



autorob.org

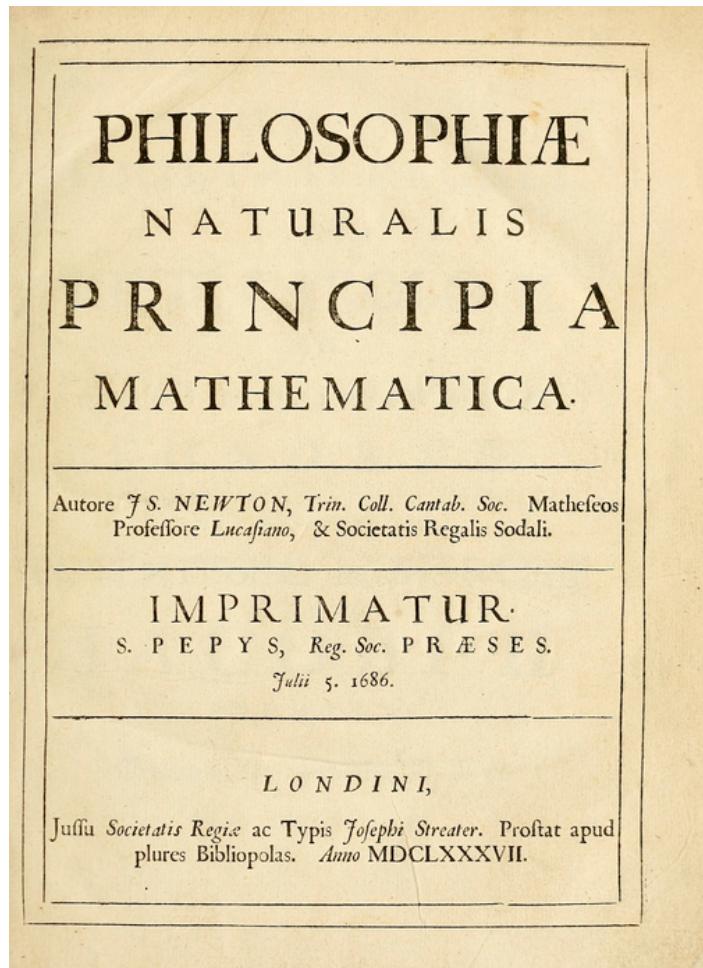
Dynamics and Numerical Integration

EECS 398
Intro. to Autonomous Robotics

ME/EECS 567 ROB 510
Robot Modeling and Control

Fall 2018

1687



2012



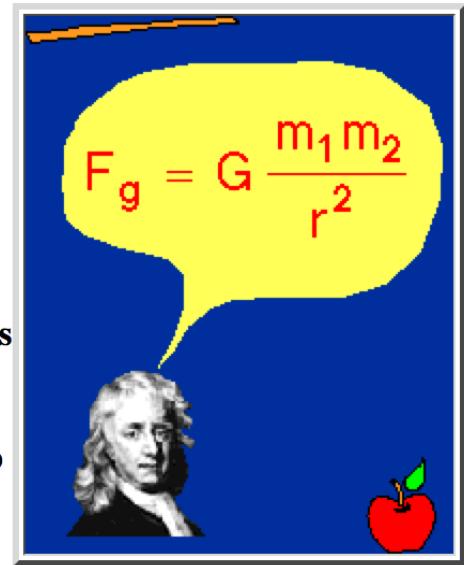
<http://drawception.com/viewgame/azFl9y6c8s/edible-science/>

UM EECS 398/598 - autorob.github.io

There is a popular story that Newton was sitting under an apple tree, an apple fell on his head, and he suddenly thought of the Universal Law of Gravitation. As in all such legends, this is almost certainly not true in its details, but the story contains elements of what actually happened.

What Really Happened with the Apple?

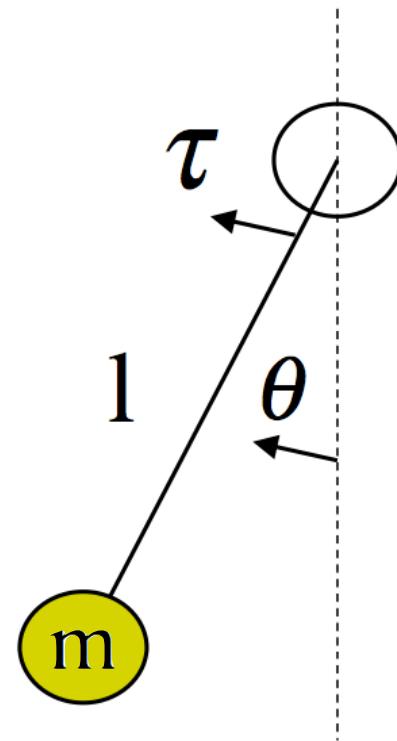
Probably the more correct version of the story is that Newton, upon observing an apple fall from a tree, began to think along the following lines: The apple is accelerated, since its velocity changes from zero as it is hanging on the tree and moves toward the ground. Thus, by Newton's 2nd Law there must be a force that acts on the apple to cause this acceleration. Let's call this force "gravity", and the associated acceleration the "acceleration due to gravity". Then imagine the apple tree is twice as high. Again, we expect the apple to be accelerated toward the ground, so this suggests that this force that we call gravity reaches to the top of the tallest apple tree.



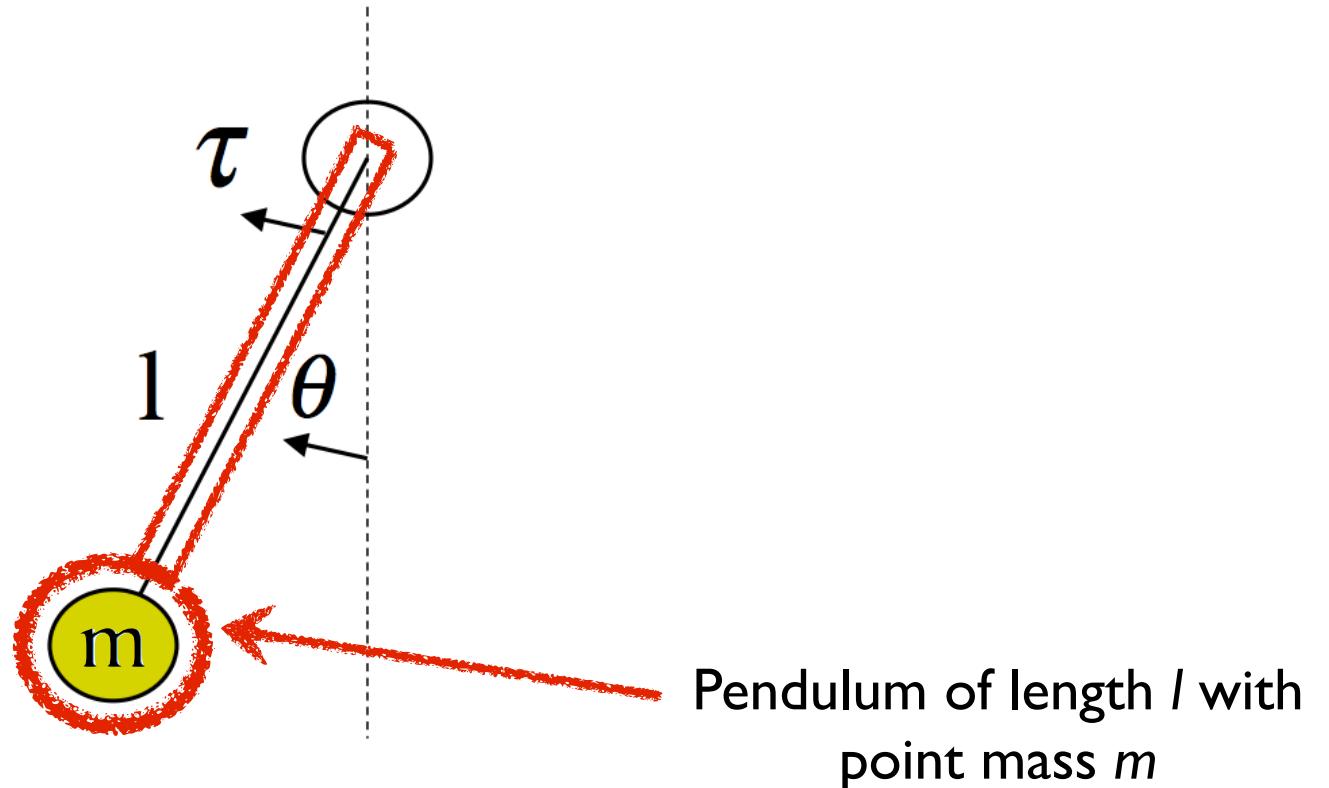
Administrivia

- Project 1 (“Path Planning”) due next Monday Sep 24
 - “git pizza” extended office hours Friday 5-7pm, Beyster 3725
- If you are not on course slack group, email me (ocj)
- Send me point to your git repository (and your picture)
- Anyone still looking for permission to enroll from waitlist?

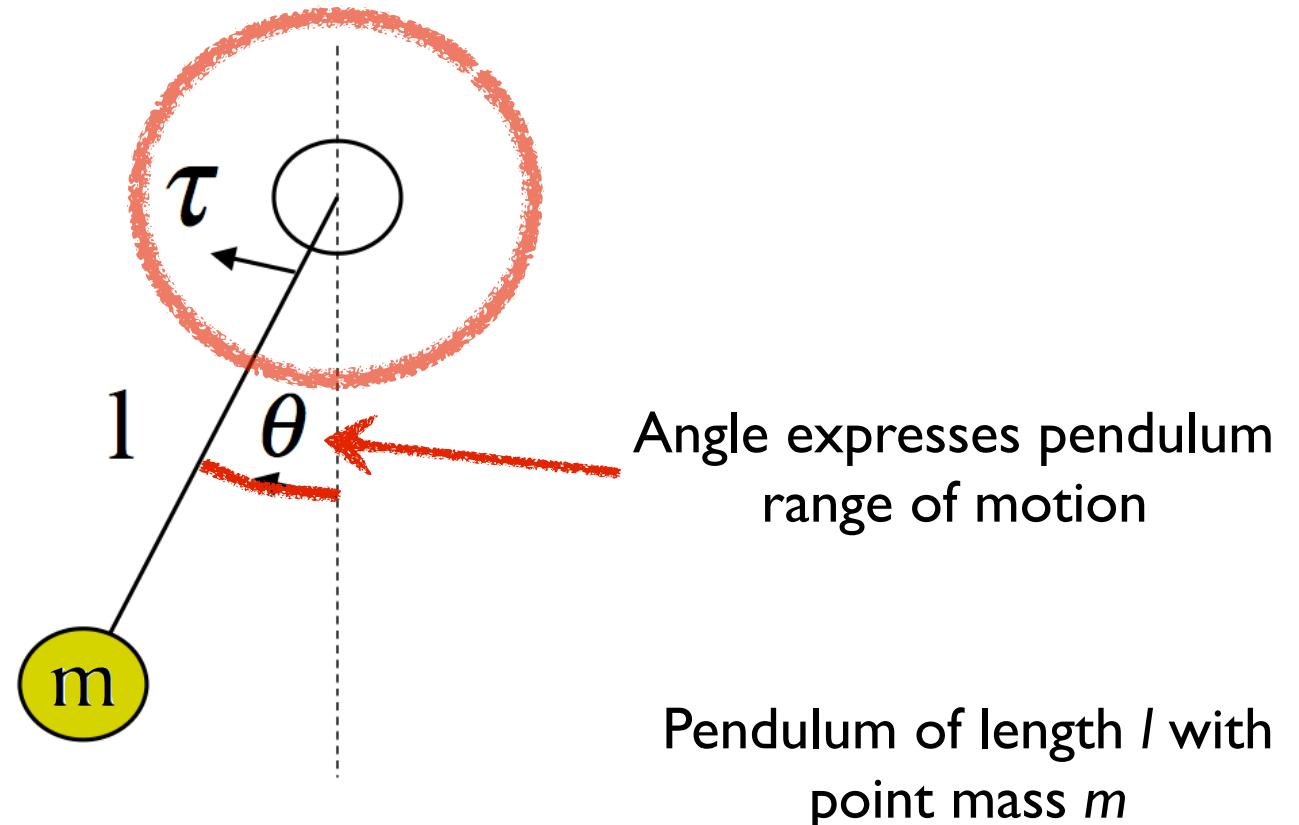
Example: Motorized Pendulum



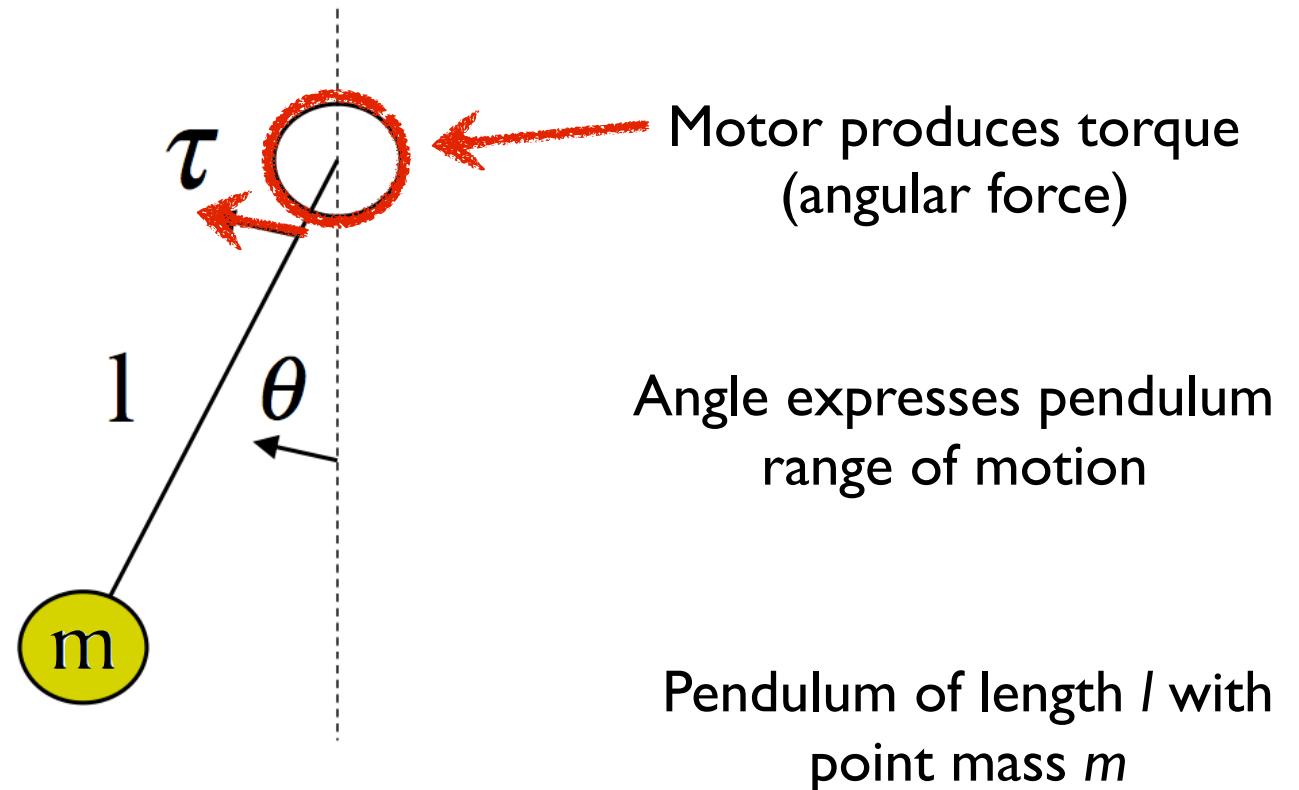
Example: Motorized Pendulum



Example: Motorized Pendulum



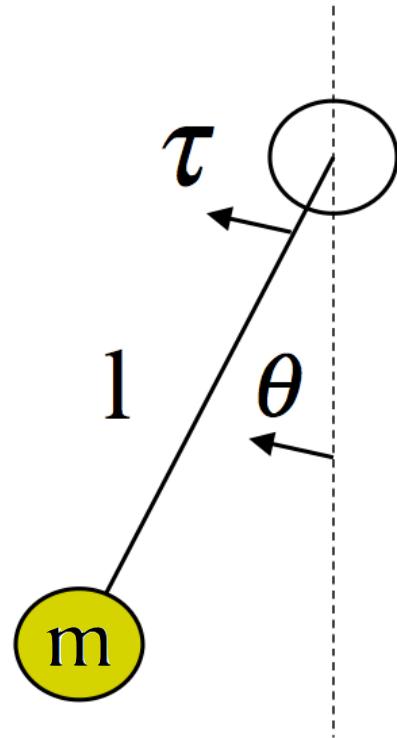
Example: Motorized Pendulum



Example: Motorized Pendulum

Equation of motion
(with rotational inertia I)

$$I\ddot{\theta} = -mgl \sin(\theta) + \tau$$



Motor produces torque
(angular force)

Angle expresses pendulum
range of motion

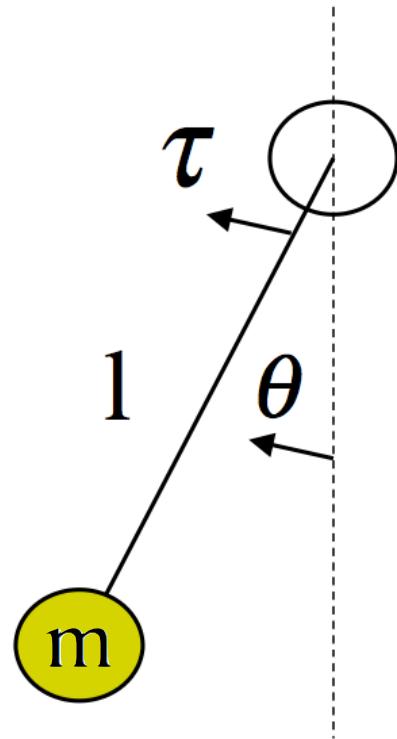
Pendulum of length l with
point mass m

Dynamics

Equation of motion
(with rotational inertia I)

$$I\ddot{\theta} = -mgl \sin(\theta) + \tau$$

Motorized Pendulum



Controls

Motor produces torque
(angular force)

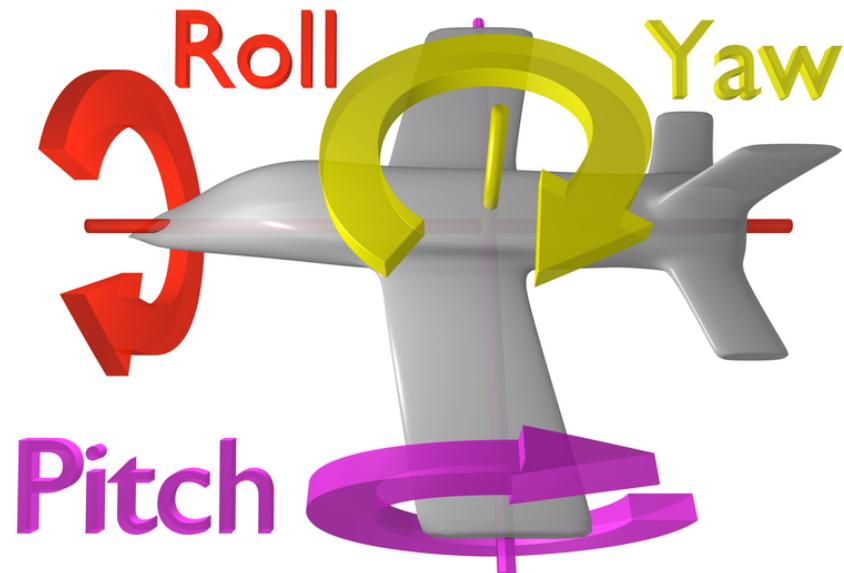
State (or Configuration)
Angle expresses pendulum
range of motion

