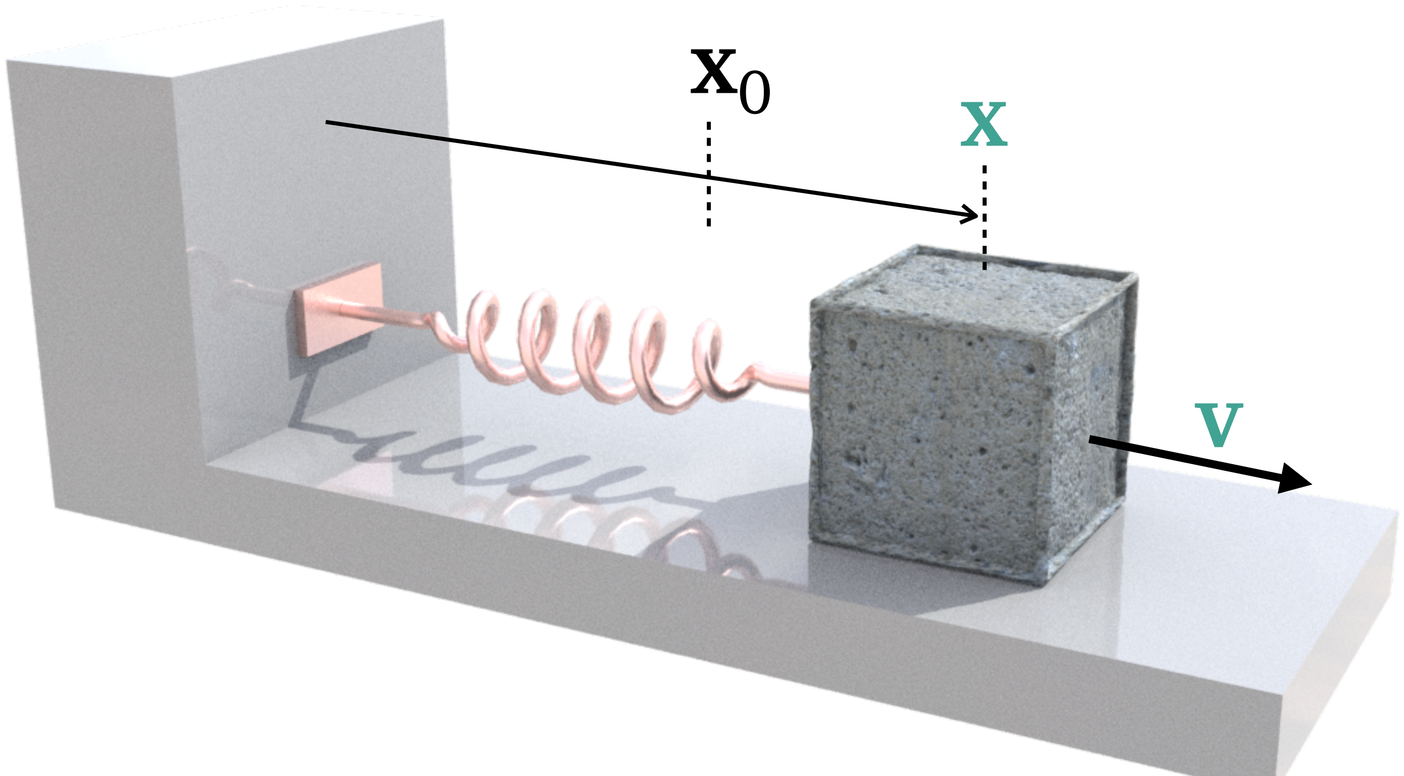


Example

- Force

$$\mathbf{f}(\mathbf{x}, \mathbf{v}) = -k(\mathbf{x} - \mathbf{x}_0) - \mu\mathbf{v}$$

stiffness of spring

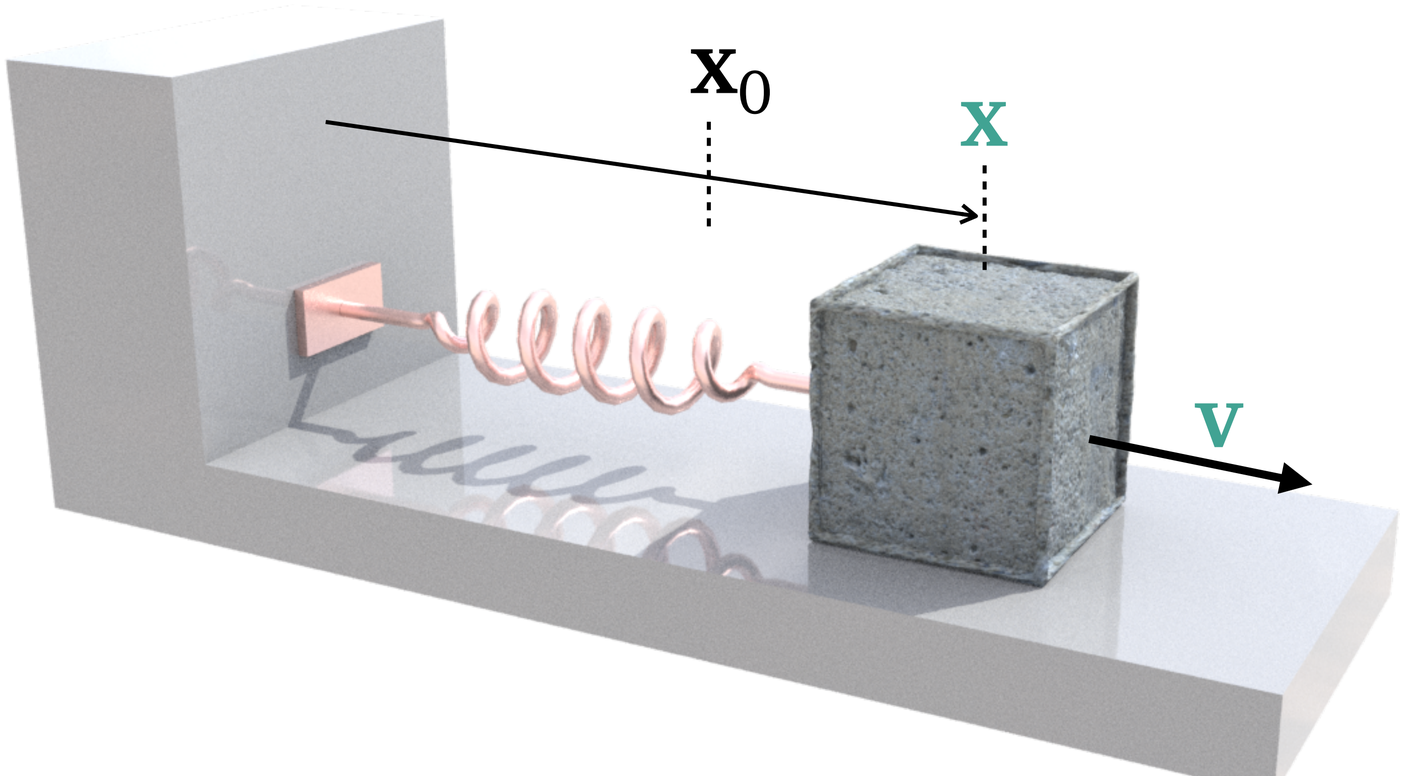


Example

- Force

$$\mathbf{f}(\mathbf{x}, \mathbf{v}) = -k(\mathbf{x} - \mathbf{x}_0) - \mu\mathbf{v}$$

friction



Example

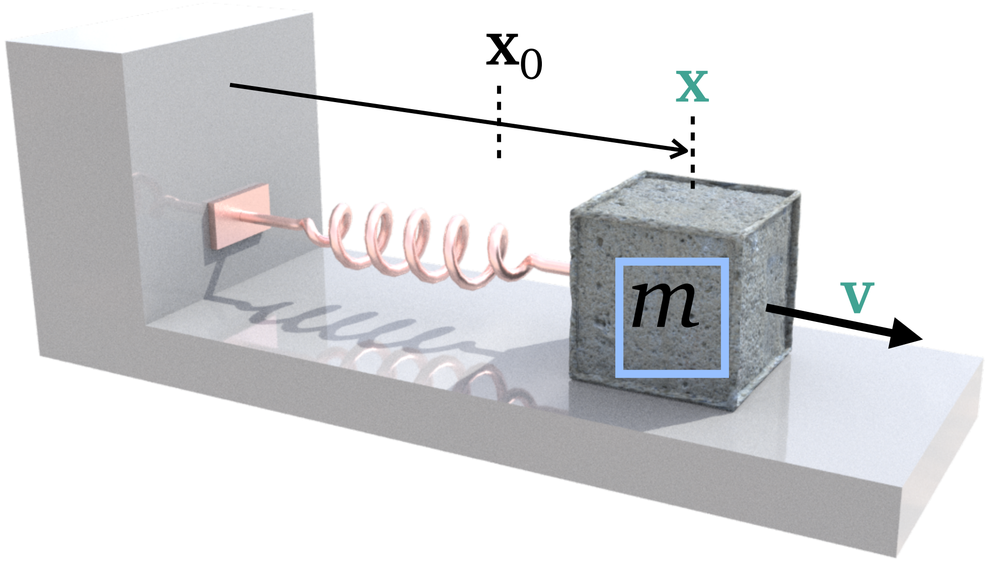
- Force

$$\mathbf{f}(\mathbf{x}, \mathbf{v}) = -k(\mathbf{x} - \mathbf{x}_0) - \mu\mathbf{v}$$