System

Pendulum of length l with
point mass m

Defining State

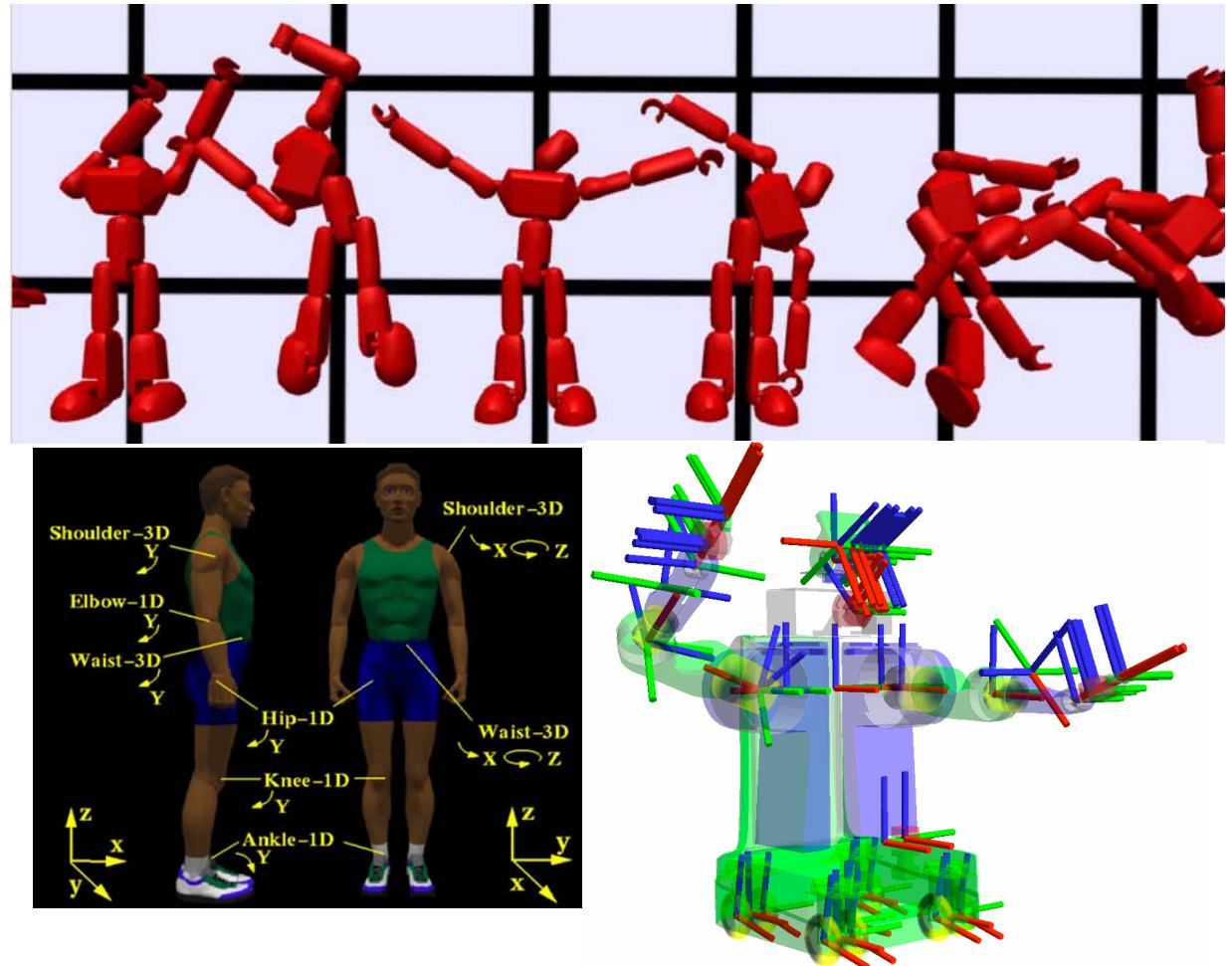
- State comprised of degrees-of-freedom (DOFs)
- DOFs describe translational and rotational axes for joints
- How many DOFs in the example pendulum?
- Airplane DOFs?



Note: particles vs. rigid bodies, COM, geom, inertia
Note: types of joints, global vs. local DOFs

Defining State

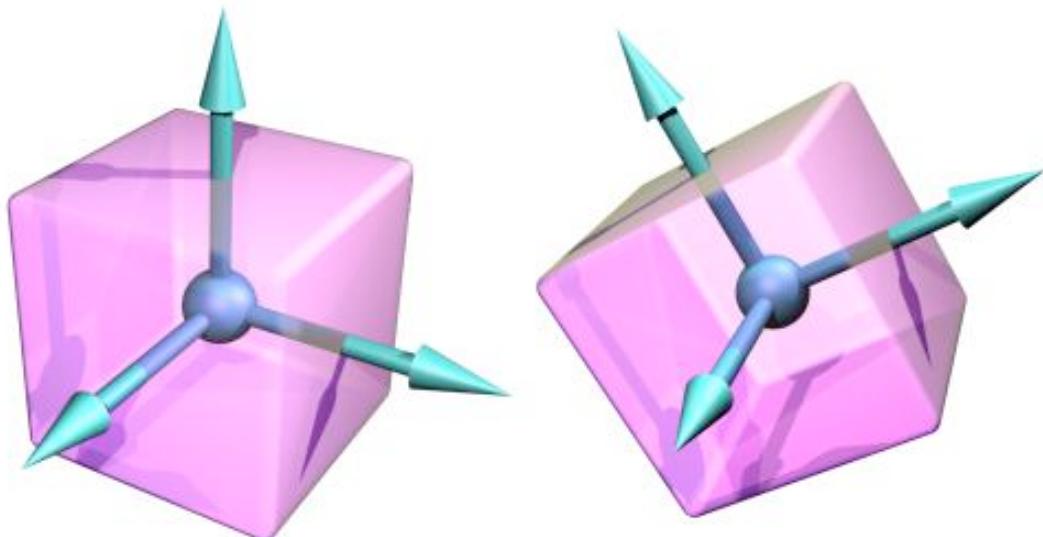
- State comprised of degrees-of-freedom (DOFs)
- DOFs describe translational and rotational axes of system
- Humanoid DOFs?
 - joint angles
 - global positioning



DOFs and Coordinate Spaces

- Each body has its own coordinate system

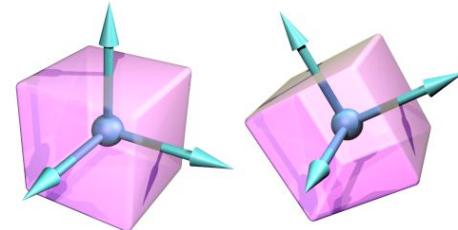
Link frame



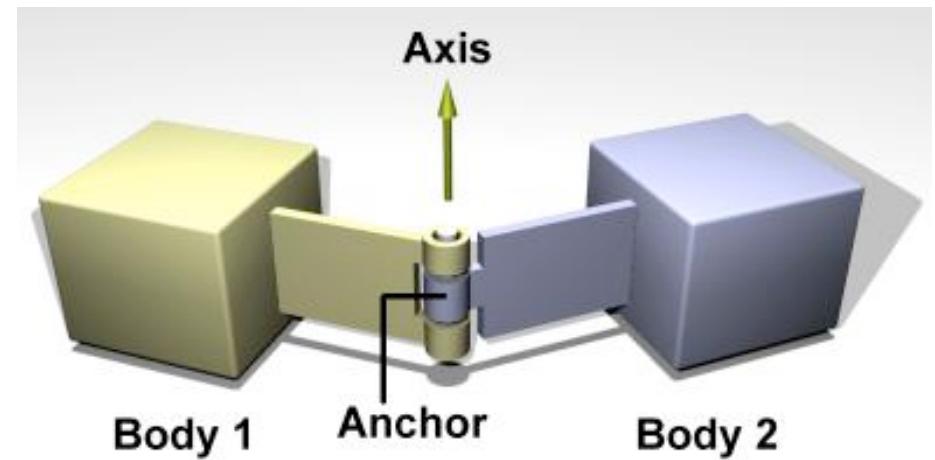
DOFs and Coordinate Spaces

- Each body has its own coordinate system

 Link  frame

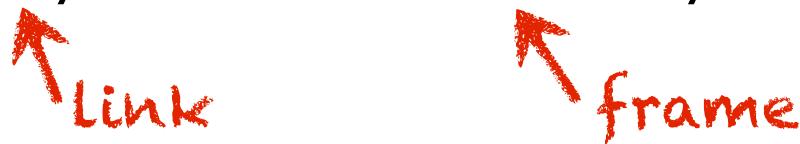


- Joints connect two links (rigid bodies)
- Hinge (1 rotational DOF)

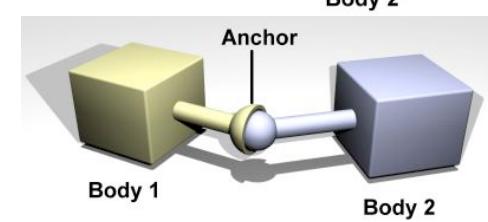
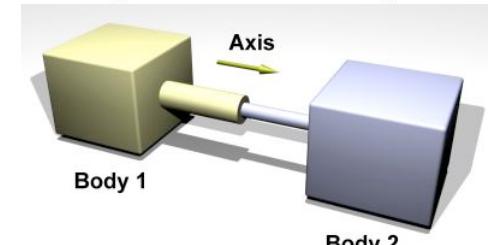
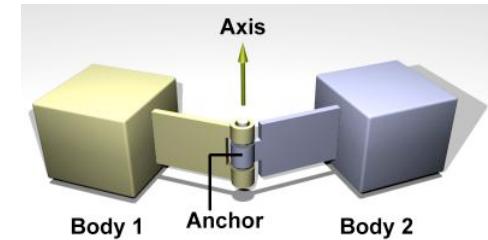
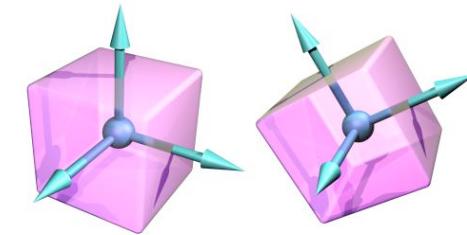


DOFs and Coordinate Spaces

- Each body has its own coordinate system

Link frame

- Joints connect two links (rigid bodies)
 - Hinge (1 rotational DOF)
 - Prismatic (1 translational DOF)
 - Ball-socket (3 DOFs)

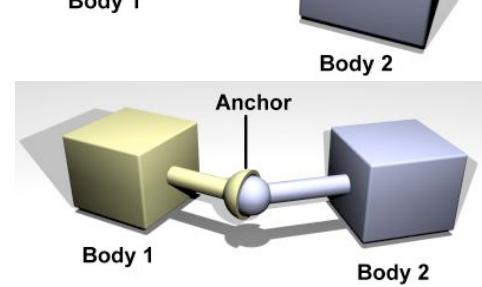
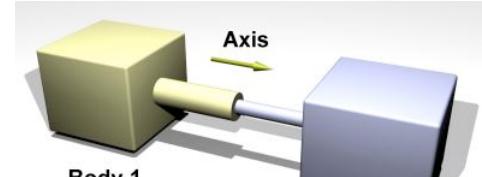
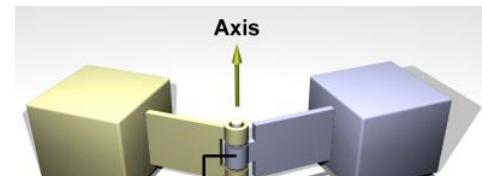
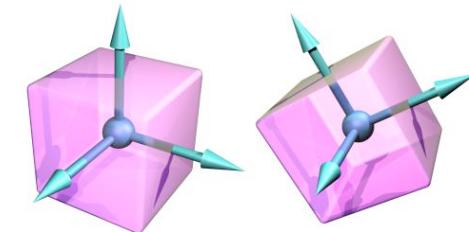


DOFs and Coordinate Spaces

- Each body has its own coordinate system



- Joints connect two links (rigid bodies)
 - Hinge (1 rotational DOF)
 - Prismatic (1 translational DOF)
 - Ball-socket (3 DOFs)
- A motor exerts force along a DOF axis



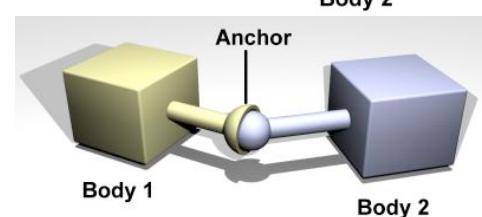
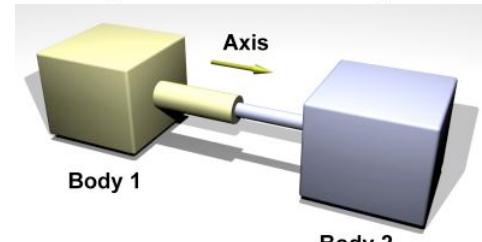
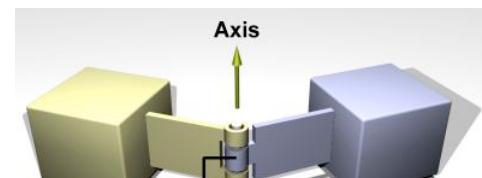
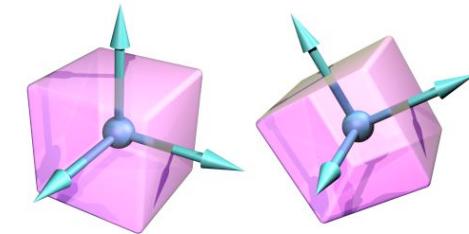
DOFs and Coordinate Spaces

- Each body has its own coordinate system

 Link frame

- Joints connect two links (rigid bodies)

- Hinge (1 rotational DOF)
 - Prismatic (1 translational DOF)
 - Ball-socket (3 DOFs)
-
- A motor exerts force along a DOF axis
 - Linear transformations used to relate coordinate systems of bodies and joints
 - Spatial geometry attached to each link, but does not affect the body's coordinate frame



Robotic machines are comprised of N joints and $N+1$ links

Joints and links form a tree hierarchy of articulated motion

The “base” is the root link of this hierarchy

A link has one parent (inboard) joint and potentially zero, one, or multiple child (outboard) joints

A serial chain is a robot where every link has only one child joint

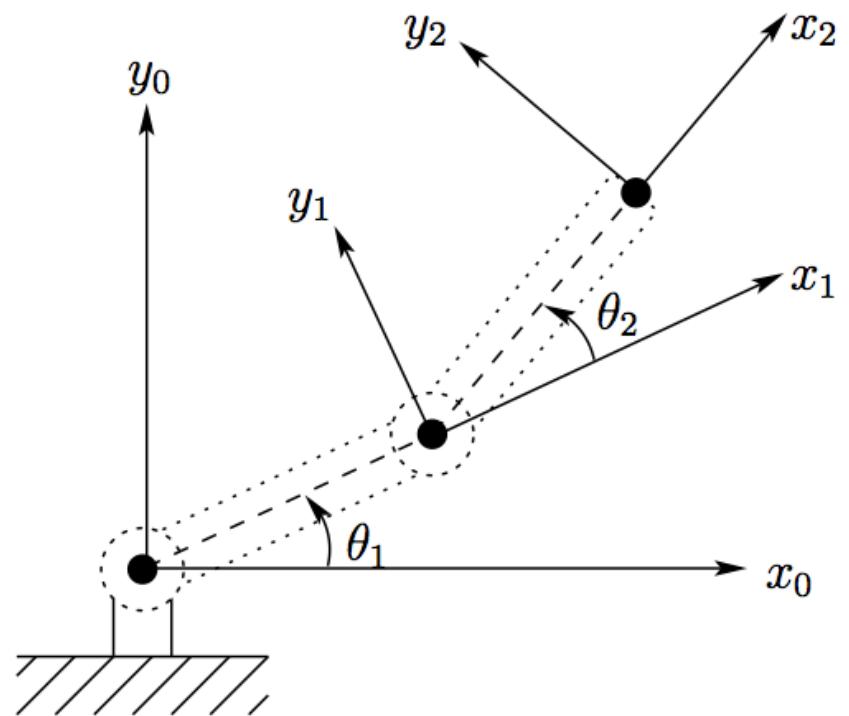
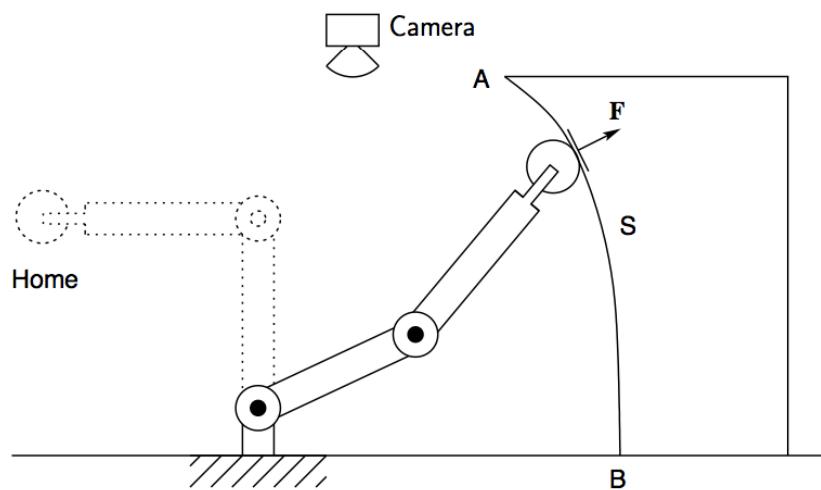
DLR Lightweight arm