- Equations of motion

$$\frac{d\mathbf{x}}{dt} = \mathbf{v}$$

relationship between position and velocity



$$\frac{d\mathbf{v}}{dt} = \frac{1}{m}\mathbf{f}(\mathbf{x}, \mathbf{v})$$

relationship between acceleration and force

mass

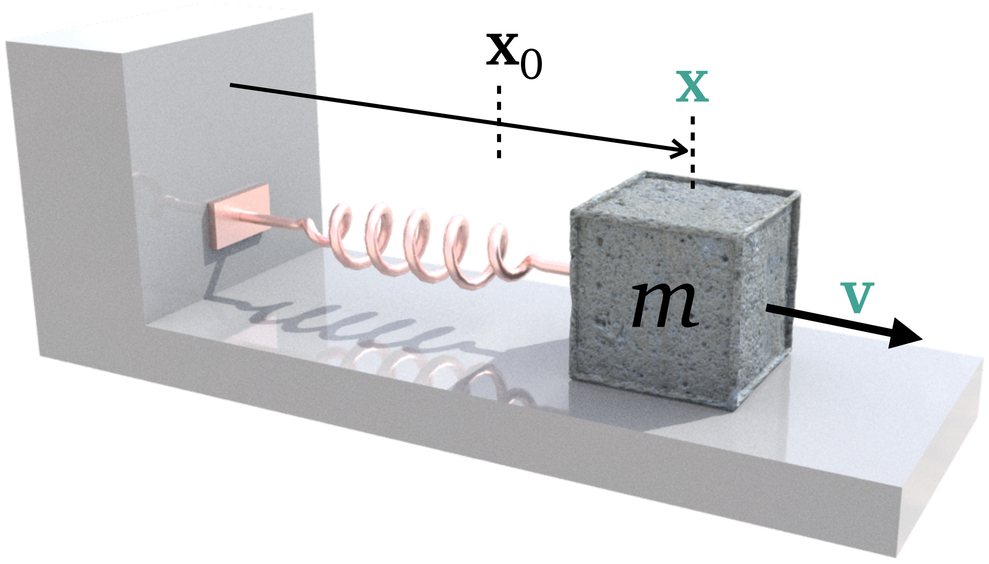
Example

- Force

$$\mathbf{f}(\mathbf{x}, \mathbf{v}) = -k(\mathbf{x} - \mathbf{x}_0) - \mu\mathbf{v}$$

- Equations of motion

$$\frac{d\mathbf{x}}{dt} = \mathbf{v}$$



$$\frac{d\mathbf{v}}{dt} = \frac{1}{m}\mathbf{f}(\mathbf{x}, \mathbf{v})$$

$$= -\frac{k}{m}(\mathbf{x} - \mathbf{x}_0) - \frac{\mu}{m}\mathbf{v}$$

Example

- We use the overhead dots to indicate time derivatives

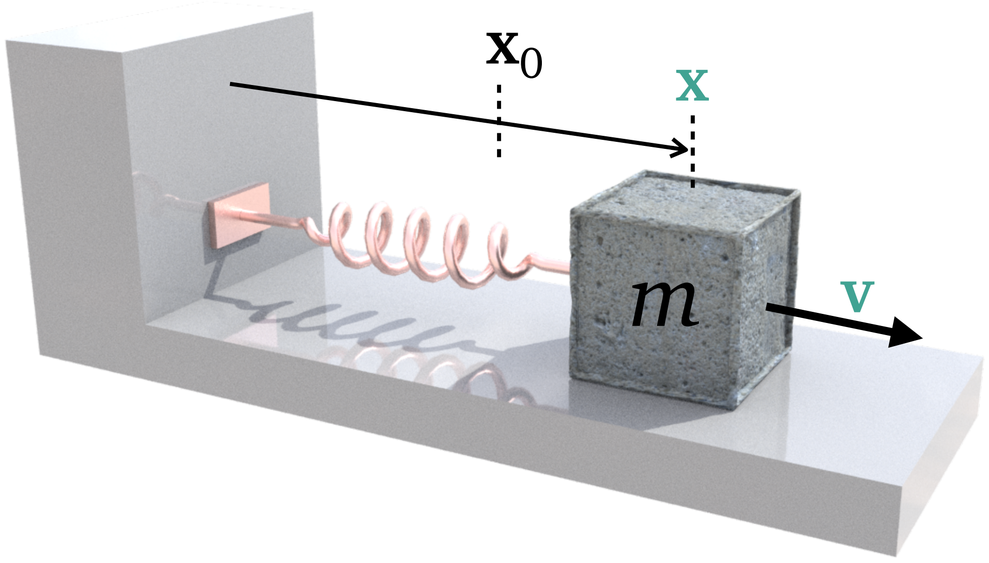
$$\frac{d\mathbf{x}(t)}{dt} = \dot{\mathbf{x}}(t) \qquad \frac{d^2\mathbf{x}(t)}{dt^2} = \ddot{\mathbf{x}}(t)$$

Example

- Equations of motion $\dot{\mathbf{x}} = \mathbf{v}$

$$\dot{\mathbf{v}} = -\frac{k}{m}(\mathbf{x} - \mathbf{x}_0) - \frac{\mu}{m}\mathbf{v}$$

- Substitute **v**: $$\ddot{\mathbf{x}} = -\frac{k}{m}(\mathbf{x} - \mathbf{x}_0) - \frac{\mu}{m}\dot{\mathbf{x}}$$



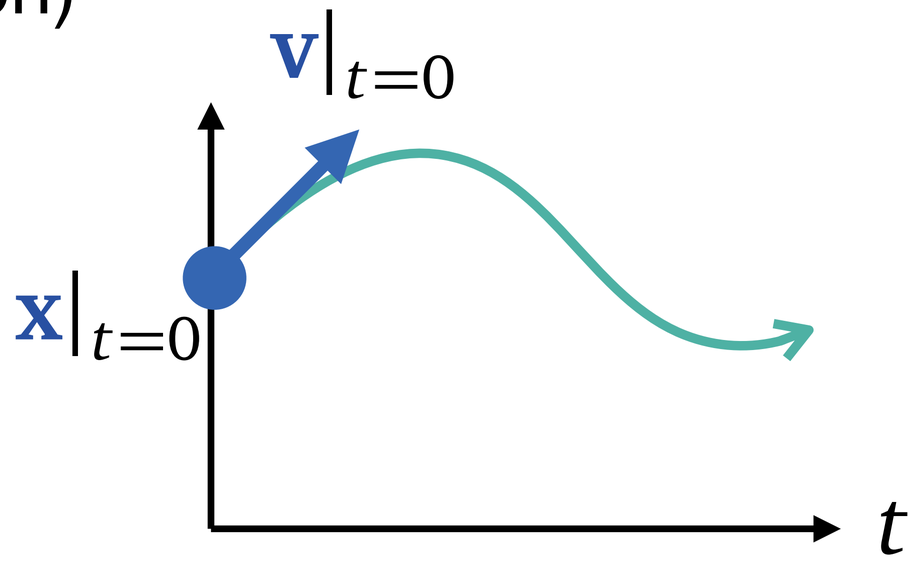
- Equation involving $\mathbf{x}$ and its derivatives
- This is called an *ordinary differential equation (ODE)*

Types of problems

- Equations of motion $\ddot{\mathbf{x}} = -\frac{k}{m}(\mathbf{x} - \mathbf{x}_0) - \frac{\mu}{m}\dot{\mathbf{x}}$
  - This is called an *ordinary differential equation (ODE)*
- Initial value problem (forward simulation)
  - Given initial conditions
    i.e. the values of $\mathbf{x}|_{t=0}, \mathbf{v}|_{t=0}$

    Extend them into function of time

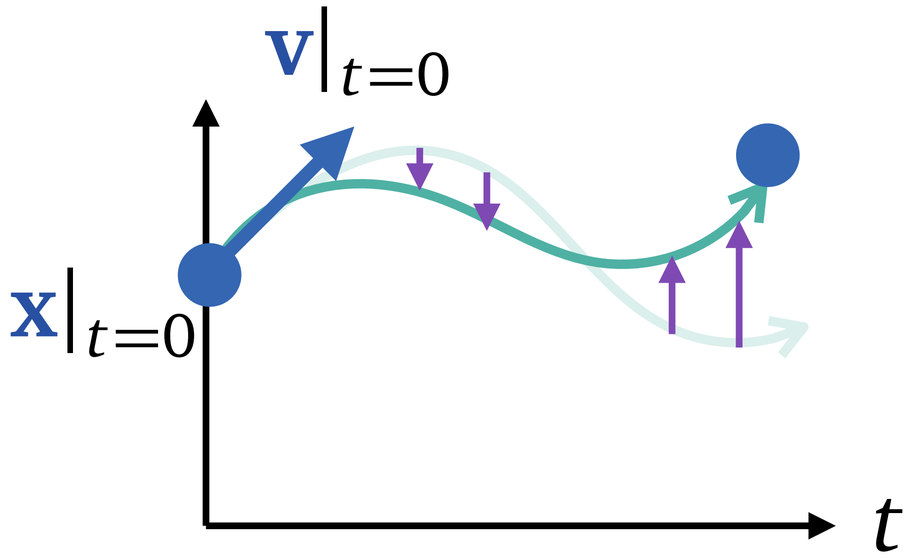
    $\mathbf{x}(t), \mathbf{v}(t) \quad t \geq 0$