Dynamixel robot kit

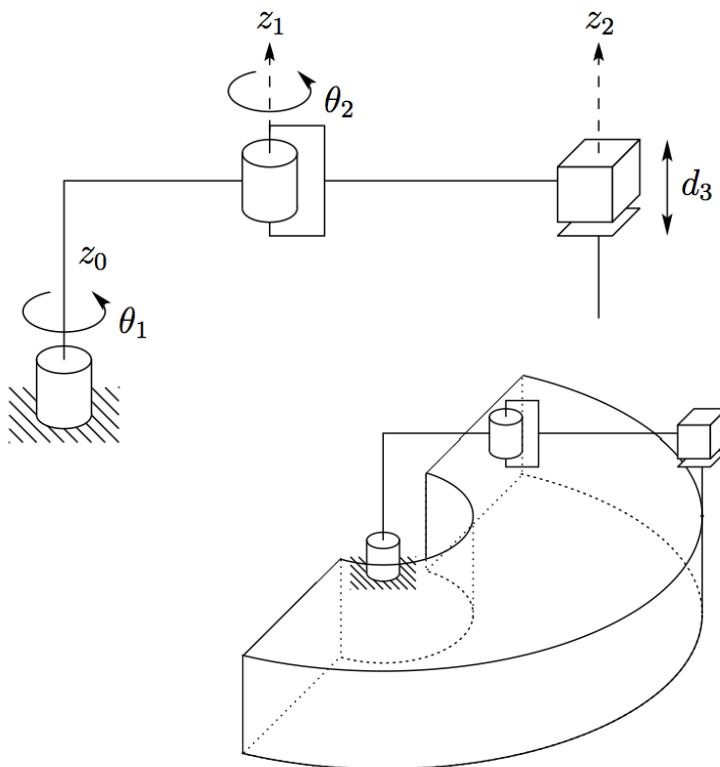


Planar 2-DOF 2-link Arm

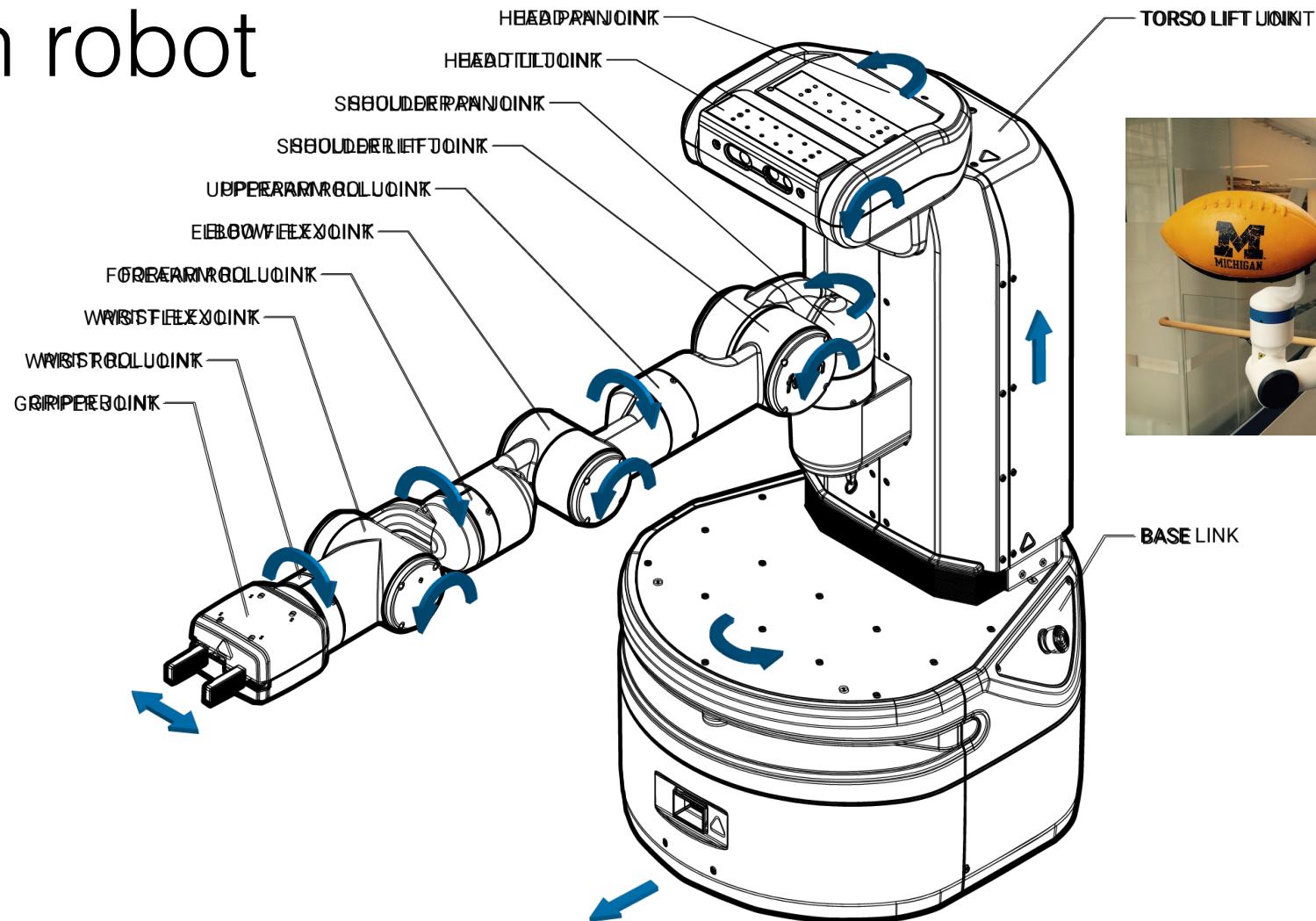


SCARA Arm

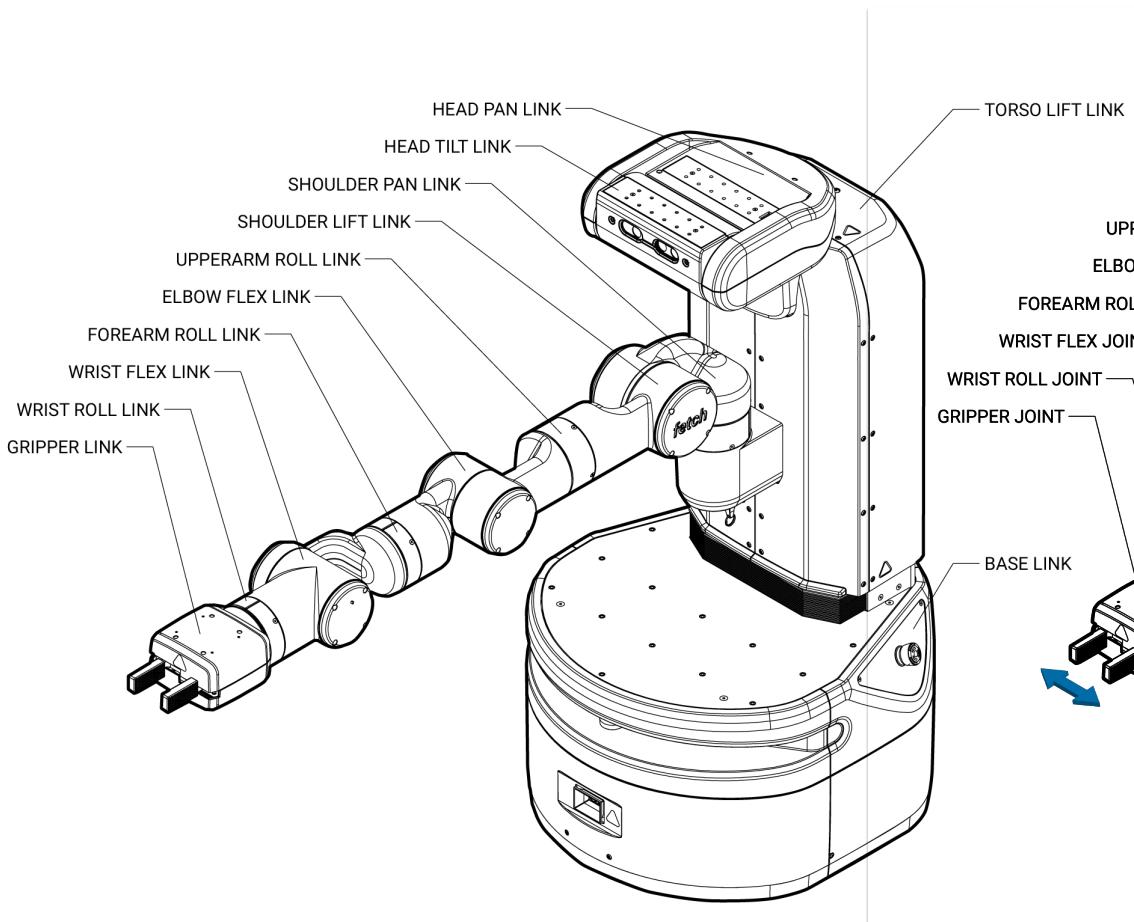
Selective Compliance Assembly Robot Arm



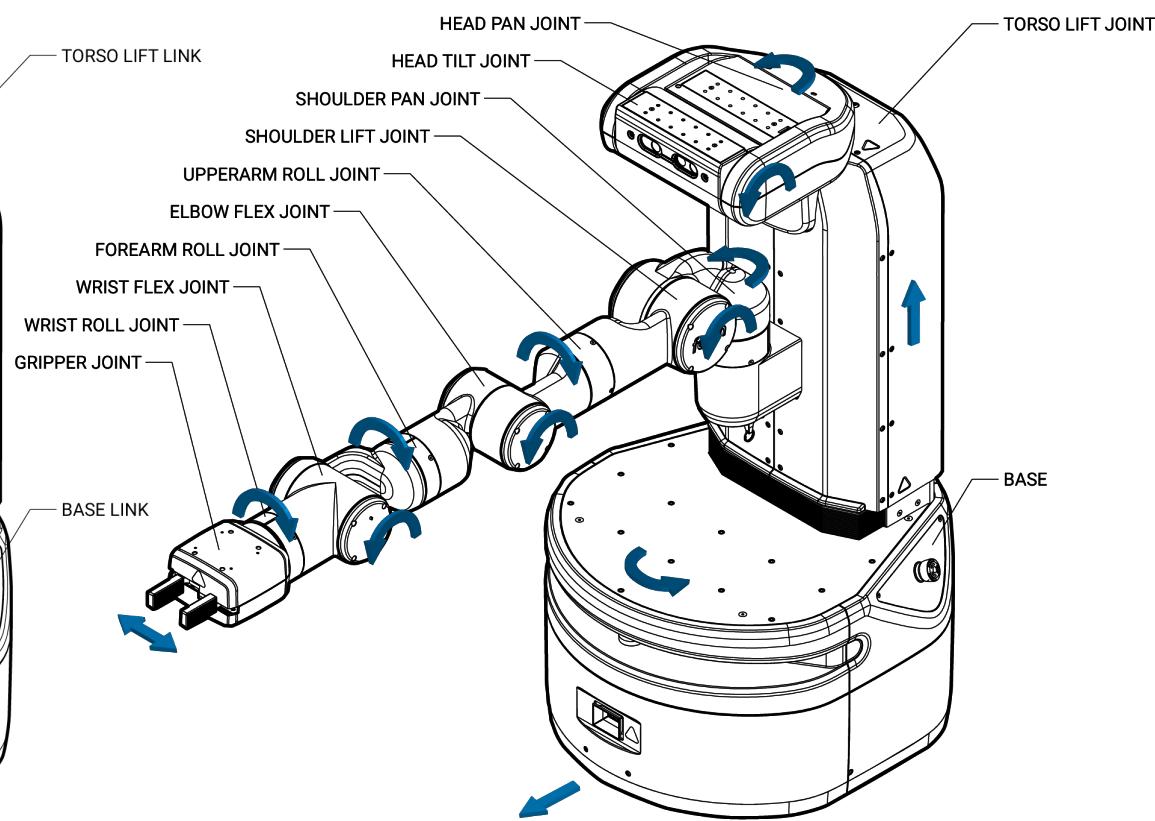
Fetch robot

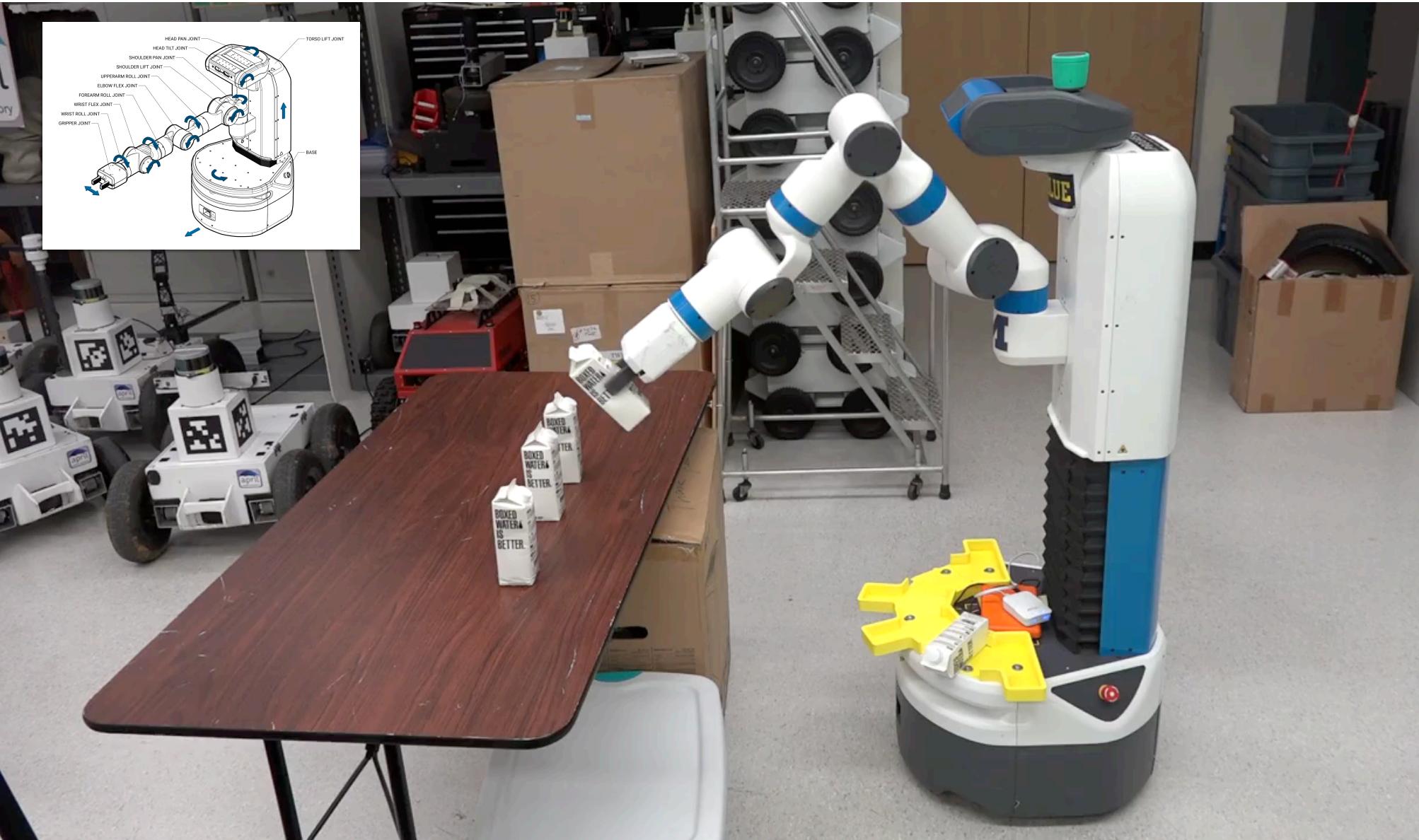
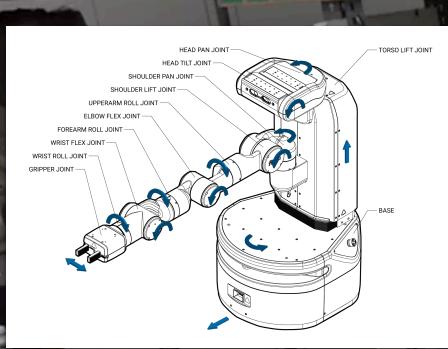


Fetch links

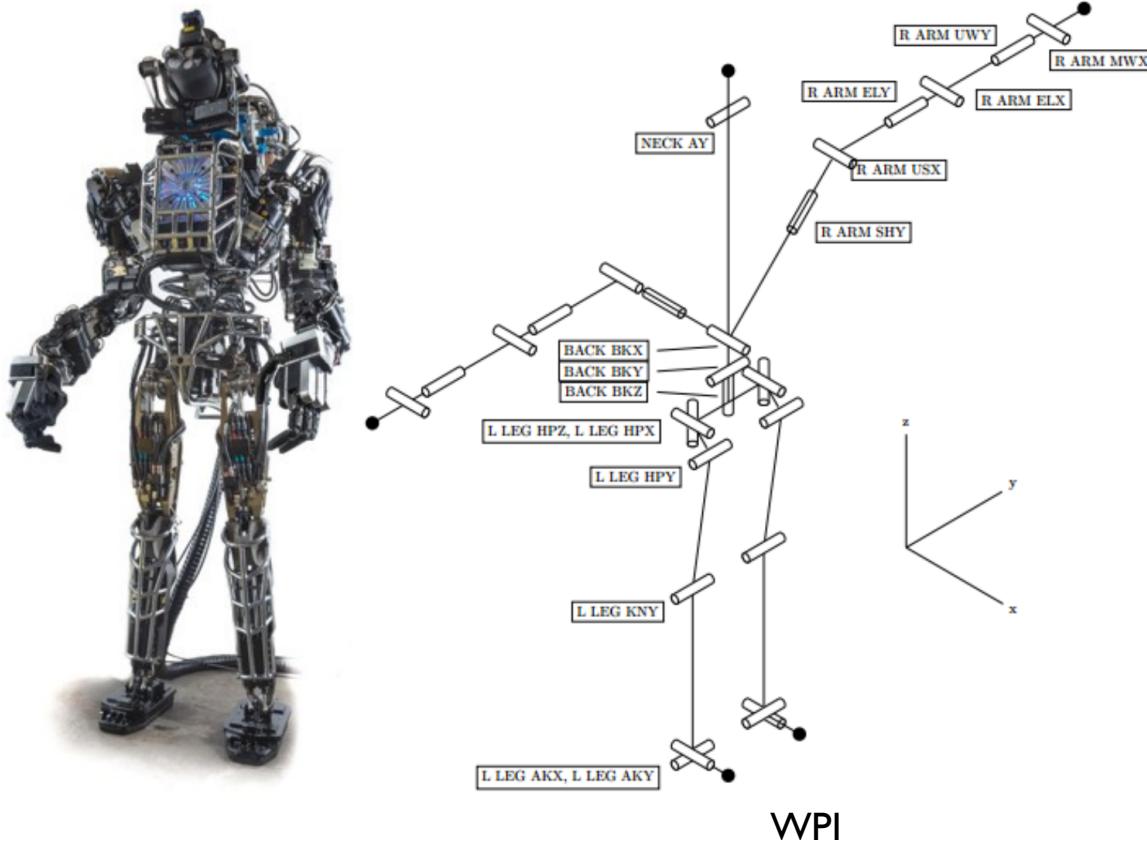


Fetch joints





Atlas robot



WPI

UM EECS 398/598 - autorob.github.io

Physical Simulation (Gazebo)

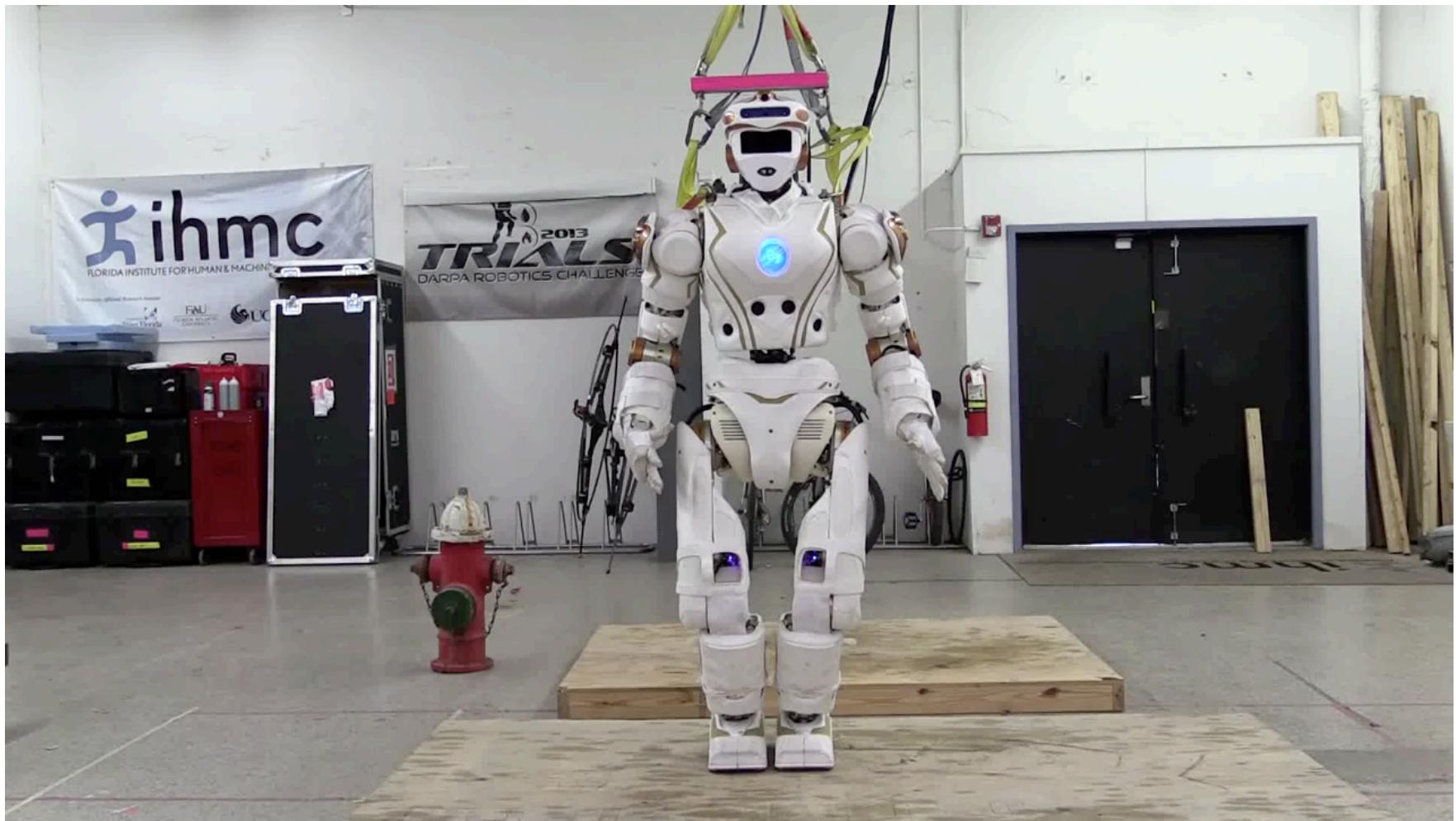


Why simulate robots?