Types of problems

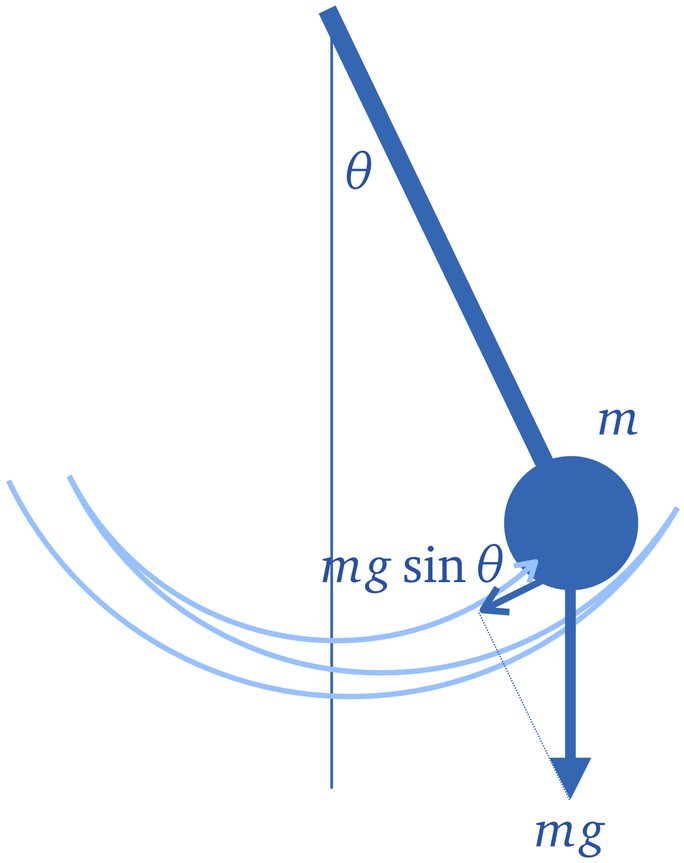
- Equations of motion $\ddot{\mathbf{x}} = -\frac{k}{m}(\mathbf{x} - \mathbf{x}_0) - \frac{\mu}{m}\dot{\mathbf{x}}$
  - This is called an *ordinary differential equation (ODE)*
- Control problem / inverse problem
  - Given desired location to arrive at some future time,

    find minimal correction force to achieve the goal.
  - Robotics, control systems, physics-based keyframe animations



Another example

- Pendulum



$$m\ddot{\theta} = -mg \sin \theta$$

Classification of ODEs

- Simulation, Physics, Math
- Getting started: F = ma
- Classification of ODEs
- Solve ODEs numerically

Classification of ODEs

Definition
An ordinary differential equation (ODE) is a condition for a function

$$\mathbf{y}\colon [0,T] \to \mathbb{R}^m \qquad \mathbf{y}(t) = \begin{bmatrix} y_1(t) \\ | \\ y_m(t) \end{bmatrix}$$

of the form

$$F_t\left(\mathbf{y}(t), \dot{\mathbf{y}}(t), \ddot{\mathbf{y}}(t), \ldots, \mathbf{y}^{(k)}(t)\right) = 0 \quad \text{for all} \quad t \in [0,T]$$

- If $m = 1$: ***scalar ODE***; otherwise ($m > 1$): ***ODE system***
- *k* is called the **order** of the ODE.
- If F_t is linear function of its argument, then it's a **linear ODE**.
- If $F_t = F$ is independent of t, then it's **autonomous**.
- If $F_t(\alpha\mathbf{u}_1, \ldots, \alpha\mathbf{u}_k) = \alpha^r F_t(\mathbf{u}_1, \ldots, \mathbf{u}_k)$ it's **homogeneous ODE**.

A few simple cases

Theorem *If* $\mathbf{y}_1, \mathbf{y}_2$ *are two solutions of a* ***homogeneous linear ODE****,*
then $\mathbf{y}(t) = c_1\mathbf{y}_1(t) + c_2\mathbf{y}_2(t)$ *will also be a solution.*

That is, the set of solutions form a ***vector space****.*

The dimension of this vector space is $k \times m$*.*

Definition *A set of basis for the solution space of homogeneous linear ODE is called the* ***fundamental solutions****.*

Theorem *If the ODE is linear (but inhomogeneous), then the solution space is an* ***affine space****. That is, all solutions can be written as a solution plus linear combinations of the fundamental solutions to the homogeneous part of the ODE.*

A few simple cases

Remark *(Linear/nonlinear) first order scalar ODE can be solved by hand if you can separate* y, t *variables.*

- Example (exponential type) $\dot{y} = \lambda y \implies y = ce^{\lambda t}$
- Riccati equation $\dot{y} = ty^2 \implies y = \frac{2}{C - t^2}$

Remark *If a first order ODE is not separable, we can still solve it when it is linear. (First solve homogeneous part, and then solve the full system using variation of constant, a.k.a. integrating factor)*

- Example (volume rendering equation) $\dot{y} = -\sigma(t)y + \rho(t)$

Order (m) and size of system (k)

Theorem *An m-th order ODE system of size k is equivalent to an 1st order ODE of size* $k \times m$.

- Try convert a 2nd order ODE $m\ddot{\theta} = -mg \sin\theta$ to a first order system.
- The conversion preserve notions of linearity, homogeneity, autonomy.
- Linear ODEs can all be transformed to $\dot{\mathbf{y}}(t) = \mathbf{A}(t)\mathbf{y}(t) + \mathbf{b}(t)$
  square matrix, vector

First order linear system

- Linear ODEs can all be transformed to $\dot{\mathbf{y}}(t) = \underset{\text{square matrix}}{\mathbf{A}(t)}\mathbf{y}(t) + \underset{\text{vector}}{\mathbf{b}(t)}$
- When **A** depends on t, we don't have good way of solving it (unless the size of **A** is 1x1). (There are Frobenius method for recursively generating solution in power series form.)
- When **A** is constant, then solve its *eigenvalue problem* for the homogeneous solutions.
- Explicitly, homo. sol's are given by matrix exponential $\mathbf{y}(t) = e^{t\mathbf{A}}\mathbf{y}_0$
- To find a solution with nontrivial **b**(t), use *variation of constant*.

General analytical methods

- Every ODE can be converted into $\dot{\mathbf{y}}(t) = \mathbf{f}(t, \mathbf{y}(t))$.
- $\mathbf{f}(t, \cdot): \mathbb{R}^{mk} \to \mathbb{R}^{mk}$ can be thought of as a *vector field* on *mk*-space and the solution is a path on this space tangential to the vector field (best when it is autonomous).
- Points $\mathbf{y}$ for which $\mathbf{f}(\mathbf{y}) = \mathbf{0}$ (for autonomous systems) are called steady states (static state).
- Near steady states, one can linearize the ODE

$$\mathbf{f}(\mathbf{y}_{\text{static}} + \epsilon \mathbf{y}_{\text{perturb}}(t)) \approx \underbrace{\mathbf{f}(\mathbf{y}_{\text{static}})}_{=0} + \epsilon \underbrace{\mathbf{A}}_{D\mathbf{f}|_{\mathbf{y}_{\text{static}}}} \mathbf{y}_{\text{perturb}}(t) + O(\epsilon^2)$$

  and get $\frac{d}{dt}\mathbf{y}_{\text{perturb}}(t) = \mathbf{A}\mathbf{y}_{\text{perturb}}(t)$

- One can speak about stability of steady states in terms of eig-val of **A**.

General analytical methods

- For non-autonomous systems, one can still track stable "quasi-static" states that depends on t

$$\mathbf{f}_t(\mathbf{y}_{\text{static}}) = 0$$

- Assumption is **f** depends on t much slower than the system relaxes back to steady state.

- Population of fish with the presence of fisherman

$$\dot{y} = \underbrace{y(1-y)}_{\text{growth}} - a(t)$$

environment capacity

- Draw the *bifurcation diagram*

Solving ODE Numerically

- Simulation, Physics, Math
- Getting started: F = ma
- Classification of ODEs
- Solve ODEs numerically

ODE

- Derive the differential equation (ODE) from physical laws
- Solve the differential equation (ODE)
  - Hard to solve it by hand most of the time
  - Numerical method is needed

ODE

- Given any differential equation, for example,

$$\dddot{\mathbf{x}} + \ddot{\mathbf{x}}\dot{\mathbf{x}} + \sin(\ddot{\mathbf{x}}) = 1$$

- Convert it into a 1st order system of ODEs (involving at most first derivative)
  - Give each derivative a separate name (except for the highest order derivative) $\mathbf{v} = \dot{\mathbf{x}}$ $\mathbf{a} = \ddot{\mathbf{x}}$
  - Then

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{v} \\ \dot{\mathbf{v}} = \mathbf{a} \\ \dot{\mathbf{a}} = 1 - \mathbf{a}\mathbf{v} - \sin(\mathbf{a}) \end{cases}$$

ODE

- Given any differential equation, for example,

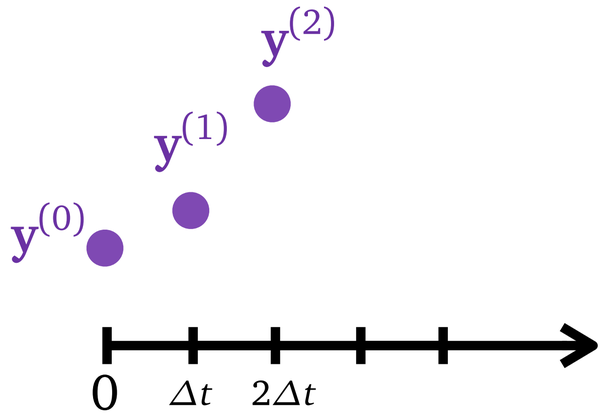
$$\dddot{\mathbf{x}} + \ddot{\mathbf{x}}\dot{\mathbf{x}} + \sin(\ddot{\mathbf{x}}) = 1$$

  - Then $\begin{cases} \dot{\mathbf{x}} = \mathbf{v} \\ \dot{\mathbf{v}} = \mathbf{a} \\ \dot{\mathbf{a}} = 1 - \mathbf{a}\mathbf{v} - \sin(\mathbf{a}) \end{cases}$
  - Let $\mathbf{y} = \begin{bmatrix} \mathbf{x} \\ \mathbf{v} \\ \mathbf{a} \end{bmatrix}$ ODE becomes $\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y})$

Numerical ODE

- Generic ODE $\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y})$
- Discretize time into time-frames $\mathbf{y}^{(n)} = \mathbf{y}(n\Delta t)$
- **(Forward) Euler method** $\frac{\mathbf{y}^{(n+1)} - \mathbf{y}^{(n)}}{\Delta t} \approx \mathbf{f}(\mathbf{y}^{(n)})$

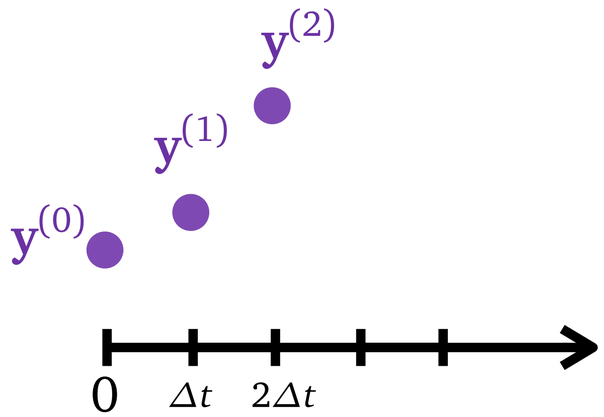
$$\mathbf{y}^{(n+1)} \approx \mathbf{y}^{(n)} + \Delta t \cdot \mathbf{f}(\mathbf{y}^{(n)})$$



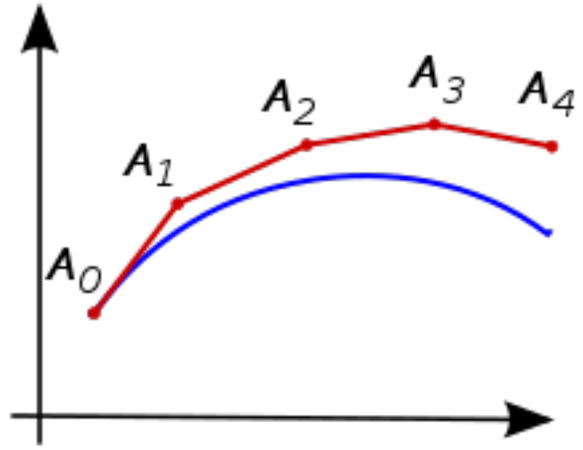
Numerical ODE

- **(Forward) Euler method** $\frac{\mathbf{y}^{(n+1)} - \mathbf{y}^{(n)}}{\Delta t} \approx \mathbf{f}(\mathbf{y}^{(n)})$

$$\mathbf{y}^{(n+1)} \approx \mathbf{y}^{(n)} + \Delta t \cdot \mathbf{f}(\mathbf{y}^{(n)})$$



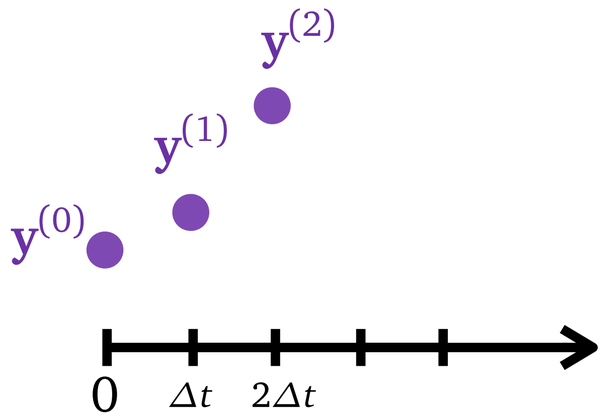
- Advantages
  - There is an **explicit** formula to plug in old state to get new state
  - Fast and simple
- Limitations
  - Not very accurate unless Δt is tiny
  - Can be energy increasing (unphysical)



Numerical ODE

- **(Forward) Euler method** $\frac{\mathbf{y}^{(n+1)} - \mathbf{y}^{(n)}}{\Delta t} \approx \mathbf{f}(\mathbf{y}^{(n)})$

$$\mathbf{y}^{(n+1)} \approx \mathbf{y}^{(n)} + \Delta t \cdot \mathbf{f}(\mathbf{y}^{(n)})$$



- **Backward Euler method**

$$\frac{\mathbf{y}^{(n+1)} - \mathbf{y}^{(n)}}{\Delta t} = \mathbf{f}(\mathbf{y}^{(n+1)})$$

- Limitations
  - Not very accurate unless Δt is tiny
  - Have to solve for new state (**implicit**) instead of explicit update

Numerical ODE

- **Backward Euler method**

$$\frac{\mathbf{y}^{(n+1)} - \mathbf{y}^{(n)}}{\Delta t} = \mathbf{f}(\mathbf{y}^{(n+1)})$$

- Limitations
  - Not very accurate unless Δt is tiny
  - Have to solve for new state (**implicit**) instead of explicit update
- Advantages
  - Energy decreasing (dissipating), which looks physical
  - Can take larger time steps Δt without instability
  - Can incorporate collision (just add constraint to the implicit solves)

Numerical ODE

- **Euler method** $\mathbf{y}^{(n+1)} \approx \mathbf{y}^{(n)} + \Delta t \cdot \mathbf{f}(\mathbf{y}^{(n)})$ $\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y})$

- **Runge–Kutta method (RK4)** (Accurate, stable, explicit)

$$\mathbf{k}_1 = \mathbf{f}(\mathbf{y}^{(n)})$$
$$\mathbf{k}_2 = \mathbf{f}(\mathbf{y}^{(n)} + \frac{\Delta t}{2}\mathbf{k}_1)$$
$$\mathbf{k}_3 = \mathbf{f}(\mathbf{y}^{(n)} + \frac{\Delta t}{2}\mathbf{k}_2)$$
$$\mathbf{k}_4 = \mathbf{f}(\mathbf{y}^{(n)} + \Delta t\mathbf{k}_3)$$
$$\mathbf{y}^{(n+1)} = \mathbf{y}^{(n)} + \frac{\Delta t}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4)$$

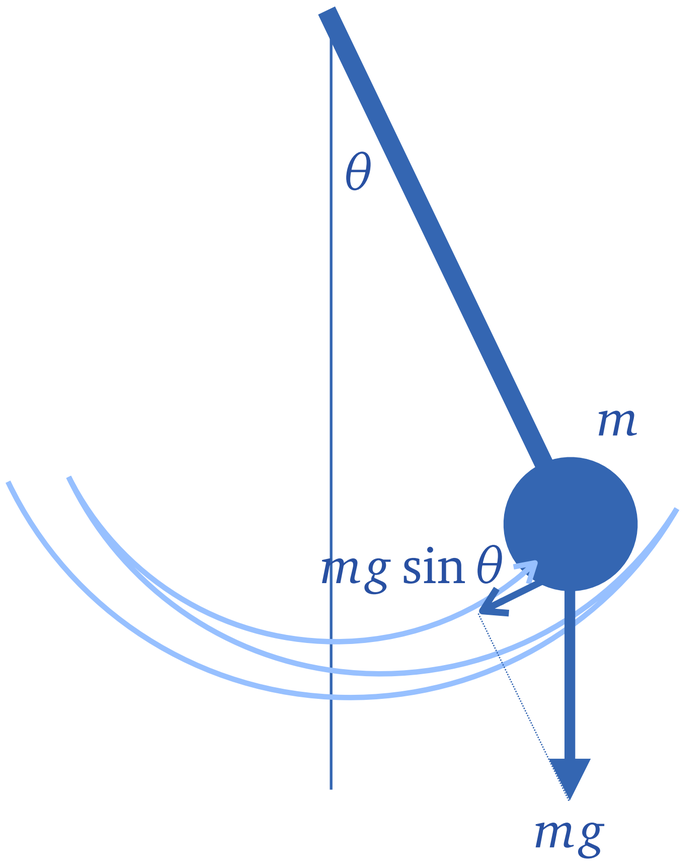
(collision handling is not as elegant as backward Euler)

Numerical ODE

- In most cases the RK4 method works very well
- Sometimes the underlying physical system has additional structures (energy conservation, momentum conservation)
- Special algorithm (non-RK4) aims at preserving energy or momentum
  - Variational integrator
  - Symplectic integrator
  - Lie group integrator

Pendulum equation

Example: pendulum equation.