- Real robots are expensive
 - Improper controllers can physically break robots
 - Inexpensive to experiment and test robot controllers in simulation
- Predictive model of dynamics
 - Necessary for some types of control (e.g., optimal control)

$$J = \Phi [\mathbf{x}(t_0), t_0, \mathbf{x}(t_f), t_f] + \boxed{\int_{t_0}^{t_f} \mathcal{L} [\mathbf{x}(t), \mathbf{u}(t), t] \, dt}$$

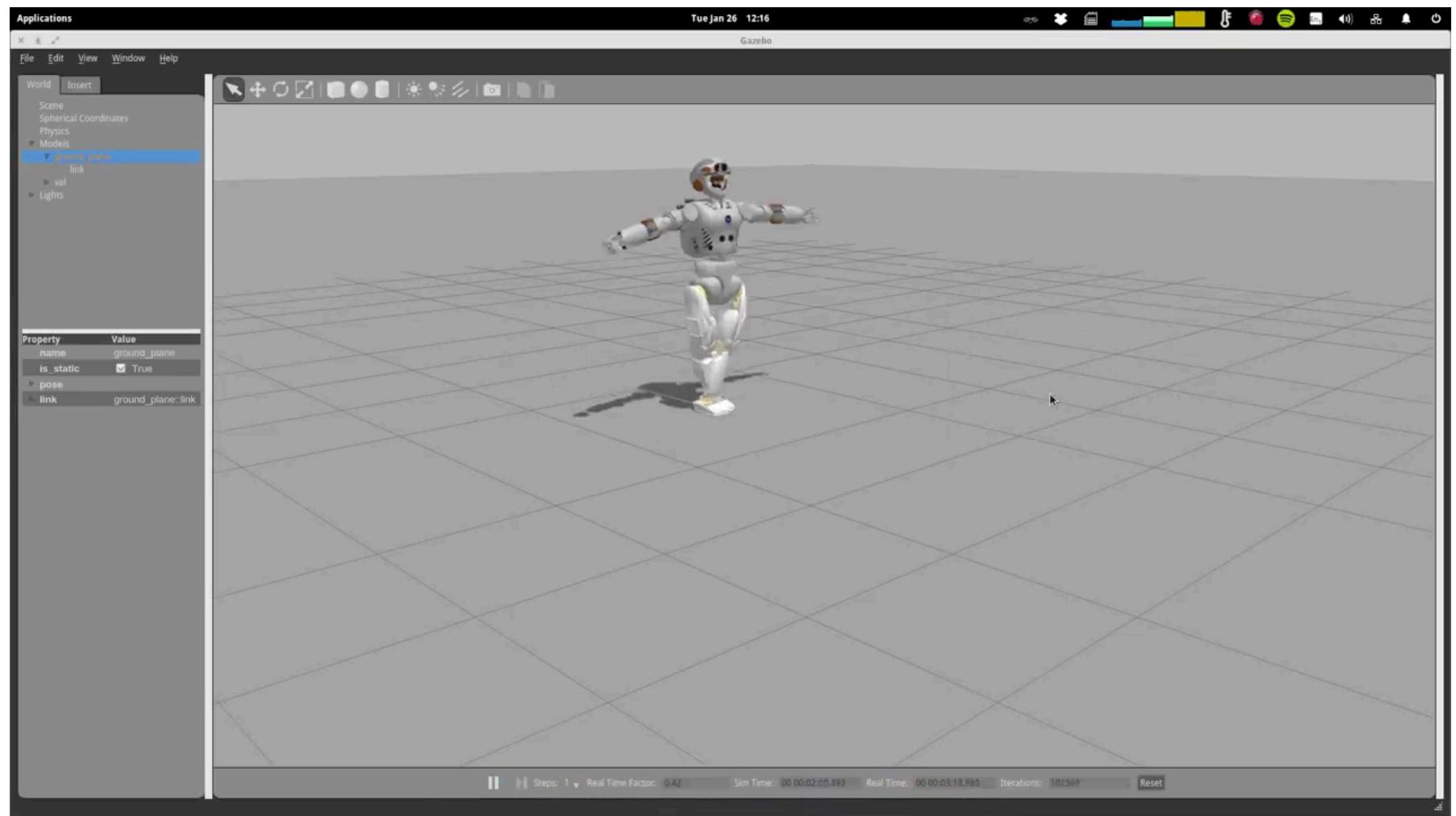
For example,
NASA Valkyrie...



Valkyrie for real - <https://youtu.be/5Ee5u2ekE8c>

UM EECS 398/598 - autorob.github.io

NASA Valkyrie...
started development in simulation



Valkyrie in simulation - <https://youtu.be/tLCpJvqgtRQ>

UM EECS 398/598 - autorob.github.io

Our first robot: Pendulum!

Project 2: 1 DOF Pendularm

← → C file:///Users/logan/git_tmp/kineval/pendularm/pendularm1.html ⭐ ⚓

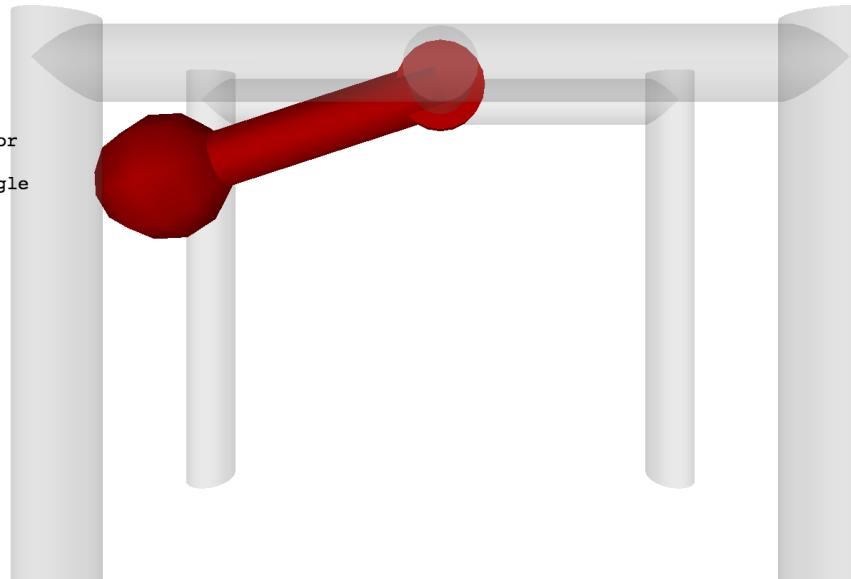
```
System
t = 162.00 dt = 0.05
integrator = velocity verlet
x = -1.26
x_dot = -0.00
x_desired = -1.26
```

```
Servo: active
u = -37.32
kp = 1500.00
kd = 15.00
ki = 150.10
```

```
Pendulum
mass = 2.00
length = 2.00
gravity = 9.81
```

```
Keys
[0-4] - select integrator
a/d - apply user force
q/e - adjust desired angle
c - toggle servo
s - disable servo
```

model, simulate, and control
1 DoF robot arm

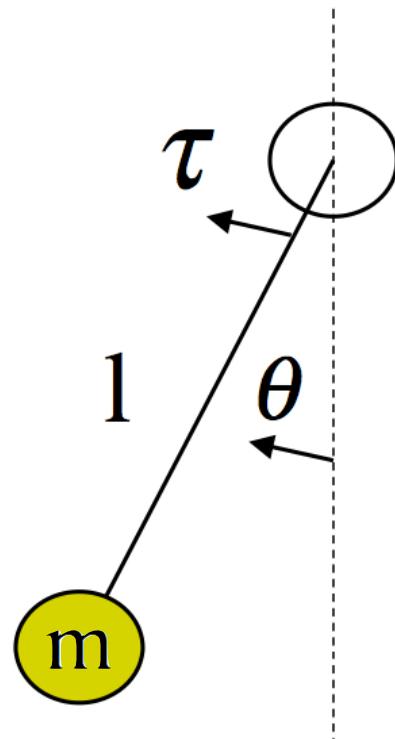


Dynamics

Equation of motion
(with rotational inertia I)

$$I\ddot{\theta} = -mgl \sin(\theta) + \tau$$

Example: Pendulum



Controls

Motor produces torque
(angular force)

State (or Configuration)
Angle expresses pendulum
range of motion

System

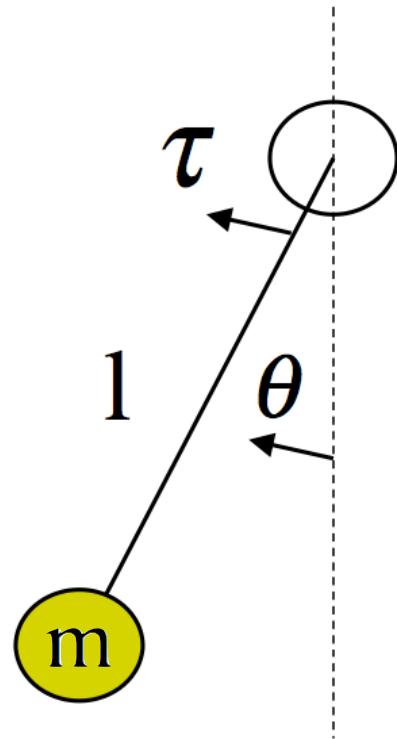
Pendulum of length l with
point mass m

Example: Pendulum

Equation of motion
(with rotational inertia I)

$$I\ddot{\theta} = -mgl \sin(\theta) + \tau$$

motor torque



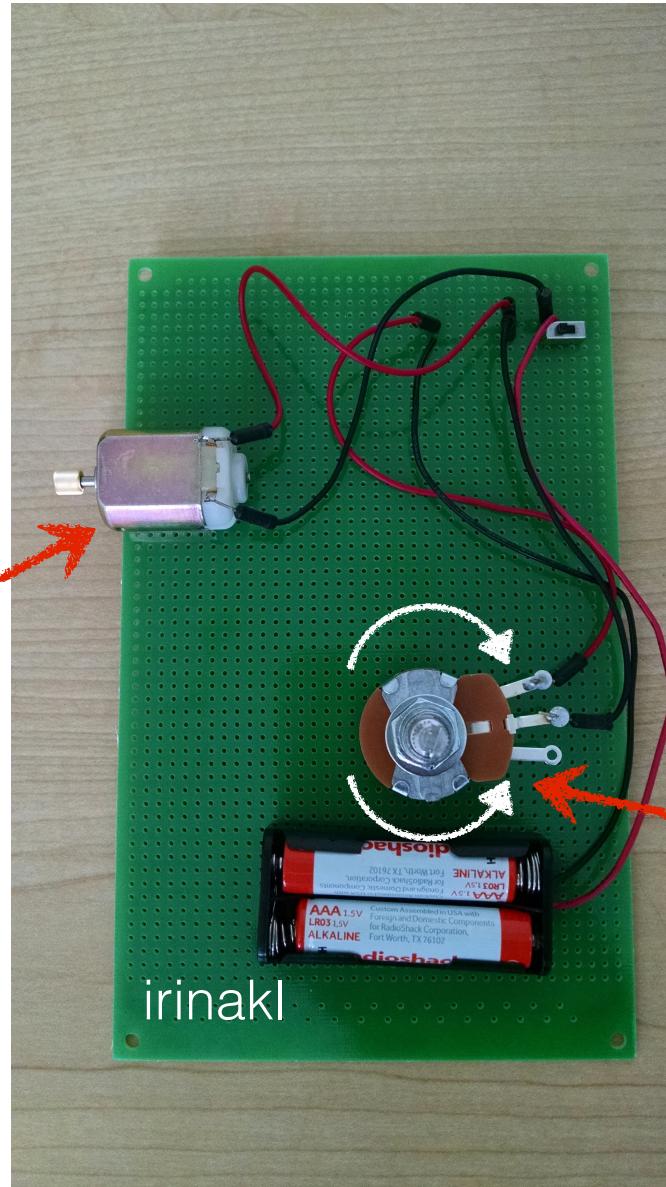
Motor produces torque
(angular force)

Angle expresses pendulum
range of motion

Pendulum of length l with
point mass m

Example device

DC motor

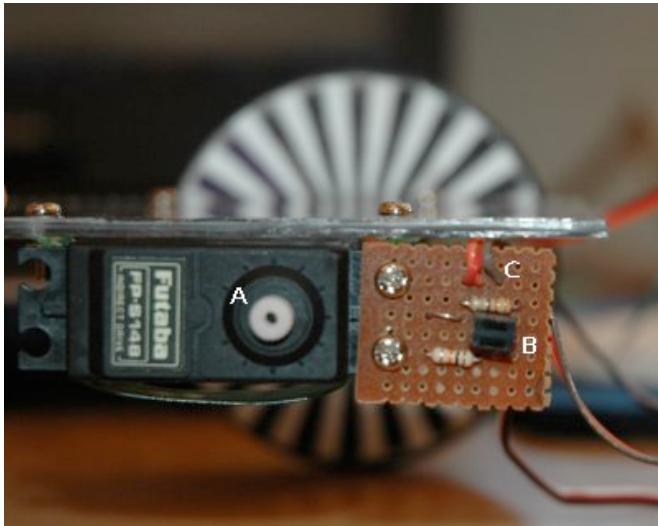


Control torque of motor
as a proportion of
current applied

Decrease resistance
to increase force

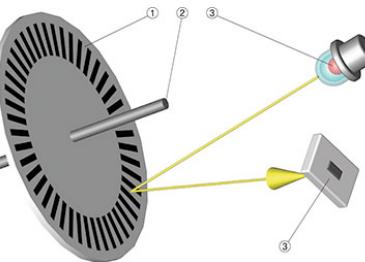
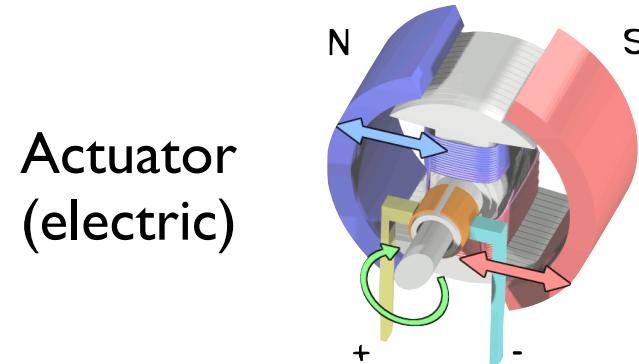
Potentiometer

Increase resistance
to decrease force



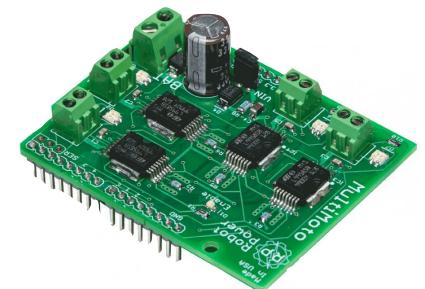
Servos have:

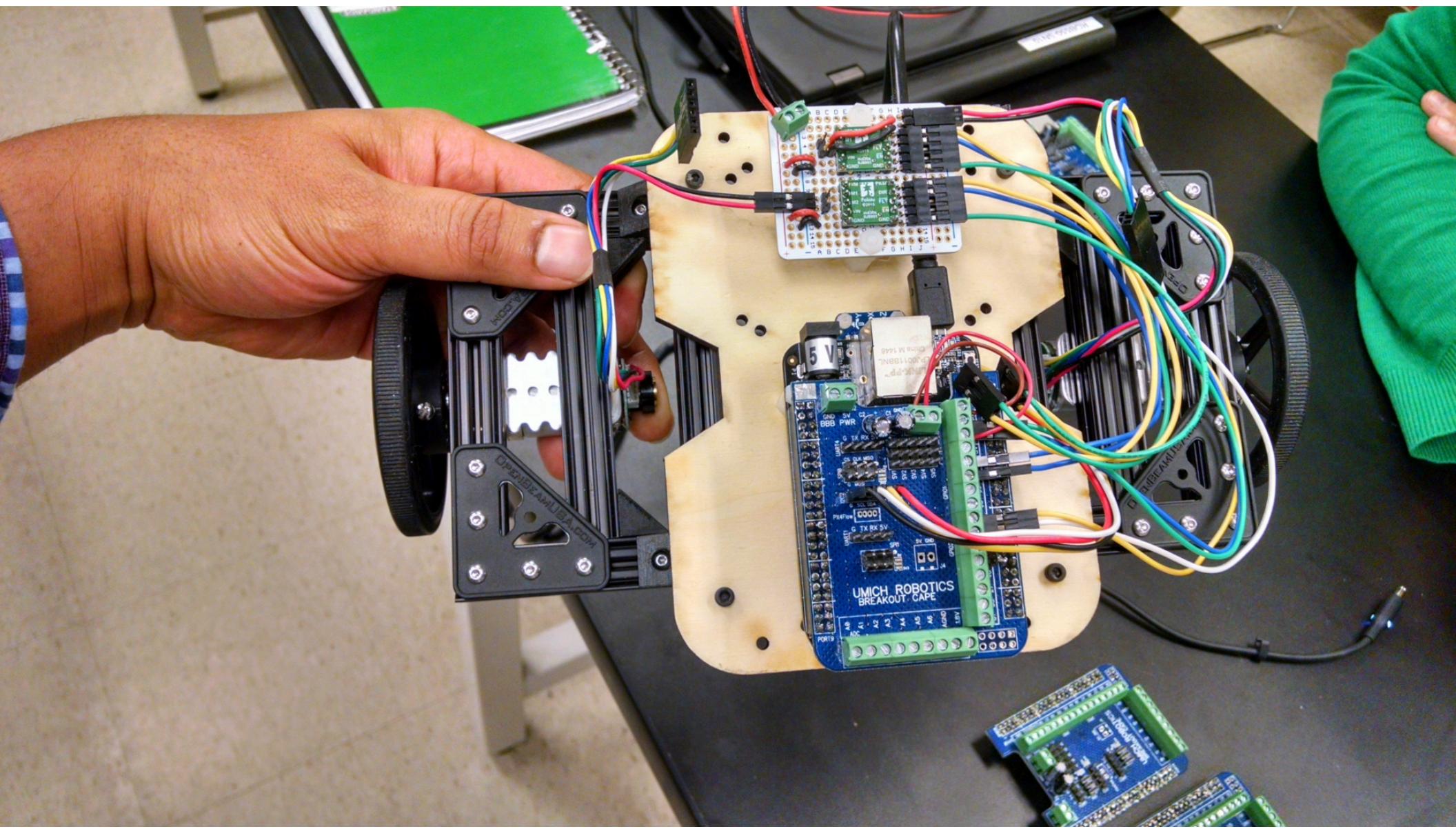
- actuators to produce motion
- proprioception to sense pose
- controller to regulate current to motor



Proprioception
(optical encoder)

Controller
(4 channel H-Bridges)



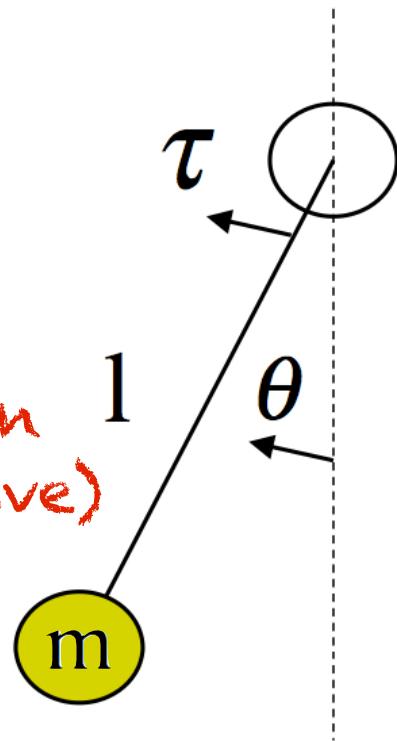


Example: Pendulum

Equation of motion
(with rotational inertia I)

$$I\ddot{\theta} = -mgl \sin(\theta) + \tau$$

↑
angular acceleration
(second time derivative)



Motor produces torque
(angular force)

Angle expresses pendulum
range of motion

Pendulum of length l with
point mass m

Derivative

From Wikipedia, the free encyclopedia

The **derivative** of a function of a real variable measures the sensitivity to change of a quantity (a function or **dependent variable**) which is determined by another quantity (the **independent variable**).