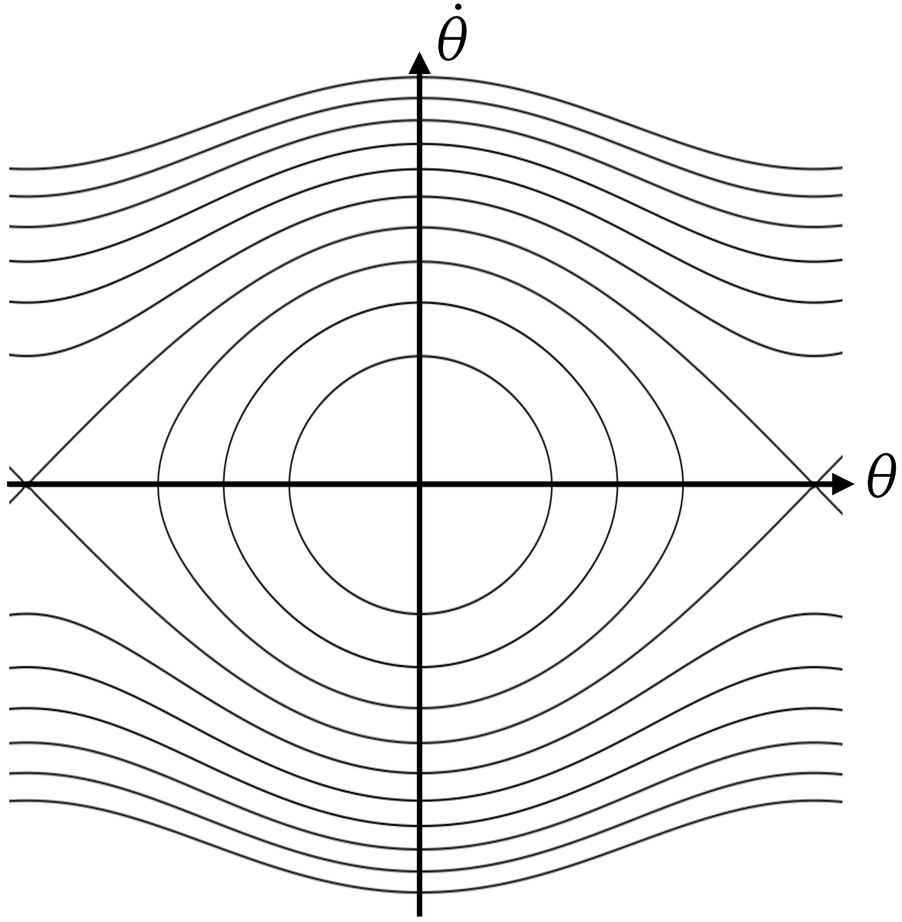


$$m\ddot{\theta} = -mg\sin\theta$$

✓ Energy conservation

$$\frac{1}{2}m\dot{\theta}^2 - mg\cos\theta = \text{const}$$

✓ Integrable system



Pendulum equation

Example: pendulum equation.

$$m\ddot{\theta} = -mg\sin\theta$$

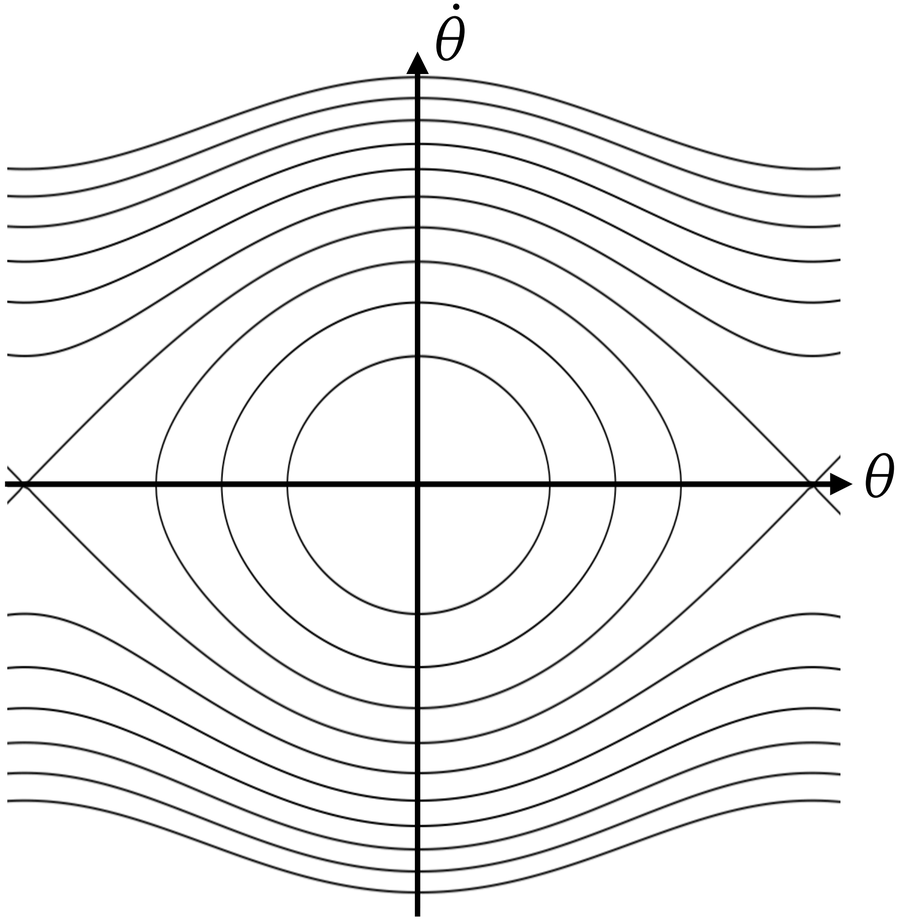
- High order differential equation solver (4th order Runge–Kutta method)

Given $(\theta_i, v_i = \dot{\theta}_i)$

- $\theta^*_{i+1/2} = \theta_i + \frac{\Delta t}{2} v_i \qquad v^*_{i+1/2} = v_i - \frac{\Delta t}{2}\sin\theta_i$
- $\theta^{**}_{i+1/2} = \theta_i + \frac{\Delta t}{2} v^*_{i+1/2} \qquad v^{**}_{i+1/2} = v_i - \frac{\Delta t}{2}\sin\theta^*_{i+1/2}$
- $\theta^{***}_{i+1} = \theta_i + \Delta t\, v^{**}_{i+1/2} \qquad v^{***}_{i+1} = v_i - \Delta t \sin\theta^{**}_{i+1/2}$

Output $\theta_{i+1} = \theta_i + \frac{\Delta t}{6}\left(v_i + 2v^*_{i+1/2} + 2v^{**}_{i+1/2} + v^{***}_{i+1}\right)$

$$v_{i+1} = v_i - \frac{\Delta t}{6}\left(\sin\theta_i + 2\sin\theta^*_{i+1/2} + 2\sin\theta^{**}_{i+1/2} + \sin\theta^{***}_{i+1}\right)$$



Pendulum equation

Example: pendulum equation.