First derivative

$$f'(a) = \lim_{h \rightarrow 0} \frac{f(a+h) - f(a)}{h}.$$

with approximation after step length h

$$f(a+h) \approx f(a) + f'(a)h$$

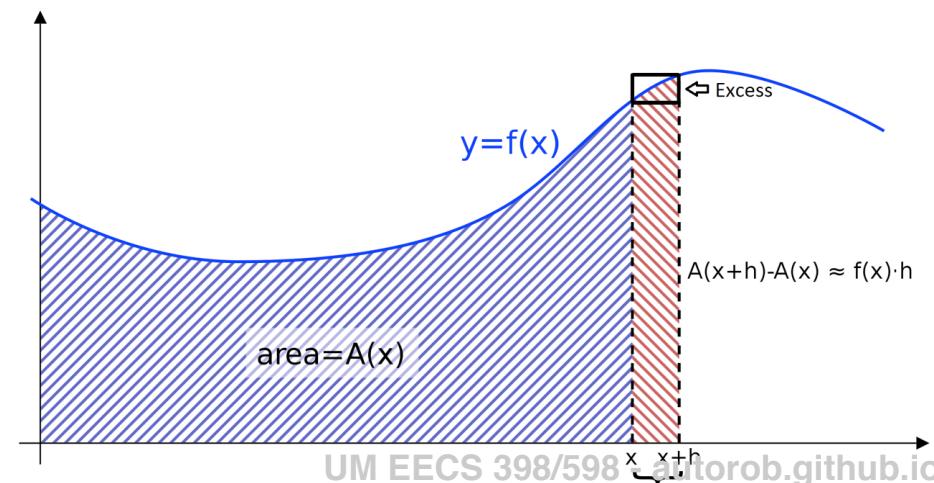
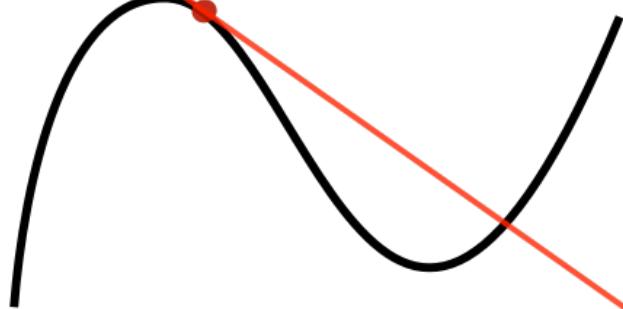
Second derivative

$$f''(a) = \lim_{h \rightarrow 0} \frac{f'(a+h) - f'(a)}{h}$$

Integration inverse operation of differentiation

$$\frac{d}{dx} f(x) = g(x), \int g(x) dx = f(x) + C$$

$$m = \frac{\Delta f(a)}{\Delta a} = \frac{f(a+h) - f(a)}{(a+h) - (a)} = \frac{f(a+h) - f(a)}{h}.$$



Differential equation

From Wikipedia, the free encyclopedia

A **differential equation** is a [mathematical equation](#) that relates some [function](#) with its [derivatives](#). In applications, the functions usually represent physical quantities, the derivatives represent their rates of change, and the equation defines a relationship between the two. Because such relations are extremely common, differential equations play a prominent role in many disciplines including [engineering](#), [physics](#), [economics](#), and [biology](#).

Power rule

$$\frac{d}{dx}x^n = nx^{n-1}, \quad n \neq 0.$$

Product rule

$$\frac{d}{dx}(u \cdot v) = u \cdot \frac{dv}{dx} + v \cdot \frac{du}{dx}$$

Chain rule

$$[f(g(x))]' = f'(g(x))g'(x)$$

Time derivatives (velocity, acceleration)

$$\frac{dx}{dt} \text{ or } \dot{x} \quad \frac{d^2x}{dt^2} \text{ or } \ddot{x}$$

Partial derivative

$$f'_x, f_x, \partial_x f, \frac{\partial}{\partial x} f, \text{ or } \frac{\partial f}{\partial x}.$$

Total derivative

$$\frac{df}{dt} = \frac{\partial f}{\partial t} \frac{dt}{dt} + \frac{\partial f}{\partial x} \frac{dx}{dt} + \frac{\partial f}{\partial y} \frac{dy}{dt}$$

Trigonometry

From Wikipedia, the free encyclopedia

Trigonometry (from Greek *trigōnon*, "triangle" and *metron*, "measure"^[1]) is a branch of **mathematics** that studies relationships involving lengths and **angles** of triangles. The field emerged during the 3rd century BC from applications of **geometry** to astronomical studies.^[2]

Math.sin(A)

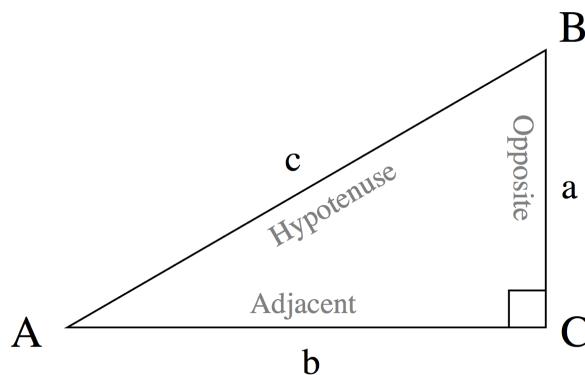
$$\sin A = \frac{\text{opposite}}{\text{hypotenuse}} = \frac{a}{c}.$$

Math.cos(A)

$$\cos A = \frac{\text{adjacent}}{\text{hypotenuse}} = \frac{b}{c}.$$

Math.tan(A)

$$\tan A = \frac{\text{opposite}}{\text{adjacent}} = \frac{a}{b} = \frac{\sin A}{\cos A}$$



Angles A, B, C
Lengths a, b, c

Math.atan2(a,b)

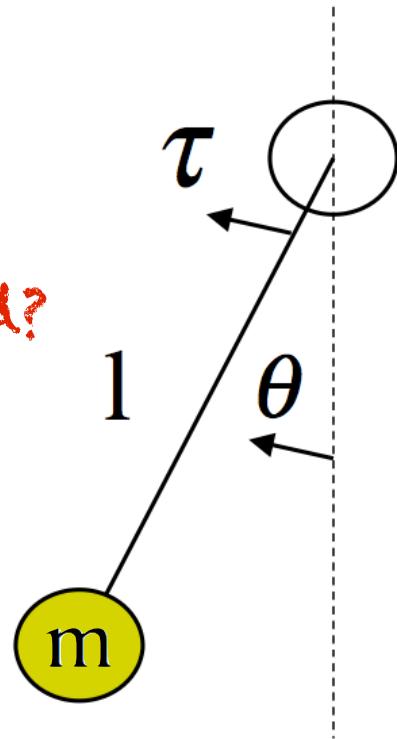
$$\text{atan2}(y, x) = \begin{cases} \arctan \frac{y}{x} & x > 0 \\ \arctan \frac{y}{x} + \pi & y \geq 0, x < 0 \\ \arctan \frac{y}{x} - \pi & y < 0, x < 0 \\ +\frac{\pi}{2} & y > 0, x = 0 \\ -\frac{\pi}{2} & y < 0, x = 0 \\ \text{undefined} & y = 0, x = 0 \end{cases}$$

Example: Pendulum

Equation of motion
(with rotational inertia I)

$$I\ddot{\theta} = -mgl \sin(\theta) + \tau$$

How was this derived?



Motor produces torque
(angular force)

Angle expresses pendulum
range of motion

Pendulum of length l with
point mass m

Example: Pendulum

Equation of motion
(with rotational inertia I)

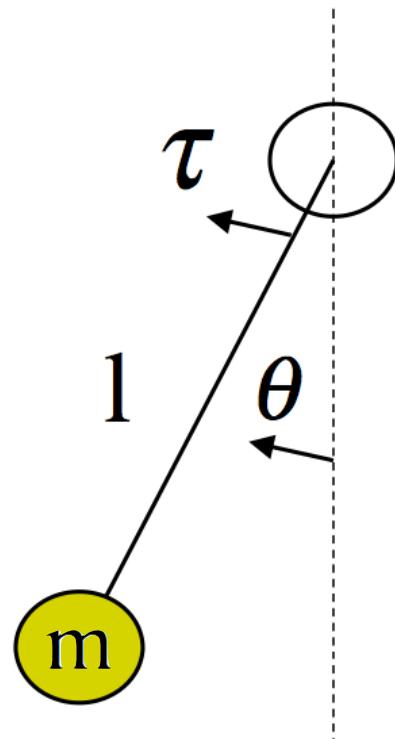
$$I\ddot{\theta} = -mgl \sin(\theta) + \tau$$

Lagrangian is kinetic minus
potential energy

$$L = T - U$$

used to generate equation
of motion

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_i} - \frac{\partial L}{\partial \theta_i} = \tau_i$$



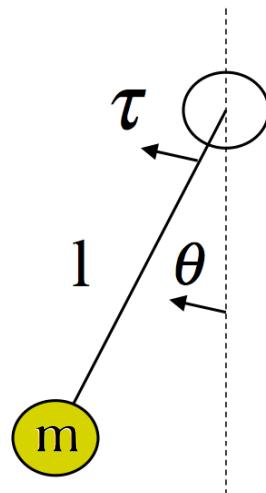
Motor produces torque
(angular force)

Angle expresses pendulum
range of motion

Pendulum of length l with
point mass m

Equation of motion
(with rotational inertia I)

$$I\ddot{\theta} = -mgl \sin(\theta) + \tau$$



Lagrangian Dynamics for Pendulum

- Kinetic Energy

$$(1/2)mv^2$$

$$T = \frac{1}{2} I \dot{\theta}^2$$

- Potential Energy

$$mgh$$

$$U = mgl(1 - \cos \theta)$$

- Lagrangian

$$L = T - U$$

$$\frac{1}{2} I \dot{\theta}^2 - mgl(1 - \cos \theta)$$

- Equation of motion

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_i} - \frac{\partial L}{\partial \theta_i} = \tau_i$$

$$I\ddot{\theta} + mgl \sin(\theta) = \tau$$

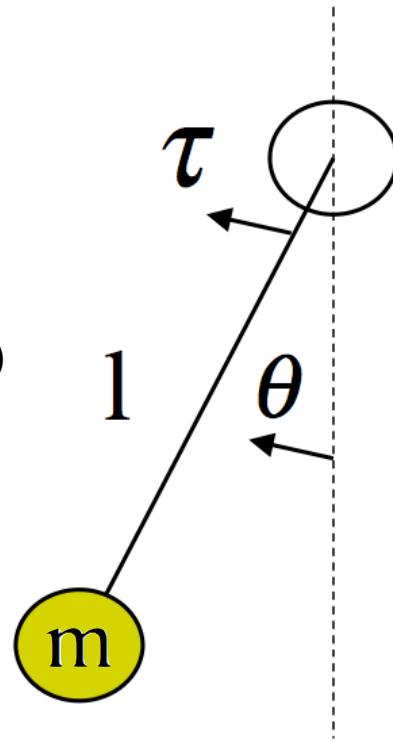
Example: Pendulum

Equation of motion
(with rotational inertia I)

$$I\ddot{\theta} = -mgl \sin(\theta) + \tau$$

with Parallel Axis Theorem ($I=ml^2$)

$$\ddot{\theta} = -\frac{g}{l} \sin(\theta) + \frac{\tau}{ml^2}$$



Motor produces torque
(angular force)

Angle expresses pendulum
range of motion

Pendulum of length l with
point mass m

Example: Pendulum

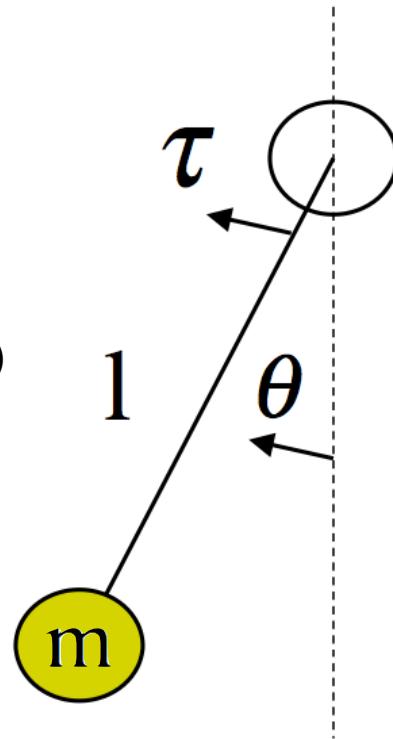
Equation of motion
(with rotational inertia I)

$$I\ddot{\theta} = -mgl \sin(\theta) + \tau$$

with Parallel Axis Theorem ($I=ml^2$)

$$\ddot{\theta} = -\frac{g}{l} \sin(\theta) + \frac{\tau}{ml^2}$$

gravity term
motor term



Motor produces torque
(angular force)

Angle expresses pendulum
range of motion

Pendulum of length l with
point mass m

Example: Pendulum

Equation of motion
(with rotational inertia I)

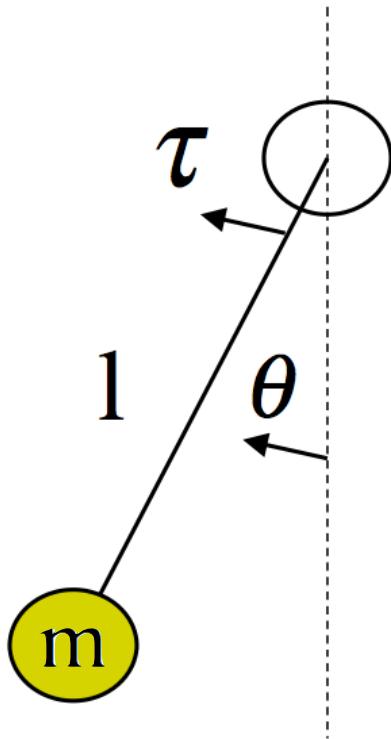
$$I\ddot{\theta} = -mgl \sin(\theta) + \tau$$

with Parallel Axis Theorem ($I=ml^2$)

$$\ddot{\theta} = -\frac{g}{l} \sin(\theta) + \frac{\tau}{ml^2}$$

Numerical integration over time

$$\begin{aligned}\theta_{t+dt} &= \theta_t + \dot{\theta}_t dt \\ \dot{\theta}_{t+dt} &= \dot{\theta}_t + \ddot{\theta}_t dt\end{aligned}$$



Motor produces torque
(angular force)

Angle expresses pendulum
range of motion

Pendulum of length l with
point mass m

Example: Pendulum

Equation of motion
(with rotational inertia I)

$$I\ddot{\theta} = -mgl \sin(\theta) + \tau$$

with Parallel Axis Theorem ($I=ml^2$)

What is this?