$$m\ddot{\theta} = -mg\sin\theta$$

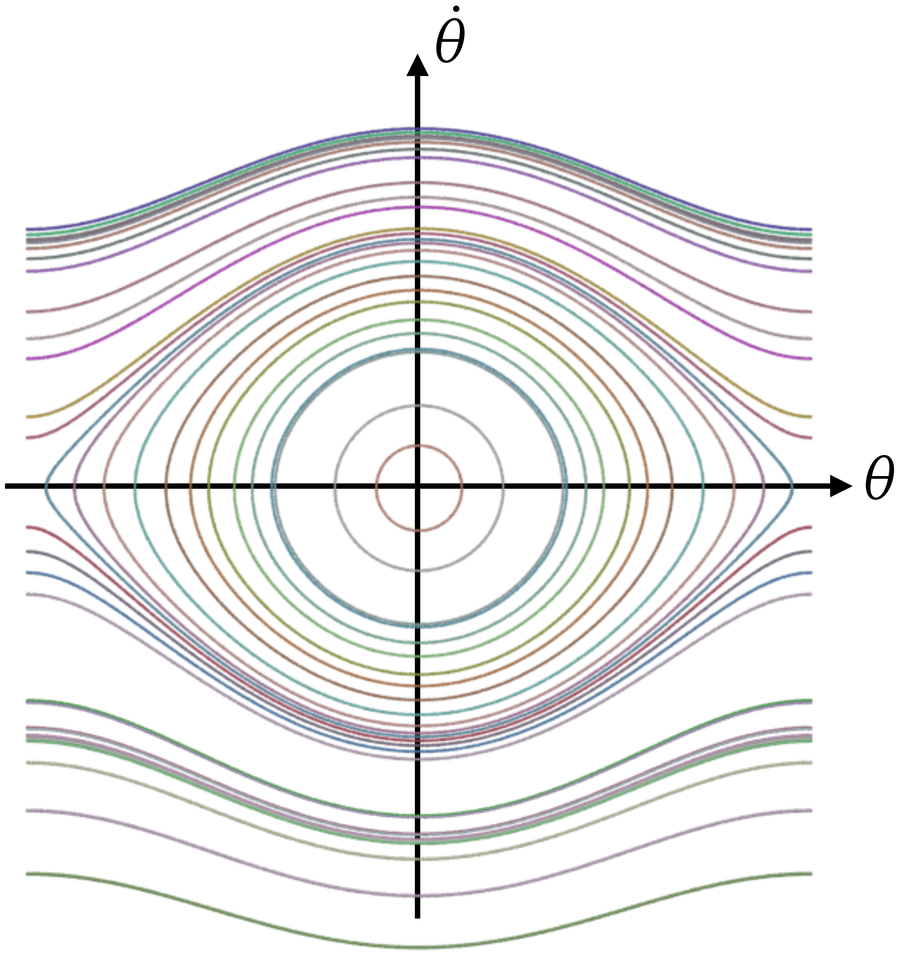
- High order differential equation solver (4th order Runge–Kutta method)

Given $(\theta_i, v_i = \dot{\theta}_i)$

- $\theta^*_{i+1/2} = \theta_i + \frac{\Delta t}{2} v_i \qquad v^*_{i+1/2} = v_i - \frac{\Delta t}{2}\sin\theta_i$
- $\theta^{**}_{i+1/2} = \theta_i + \frac{\Delta t}{2} v^*_{i+1/2} \qquad v^{**}_{i+1/2} = v_i - \frac{\Delta t}{2}\sin\theta^*_{i+1/2}$
- $\theta^{***}_{i+1} = \theta_i + \Delta t\, v^{**}_{i+1/2} \qquad v^{***}_{i+1} = v_i - \Delta t \sin\theta^{**}_{i+1/2}$

Output $\theta_{i+1} = \theta_i + \frac{\Delta t}{6}\left(v_i + 2v^*_{i+1/2} + 2v^{**}_{i+1/2} + v^{***}_{i+1}\right)$

$$v_{i+1} = v_i - \frac{\Delta t}{6}\left(\sin\theta_i + 2\sin\theta^*_{i+1/2} + 2\sin\theta^{**}_{i+1/2} + \sin\theta^{***}_{i+1}\right)$$



$\Delta t = 0.1$

Pendulum equation

Example: pendulum equation.

$$m\ddot{\theta} = -mg\sin\theta$$

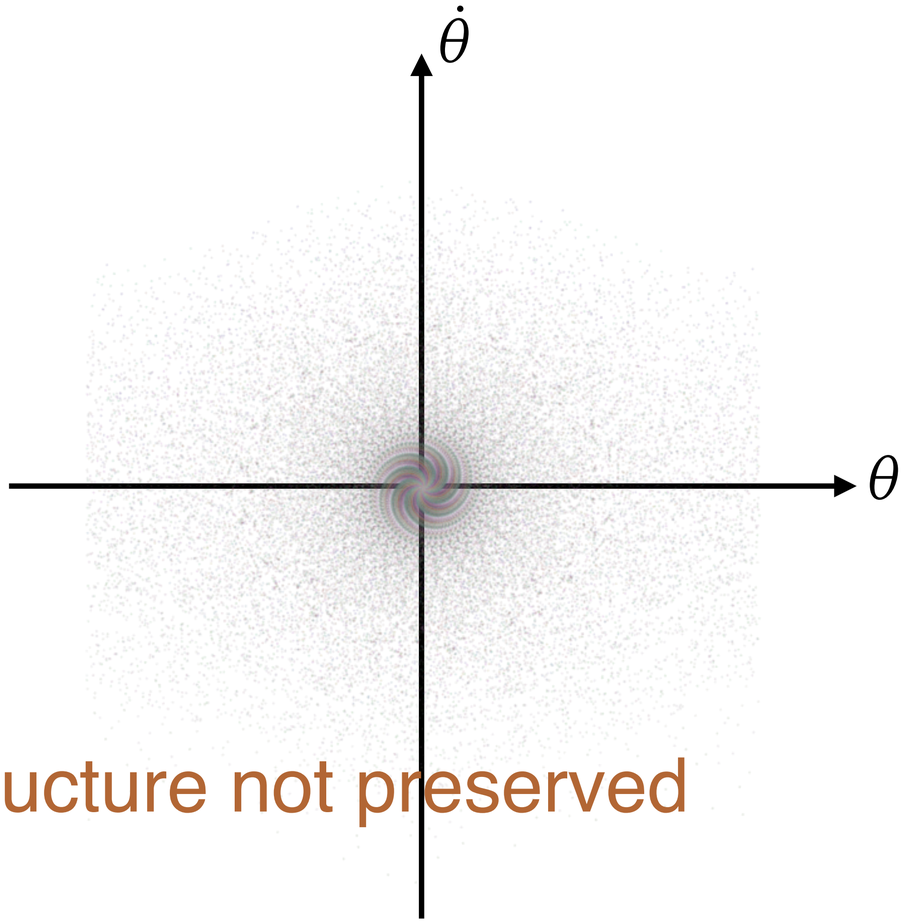
- High order differential equation solver (4th order Runge–Kutta method)

Given $(\theta_i, v_i = \dot{\theta}_i)$

- $\theta^*_{i+1/2} = \theta_i + \frac{\Delta t}{2} v_i \qquad v^*_{i+1/2} = v_i - \frac{\Delta t}{2}\sin\theta_i$
- $\theta^{**}_{i+1/2} = \theta_i + \frac{\Delta t}{2} v^*_{i+1/2} \qquad v^{**}_{i+1/2} = v_i - \frac{\Delta t}{2}\sin\theta^*_{i+1/2}$
- $\theta^{***}_{i+1} = \theta_i + \Delta t\, v^{**}_{i+1/2} \qquad v^{***}_{i+1} = v_i - \Delta t \sin\theta^{**}_{i+1/2}$

Output $\theta_{i+1} = \theta_i + \frac{\Delta t}{6}\left(v_i + 2v^*_{i+1/2} + 2v^{**}_{i+1/2} + v^{***}_{i+1}\right)$

$$v_{i+1} = v_i - \frac{\Delta t}{6}\left(\sin\theta_i + 2\sin\theta^*_{i+1/2} + 2\sin\theta^{**}_{i+1/2} + \sin\theta^{***}_{i+1}\right)$$



Structure not preserved

$\Delta t = 0.9$

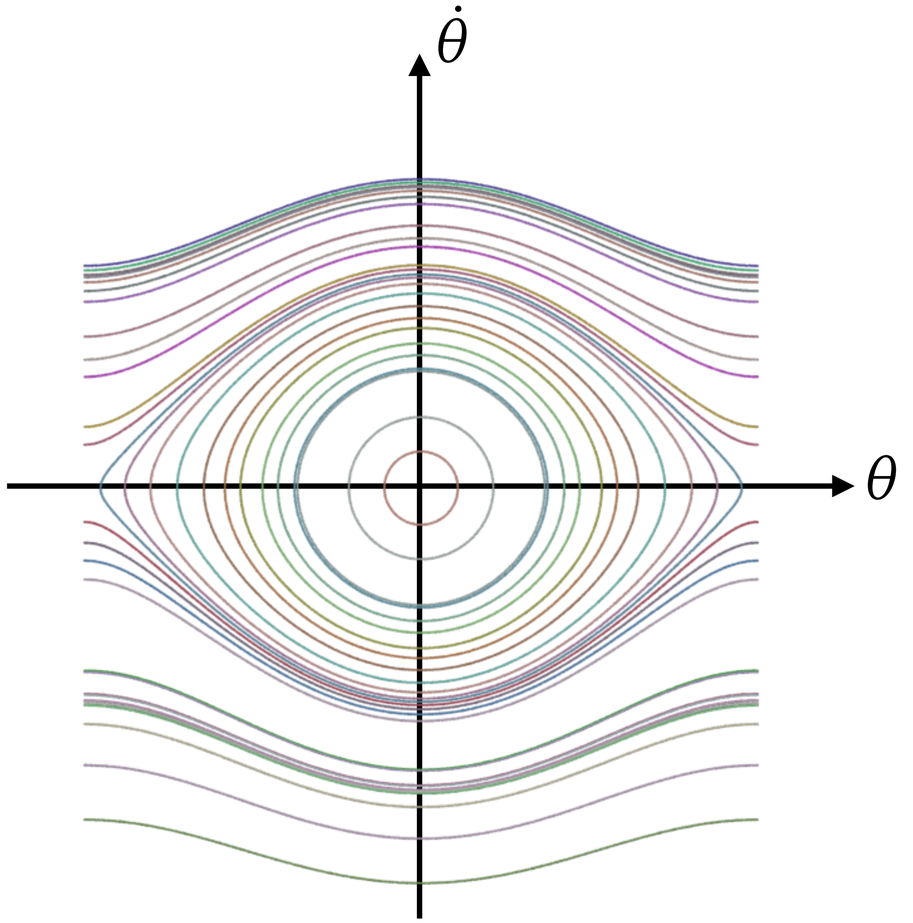
Pendulum equation

Example: pendulum equation.

$$m\ddot{\theta} = -mg\sin\theta$$

- A 2nd order discretization

$$\frac{\theta_{i-1} - 2\theta_i + \theta_{i+1}}{\Delta t^2} = -\sin\theta_i$$



$\Delta t = 0.1$

Pendulum equation

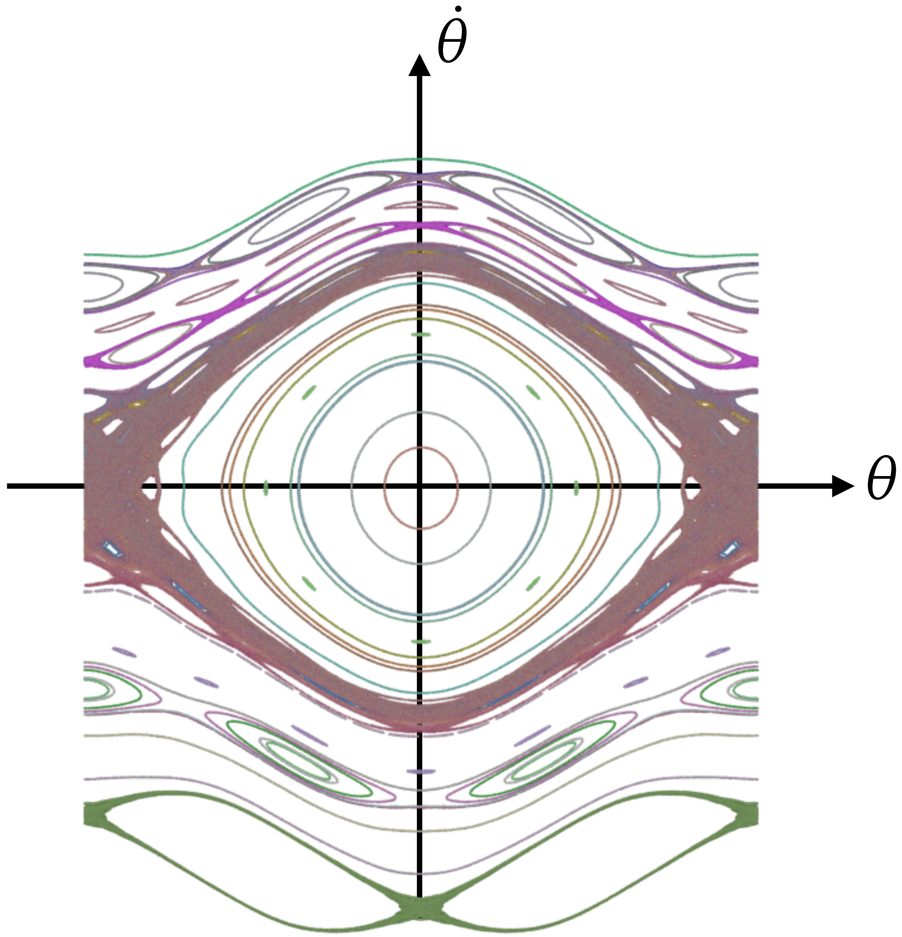
Example: pendulum equation.

$$m\ddot{\theta} = -mg\sin\theta$$

- A 2nd order discretization

$$\frac{\theta_{i-1} - 2\theta_i + \theta_{i+1}}{\Delta t^2} = -\sin\theta_i$$

✓ Energy conservation in “asteroid belts”



$\Delta t = 0.9$

Pendulum equation

Example: pendulum equation.

$$m\ddot{\theta} = -mg\sin\theta$$

Least action principle

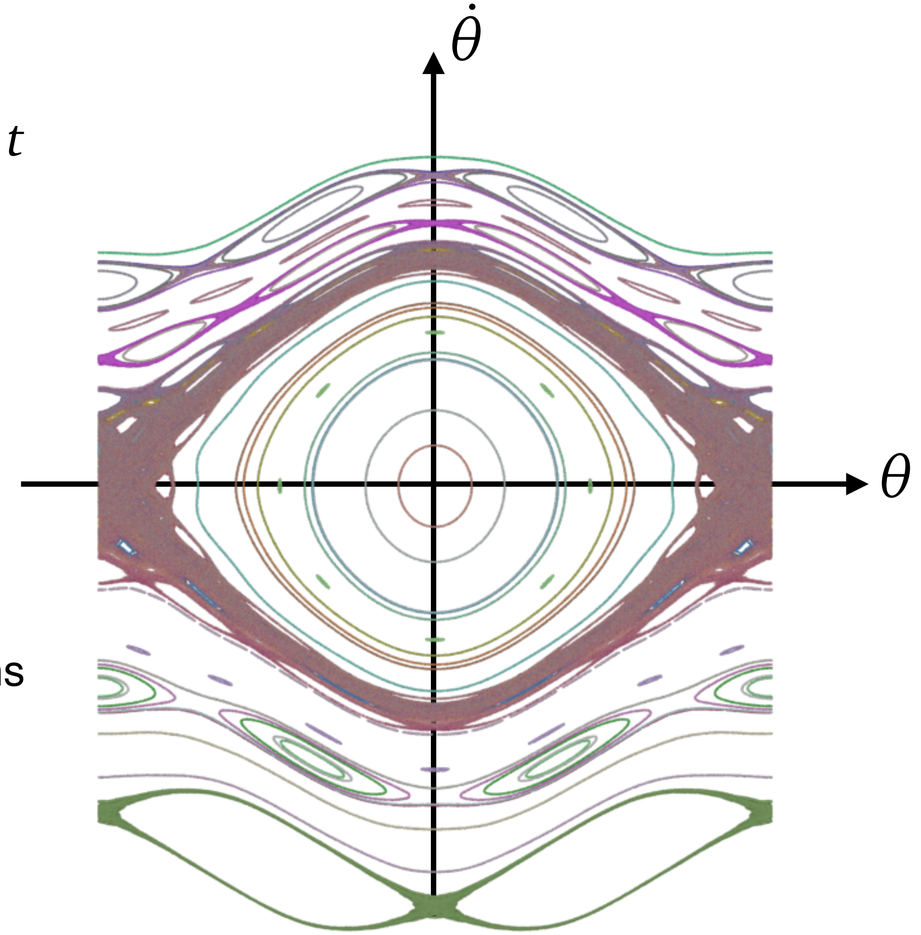
$$\int \left(\frac{m}{2}\dot{\theta}^2 + mg\cos\theta\right) dt$$

- A 2nd order discretization

$$\frac{\theta_{i-1} - 2\theta_i + \theta_{i+1}}{\Delta t^2} = -\sin\theta_i$$

First introduce discrete action,
then derive the least action paths

✓ Energy conservation
in “asteroid belts”



$\Delta t = 0.9$

Pendulum equation

Example: pendulum equation.