Numerical integration over time

$$\begin{aligned}\theta_{t+dt} &= \theta_t + \dot{\theta}_t dt \\ \dot{\theta}_{t+dt} &= \dot{\theta}_t + \ddot{\theta}_t dt\end{aligned}$$

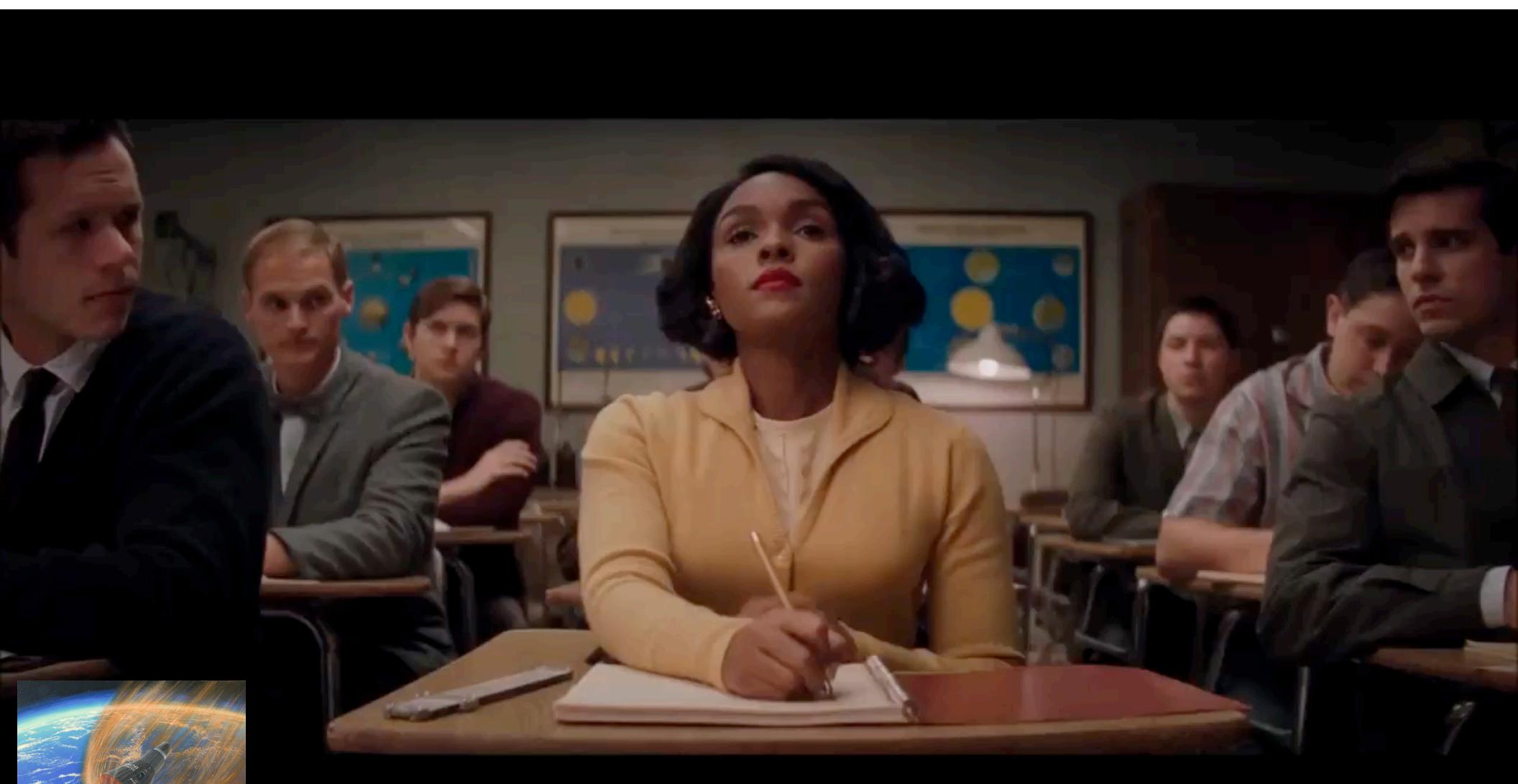


produces torque
(angular force)

reserves pendulum
of motion

Hint

Pendulum of length l with
point mass m



Hidden Figures (2017)

Example: Pendulum

Equation of motion
(with rotational inertia I)

$$I\ddot{\theta} = -mgl \sin(\theta) + \tau$$

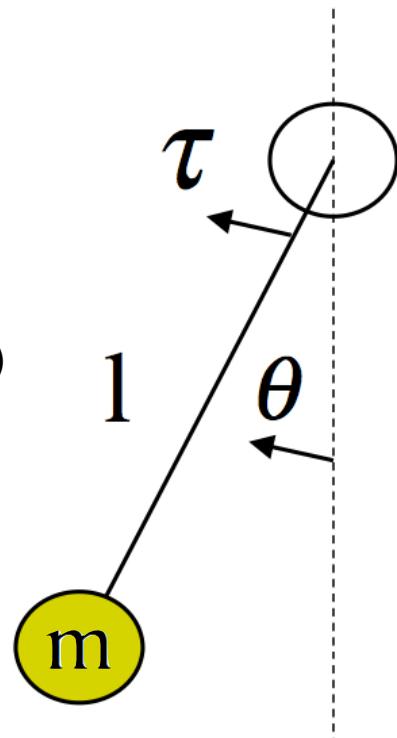
with Parallel Axis Theorem ($I=ml^2$)

$$\ddot{\theta} = -\frac{g}{l} \sin(\theta) + \frac{\tau}{ml^2}$$

Numerical integration over time

$$\theta_{t+dt} = \theta_t + \dot{\theta}_t dt$$

$$\dot{\theta}_{t+dt} = \dot{\theta}_t + \ddot{\theta}_t dt$$



Euler integration

Motor produces torque
(angular force)

Angle expresses pendulum
range of motion

Pendulum of length l with
point mass m

Let's see what happens

A screenshot of a web browser displaying a 3D simulation of a double pendulum. The simulation shows two red spherical masses connected by a rigid rod, swinging between two vertical grey cylindrical supports. The browser window includes a header with navigation icons and a file path, and a sidebar with system parameters and control keys.

System
t = 162.00 dt = 0.05
integrator = velocity verlet
x = -1.26
x_dot = -0.00
x_desired = -1.26

Servo: active
u = -37.32
kp = 1500.00
kd = 15.00
ki = 150.10

Pendulum
mass = 2.00
length = 2.00
gravity = 9.81

Keys
[0-4] - select integrator
a/d - apply user force
q/e - adjust desired angle
c - toggle servo
s - disable servo

Euler integration would kill this man



Walter Lewin - <https://commons.wikimedia.org/wiki/File:Physicsworks.ogg> UM EECS 358/558 autorob.github.io

Initial value problem

From Wikipedia, the free encyclopedia

In [mathematics](#), in the field of [differential equations](#), an **initial value problem** (also called the **Cauchy problem** by some authors) is an [ordinary differential equation](#) together with a specified value, called the **initial condition**, of the unknown function at a given point in the domain of the solution. In [physics](#) or other sciences, modeling a system frequently amounts to solving an initial value problem; in this context, the differential equation is an evolution equation specifying how, given initial conditions, the system will [evolve with time](#).

Initial value problem

From Wikipedia, the free encyclopedia

Given initial condition (t_0, y_0) and

Initial value problem

From Wikipedia, the free encyclopedia

Given initial condition (t_0, y_0) and



Initial value problem

From Wikipedia, the free encyclopedia

Given initial condition (t_0, y_0) and
differential equation $y' = \frac{dy}{dt}$ in the form

$$y'(t) = f(t, y(t))$$

velocity system dynamics

Initial value problem

From Wikipedia, the free encyclopedia

Given initial condition (t_0, y_0) and

differential equation $y' = \frac{dy}{dt}$ in the form $y'(t) = f(t, y(t))$

how to estimate of future state $y(t + dt)$ at simulation iteration n?

Initial value problem

From Wikipedia, the free encyclopedia

Given initial condition (t_0, y_0) and

differential equation $y' = \frac{dy}{dt}$ in the form $y'(t) = f(t, y(t))$

how to estimate of future state $y(t + dt)$ at simulation iteration n?

numerical integration over timestep iterations

$$y(t_0 + h) - y(t_0) = \int_{t_0}^{t_0+h} f(t, y(t)) dt.$$

next state current state integration over timestep

Given initial condition (t_0, y_0) and

differential equation $y' = \frac{dy}{dt}$ in the form $y'(t) = f(t, y(t))$

how to estimate of future state $y(t + dt)$ at simulation iteration n?

Euler's method $\int_{t_0}^{t_0+h} f(t, y(t)) dt \approx h f(t_0, y(t_0))$.

next state current state advance time

$y_{n+1} = y_n + h f(t_n, y_n)$

timestep (dt)

Example Euler integration of
2D point over 2 time steps

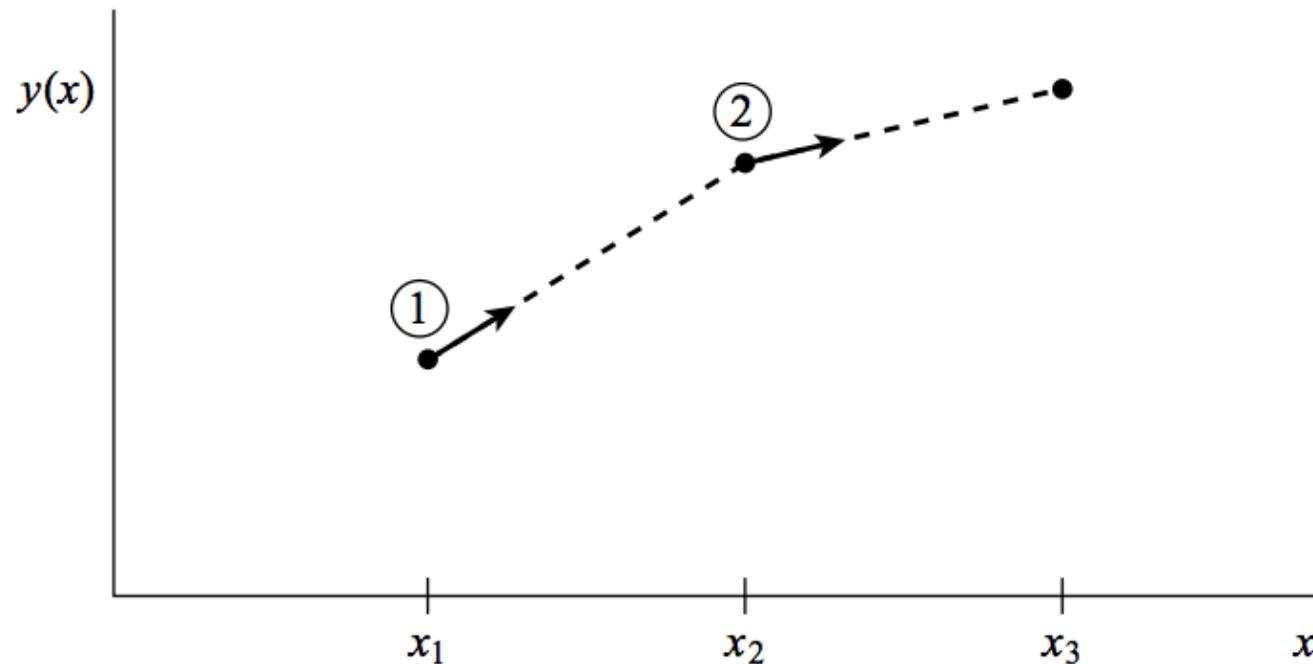


Figure 16.1.1. Euler's method. In this simplest (and least accurate) method for integrating an ODE, the derivative at the starting point of each interval is extrapolated to find the next function value. The method has first-order accuracy.

Second-order state

- State in Newtonian physics has both position (θ) and velocity ($\dot{\theta}$)

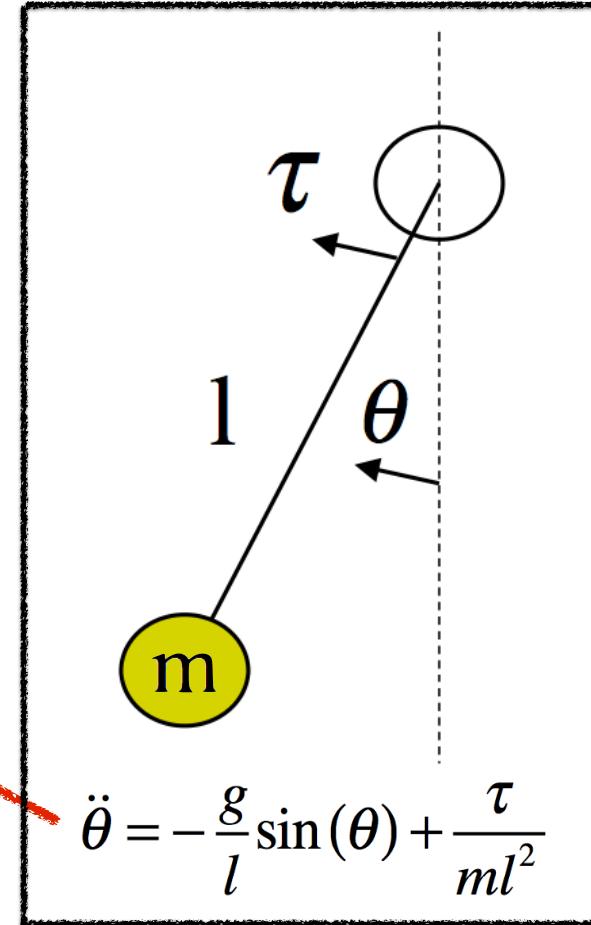
Euler's method

$$y_{n+1} = y_n + \dot{y}_n dt$$

$$\dot{y}_{n+1} = \dot{y}_n + \ddot{y}_n dt$$

acceleration

$$t_{n+1} = t_n + dt$$



assumes second order system $f(y, y', y'')$ treats velocity $y' = f(y)$ and acceleration $y'' = f'(y')$ individually

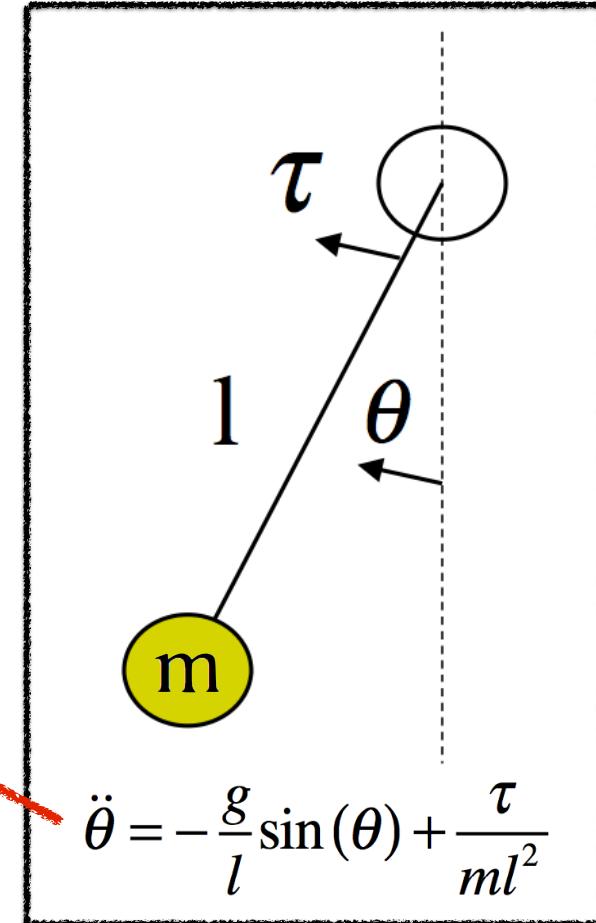
Euler's method

$$y_{n+1} = y_n + \dot{y}_n dt$$

$$\dot{y}_{n+1} = \dot{y}_n + \ddot{y}_n dt$$

acceleration

$$t_{n+1} = t_n + dt$$

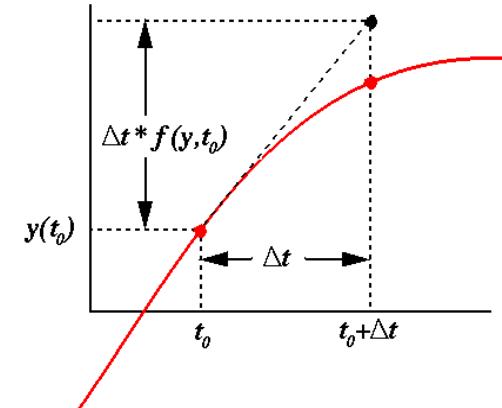


Why is the pendulum going unstable?

Given initial condition (t_0, y_0) and differential equation $y' = \frac{dy}{dt}$ in the form

$$y'(t) = f(t, y(t))$$

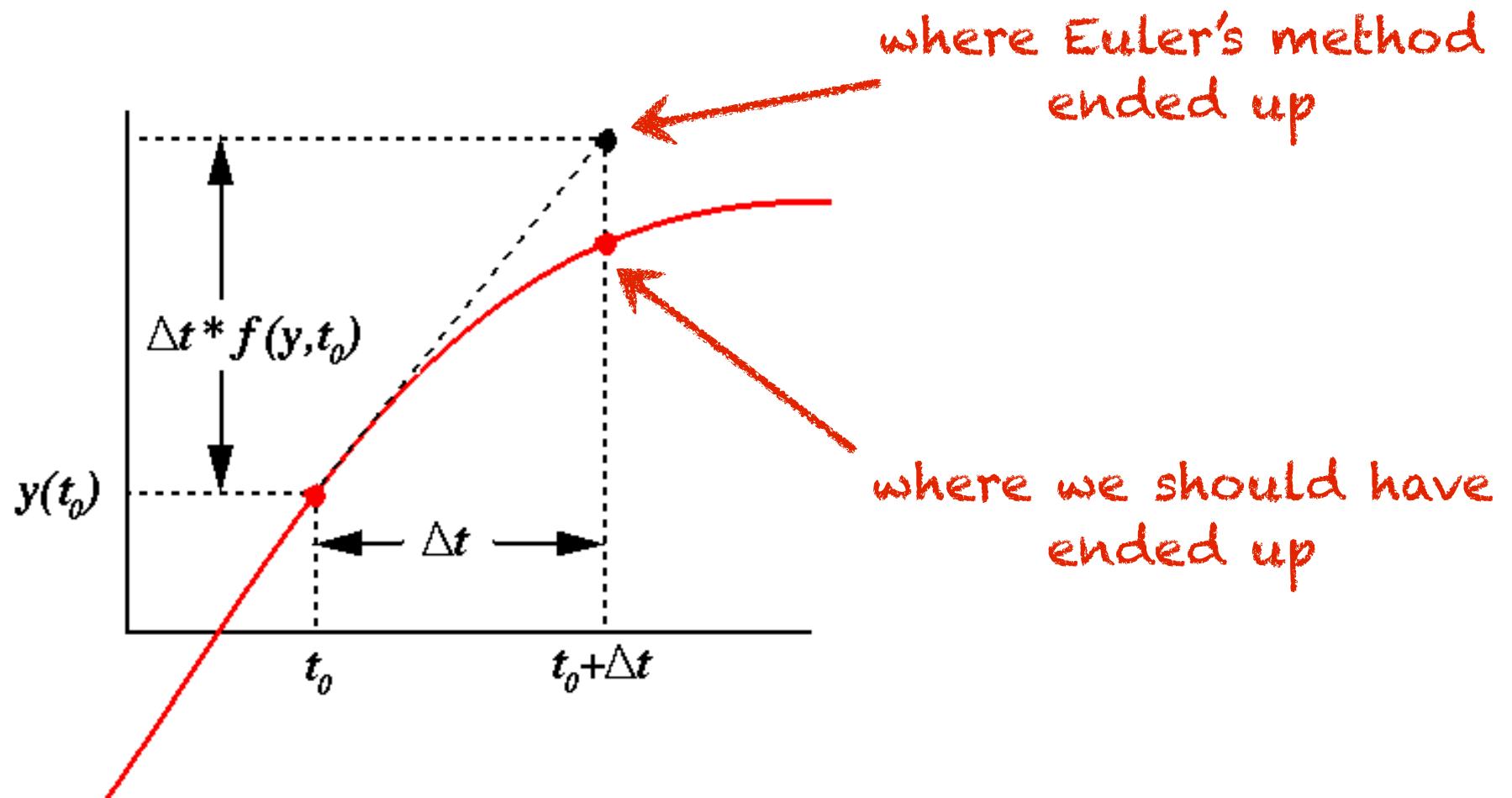
how to estimate of future state $y(t_n)$ at time n?



Euler's method $\int_{t_0}^{t_0+h} f(t, y(t)) dt \approx h f(t_0, y(t_0))$ **integral approximation**

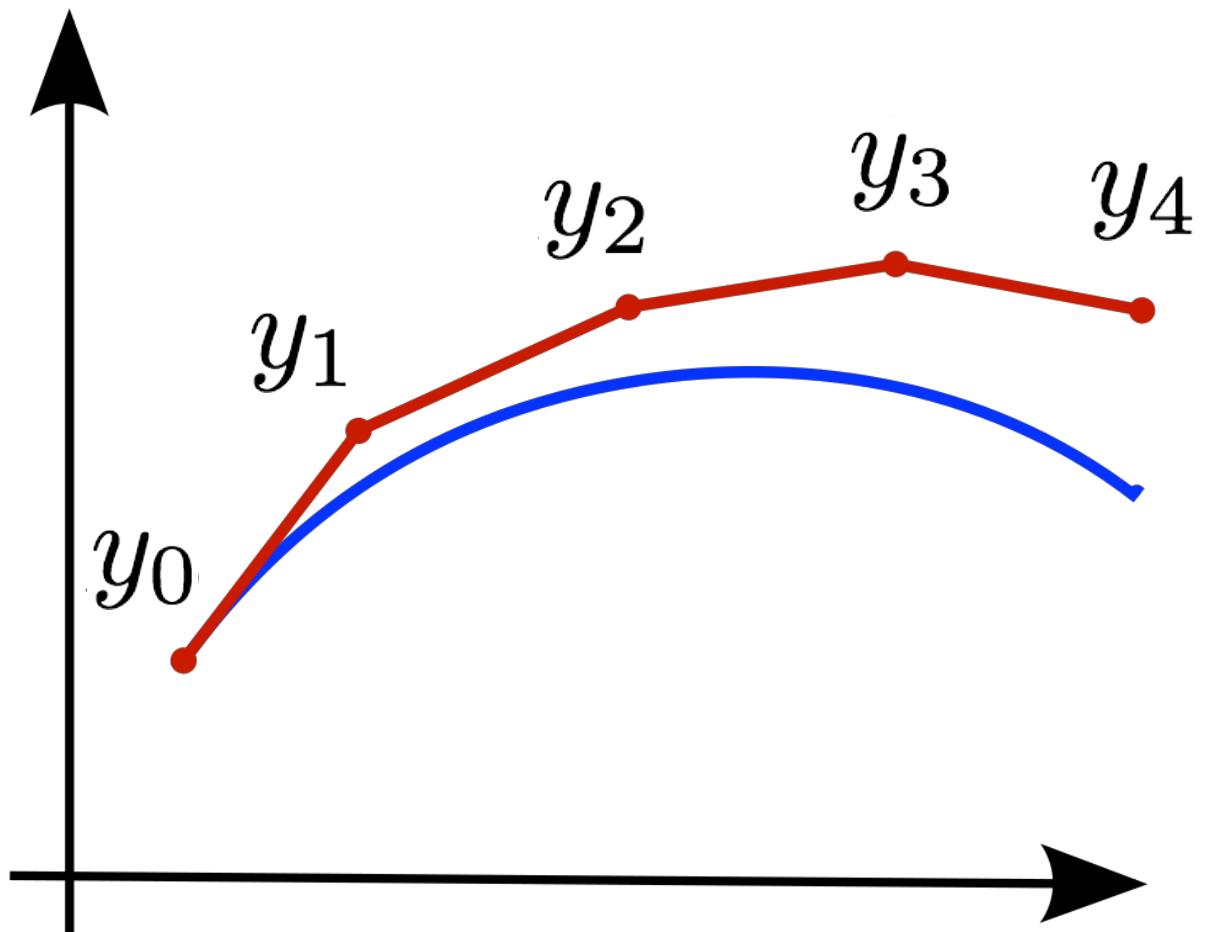
$$y_{n+1} = y_n + h f(t_n, y_n)$$

$$t_{n+1} = t_n + h$$

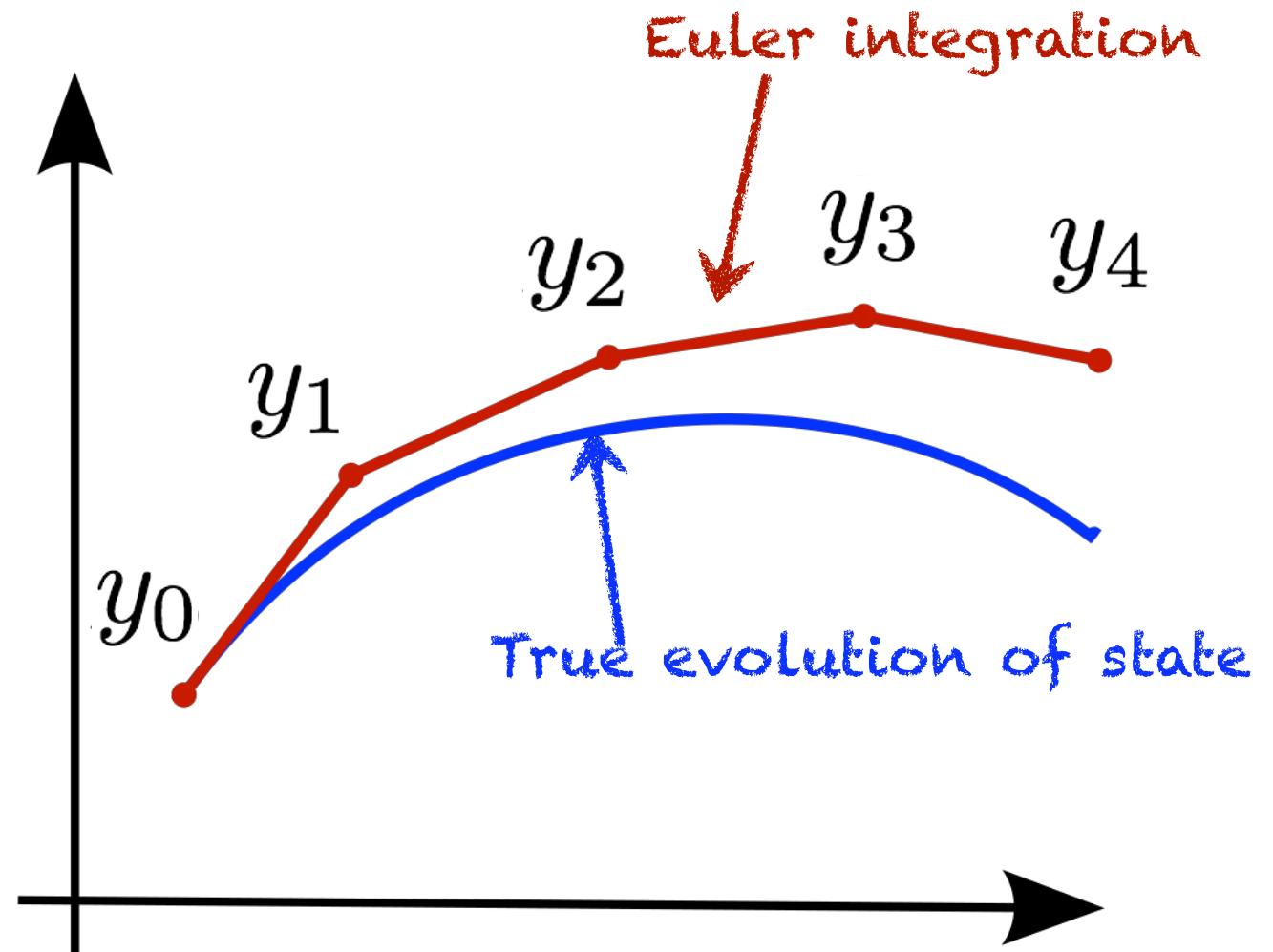


integral approximation

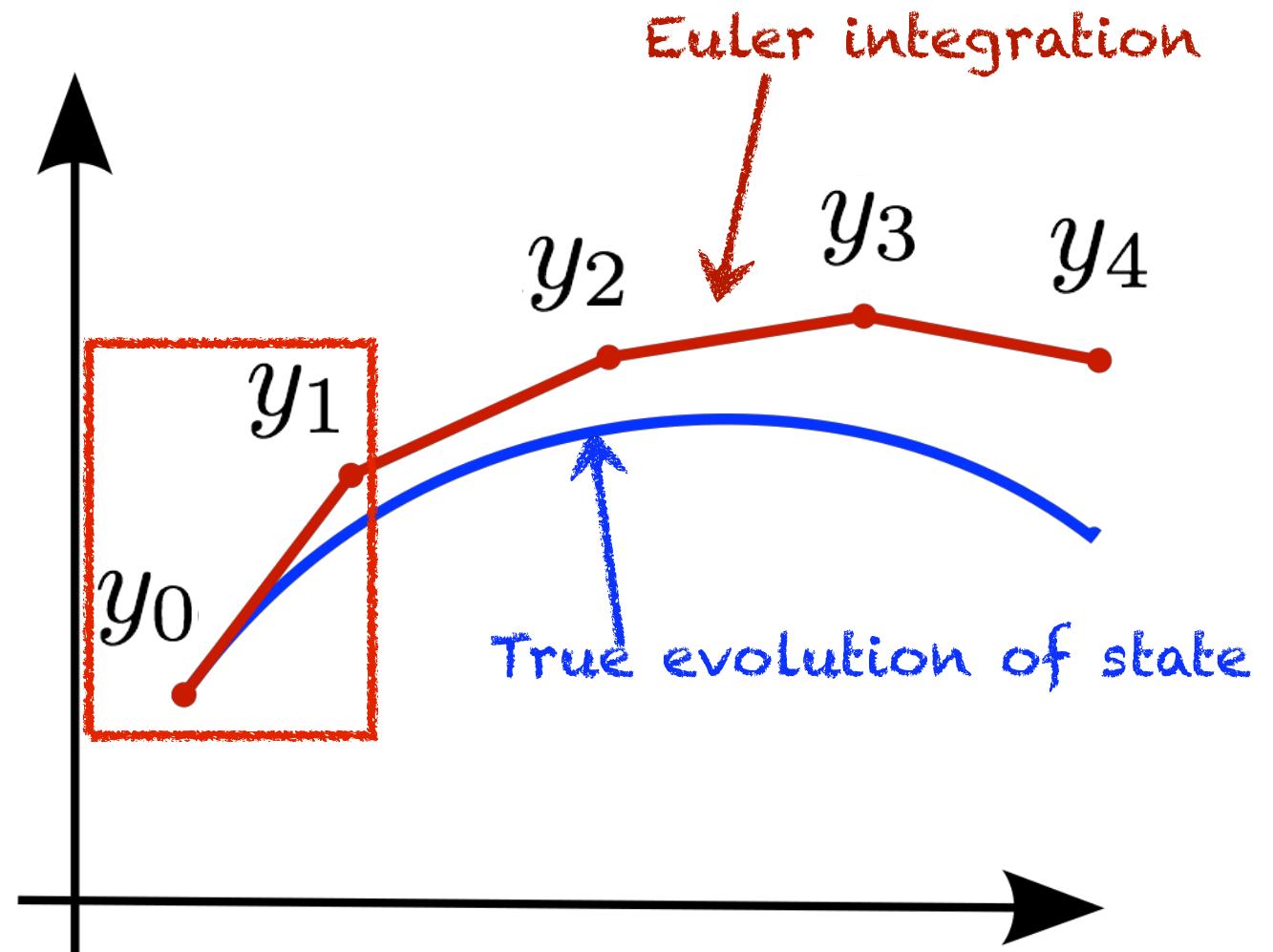
Example Euler
integration of 2D point
over 4 time steps



Example Euler
integration of 2D point
over 4 time steps

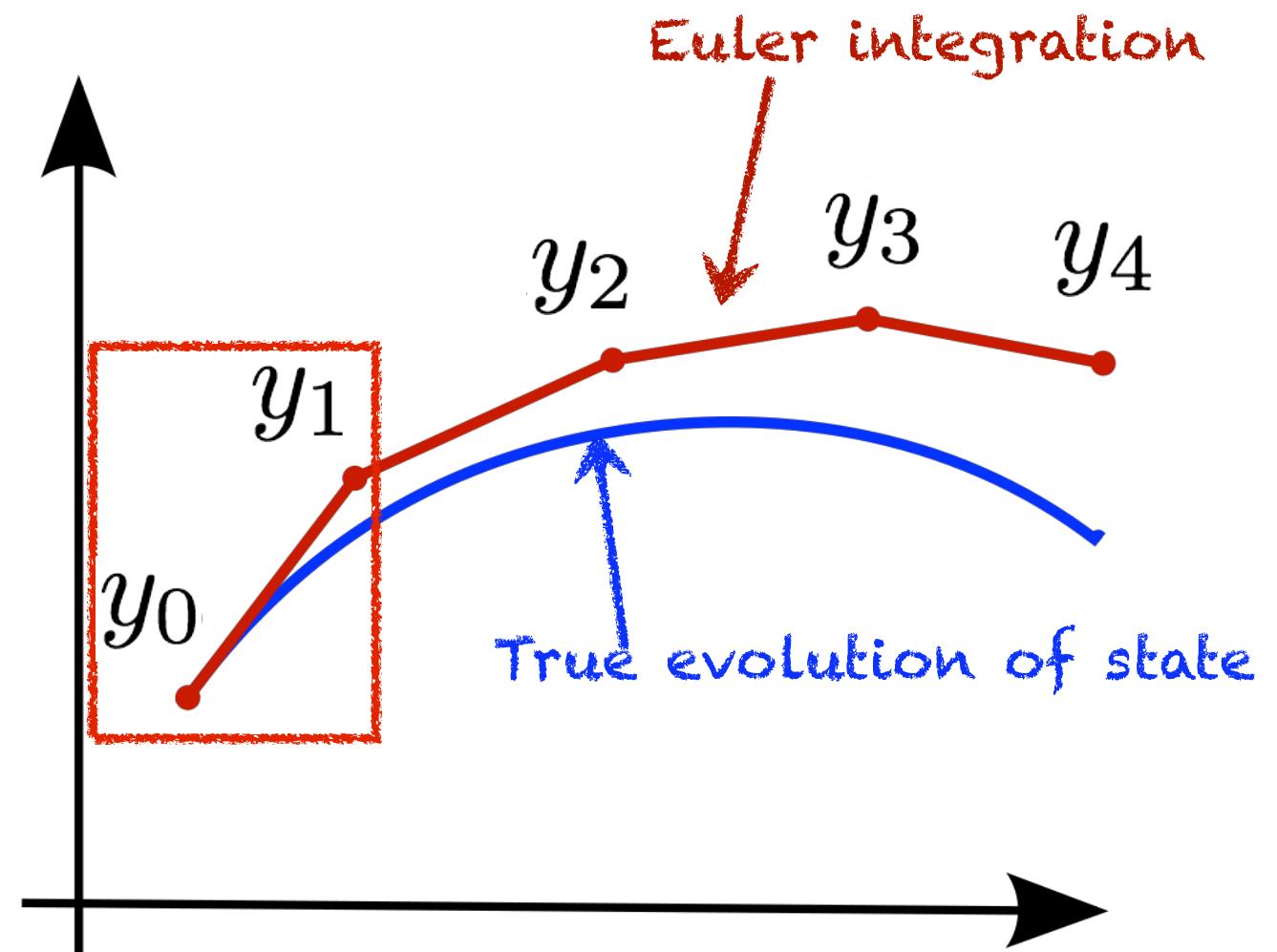


Example Euler
integration of 2D point
over 4 time steps

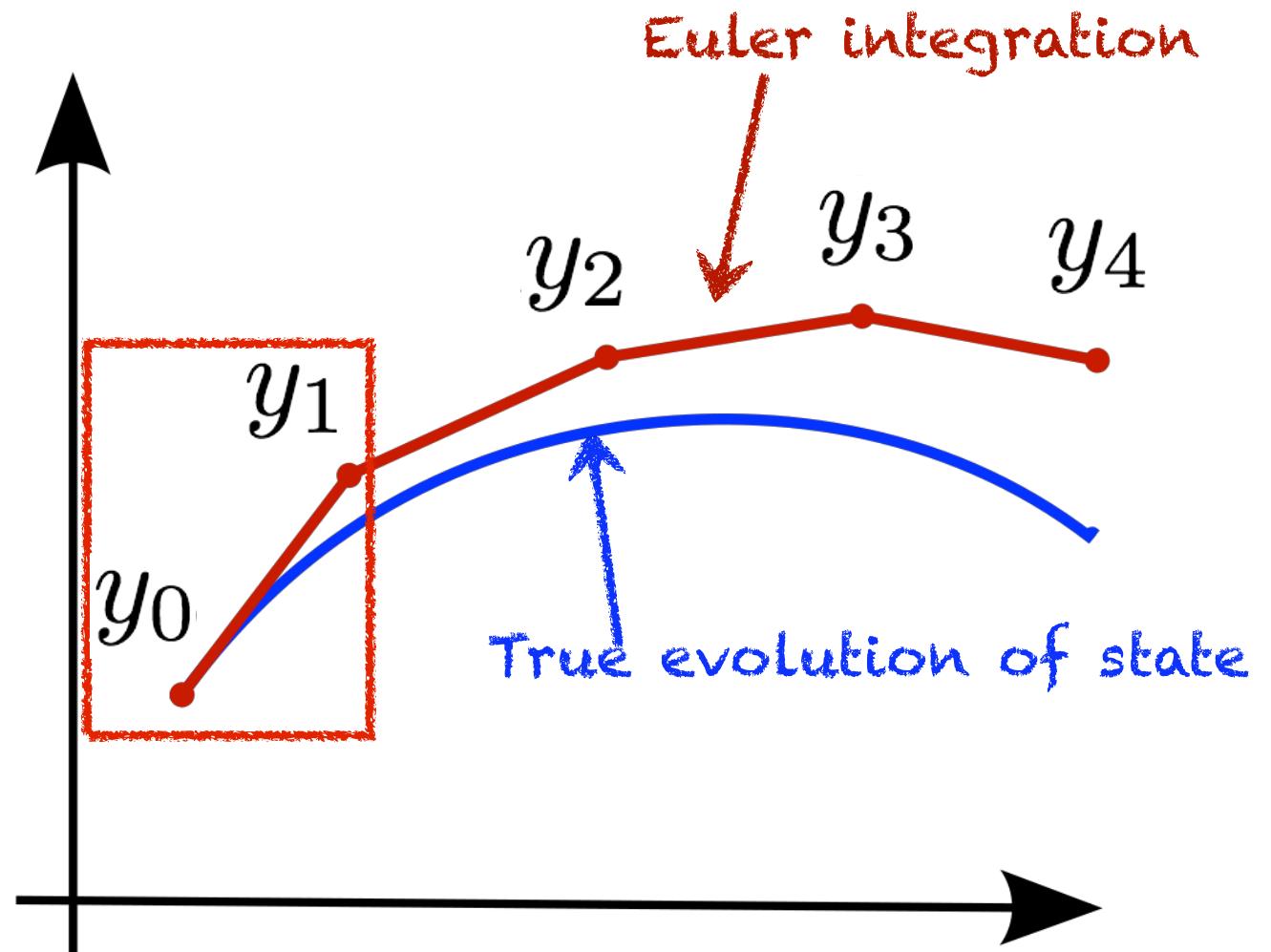
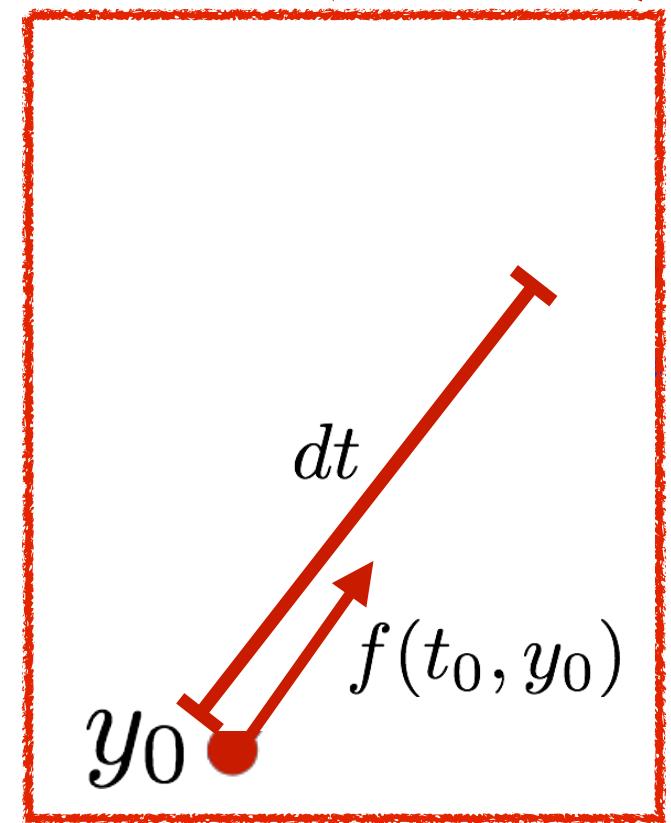