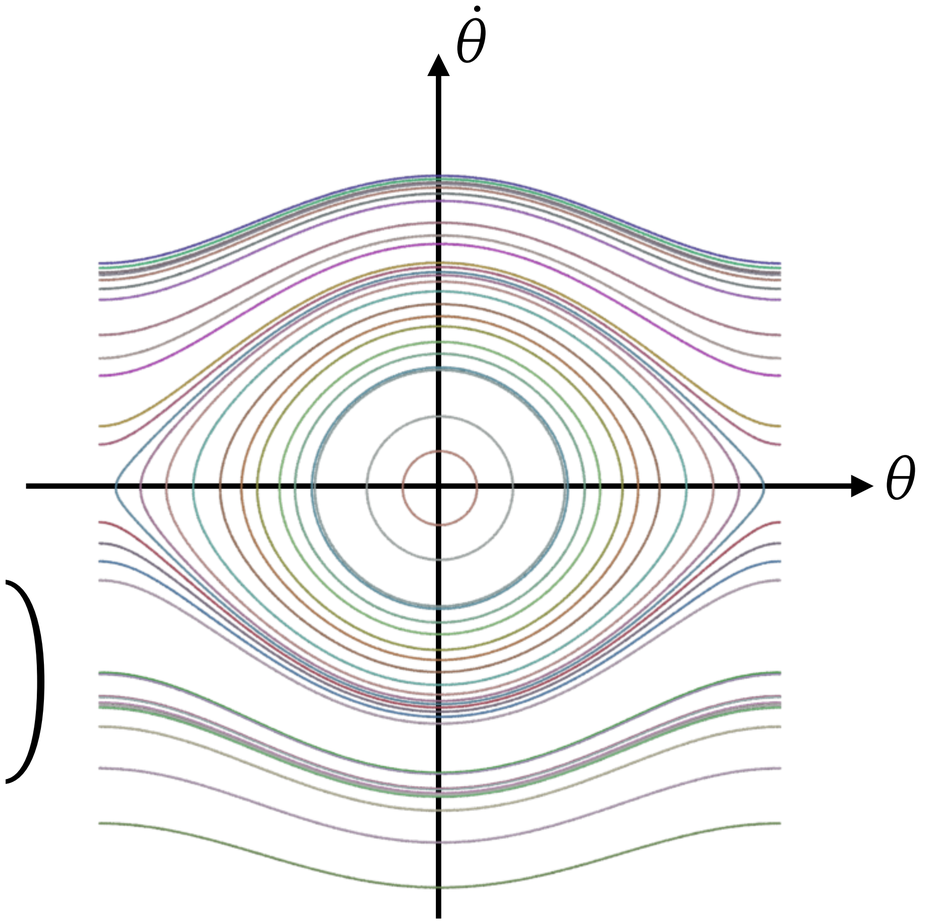
$$m\ddot{\theta} = -mg\sin\theta$$

- Another 2nd order discretization

$$\frac{\theta_{i-1} - 2\theta_i + \theta_{i+1}}{\Delta t^2} = -\sin\theta_i$$

$$\frac{\theta_{i-1} - 2\theta_i + \theta_{i+1}}{\Delta t^2} = 4\arg\left(1 + \frac{\Delta t^2}{4} e^{-\mathbf{i}\theta_i}\right)$$

✓ Integrable system



$\Delta t = 0.1$

Pendulum equation

Example: pendulum equation.

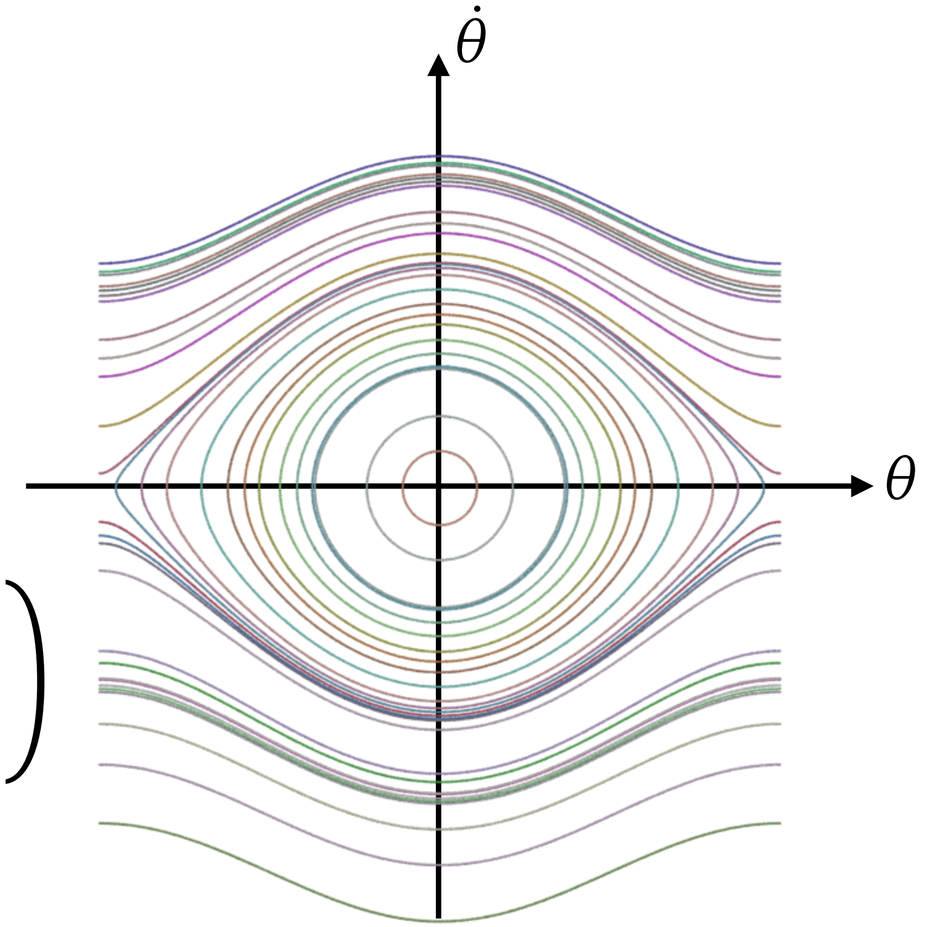
$$m\ddot{\theta} = -mg\sin\theta$$

- Another 2nd order discretization

$$\frac{\theta_{i-1} - 2\theta_i + \theta_{i+1}}{\Delta t^2} = -\sin\theta_i$$

$$\frac{\theta_{i-1} - 2\theta_i + \theta_{i+1}}{\Delta t^2} = 4\arg\left(1 + \frac{\Delta t^2}{4} e^{-\mathbf{i}\theta_i}\right)$$

✓ Integrable system



$\Delta t = 0.9$

Pendulum equation

general

no-structure

quantitative
high-precision

4th order Runge–Kutta

Variational integrator

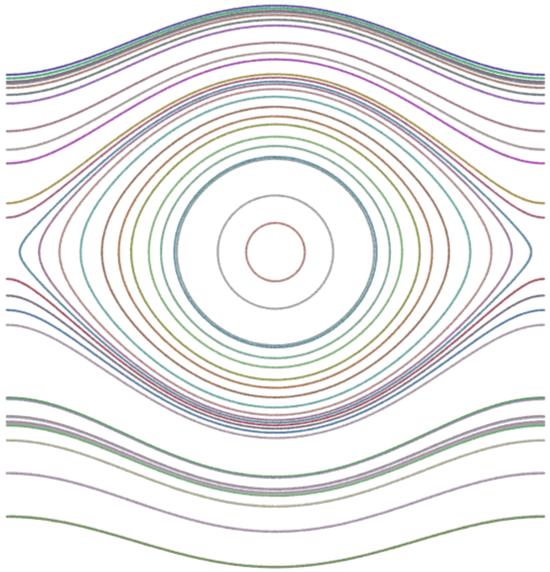
Discrete integrable system

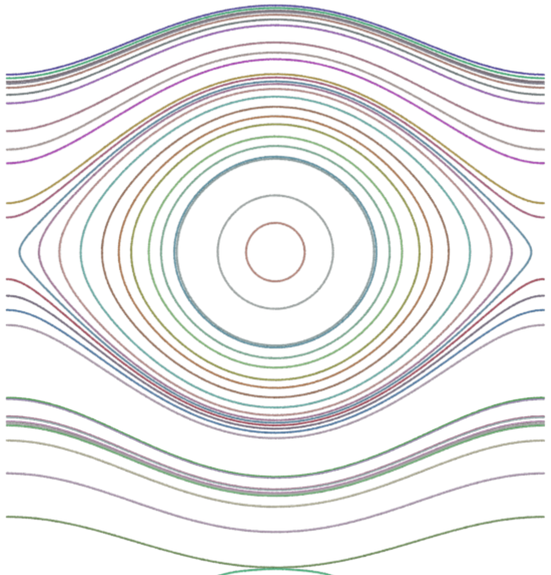
rare

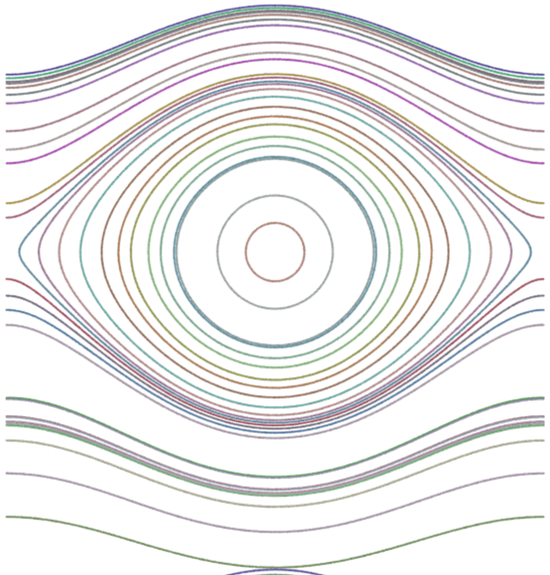
structure
preserving

qualitative
exact

$\Delta t = 0.1$







$\Delta t = 0.9$