Initial tangent

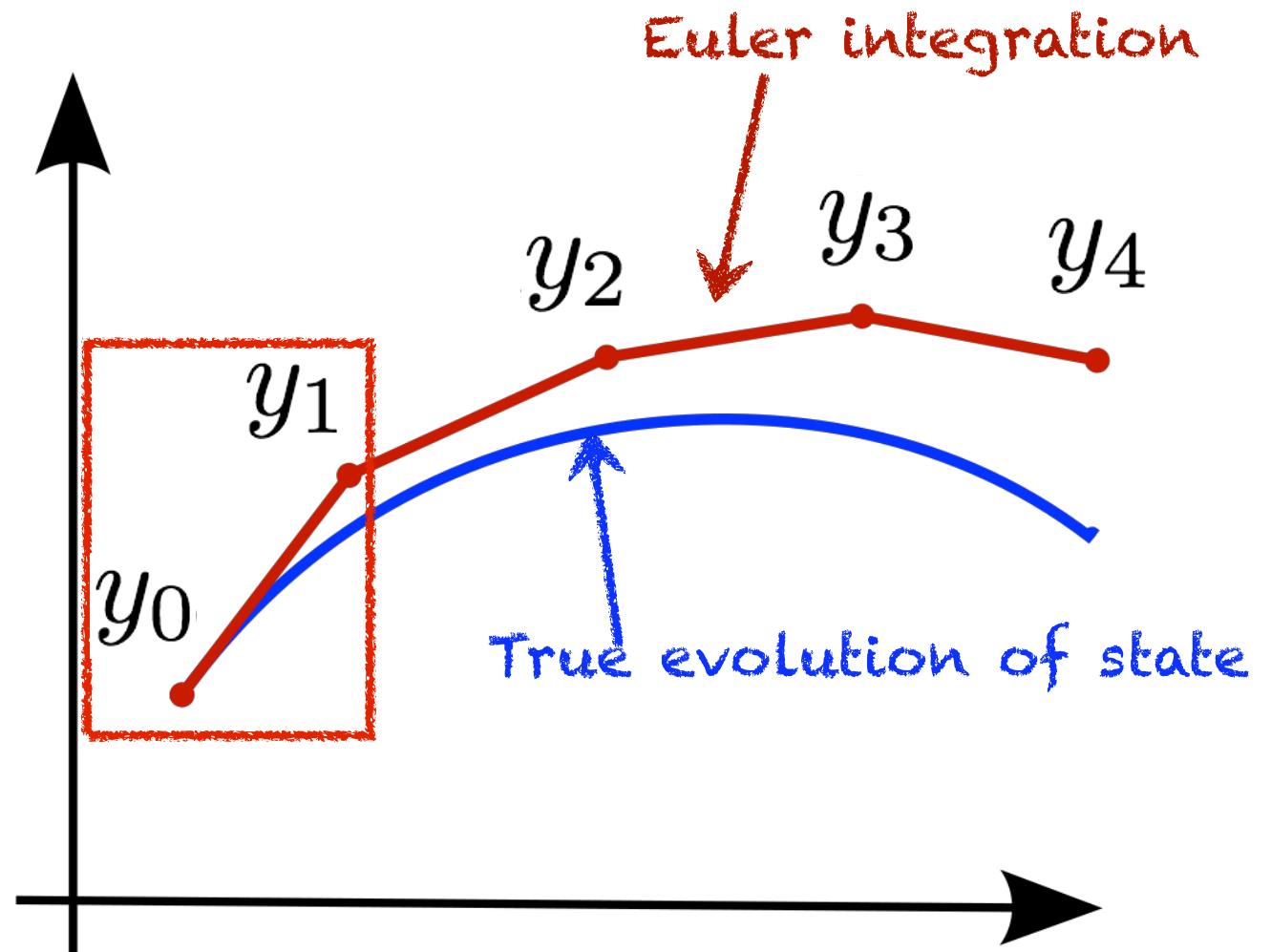
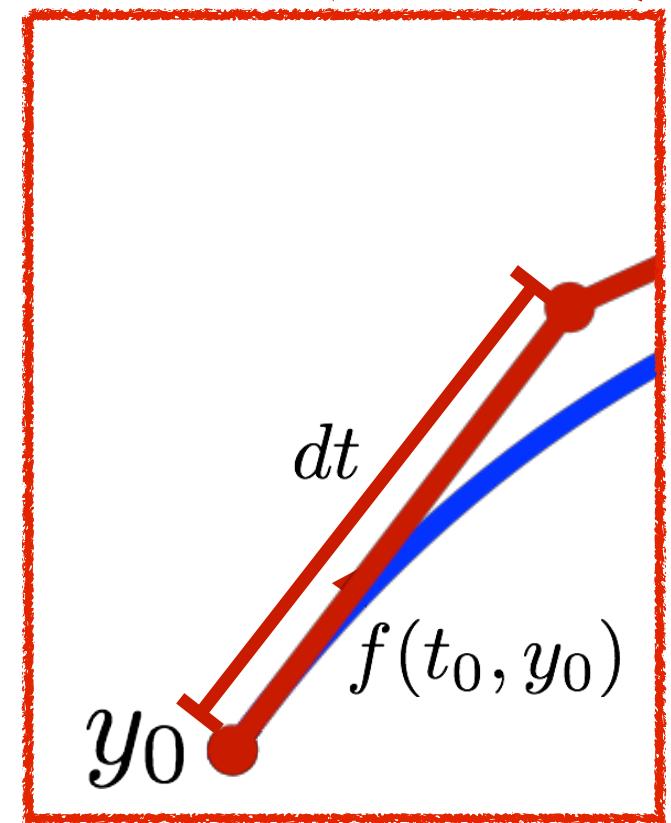
$$y_0 \quad f(t_0, y_0)$$



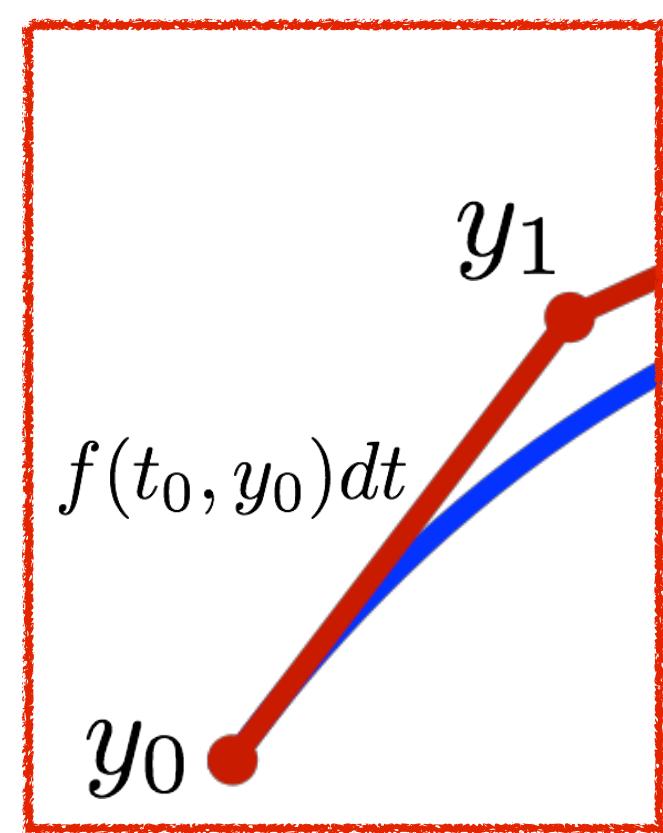
Extend by timestep



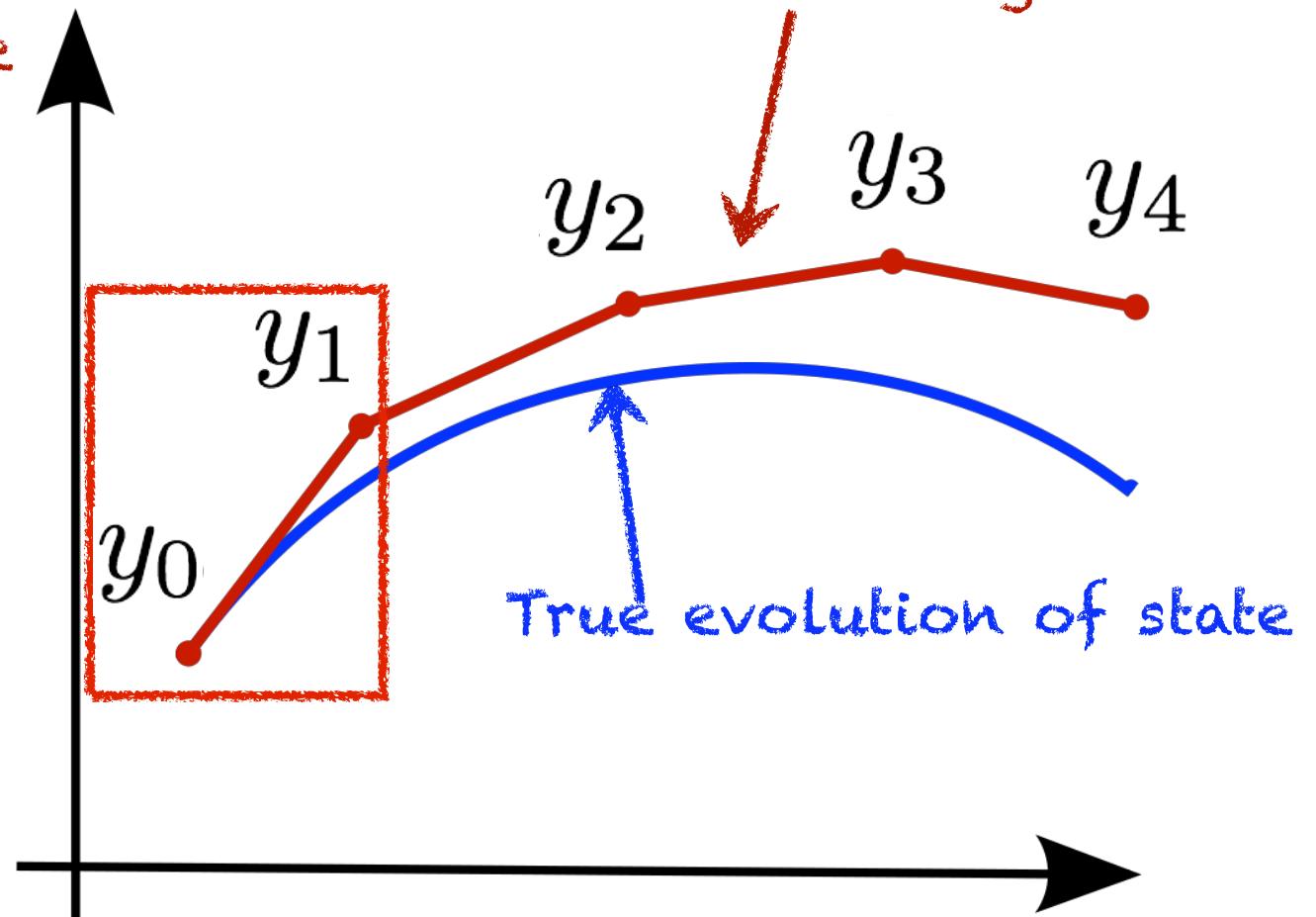
Extend by timestep

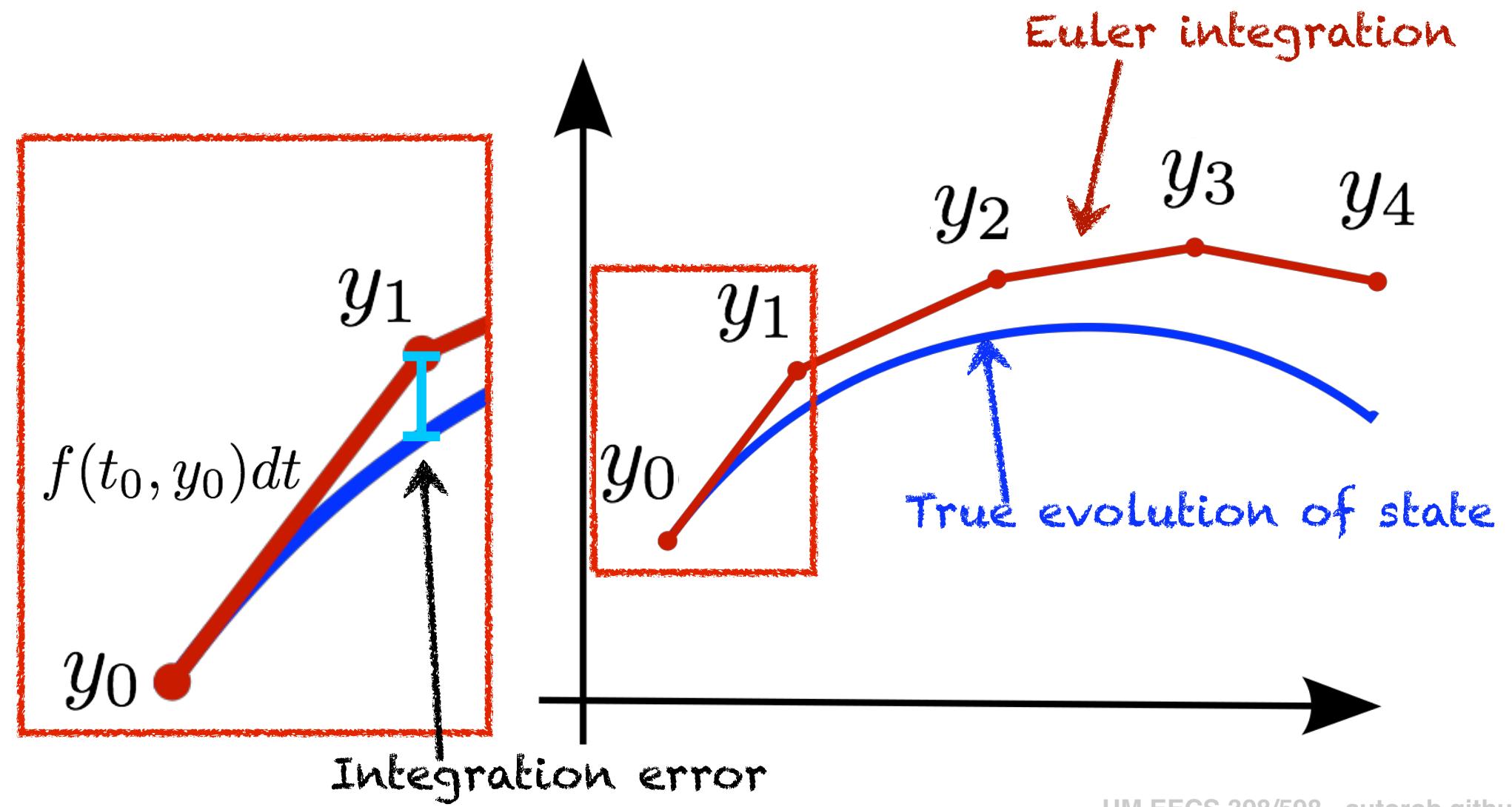


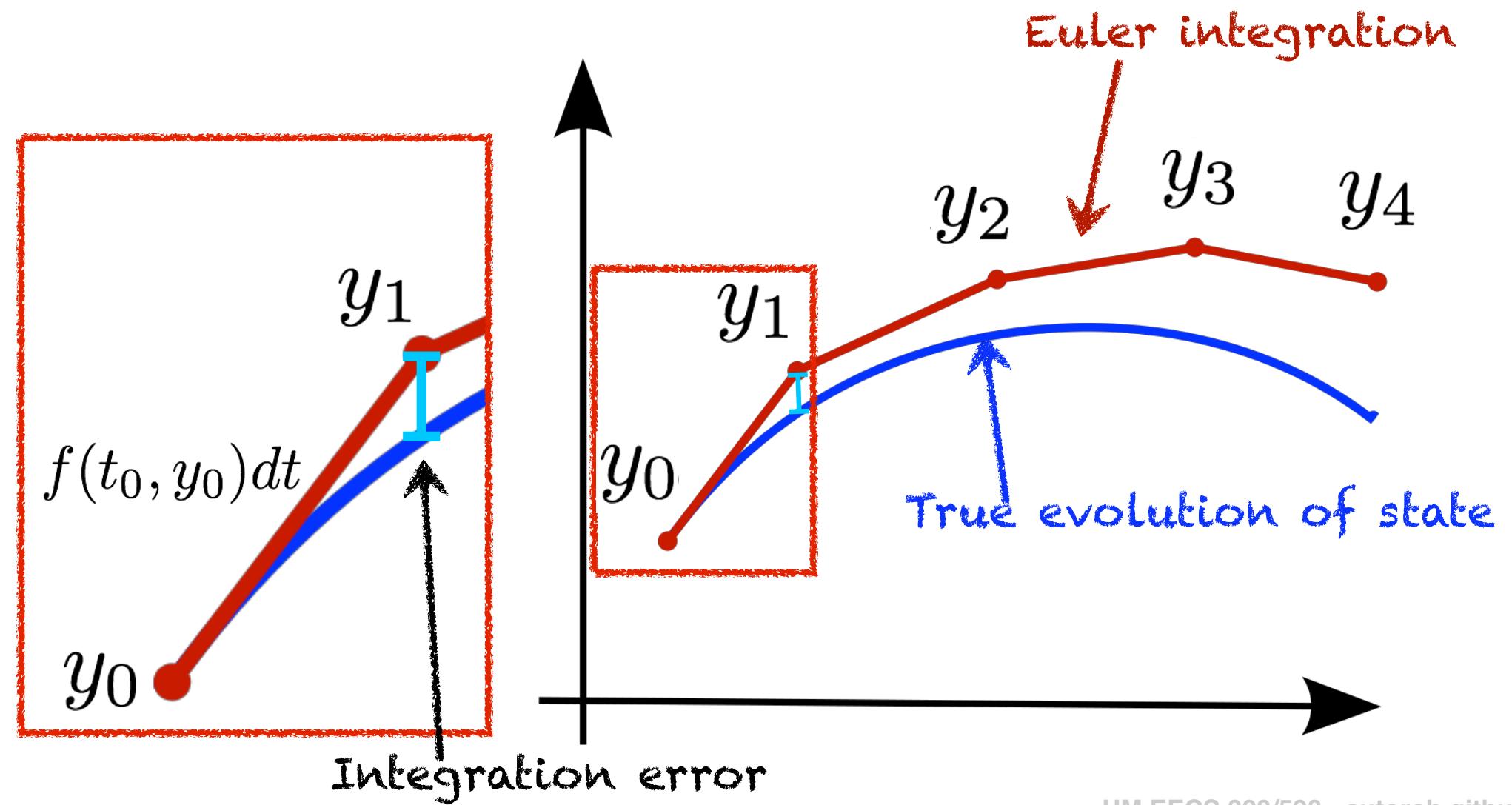
Add to current state

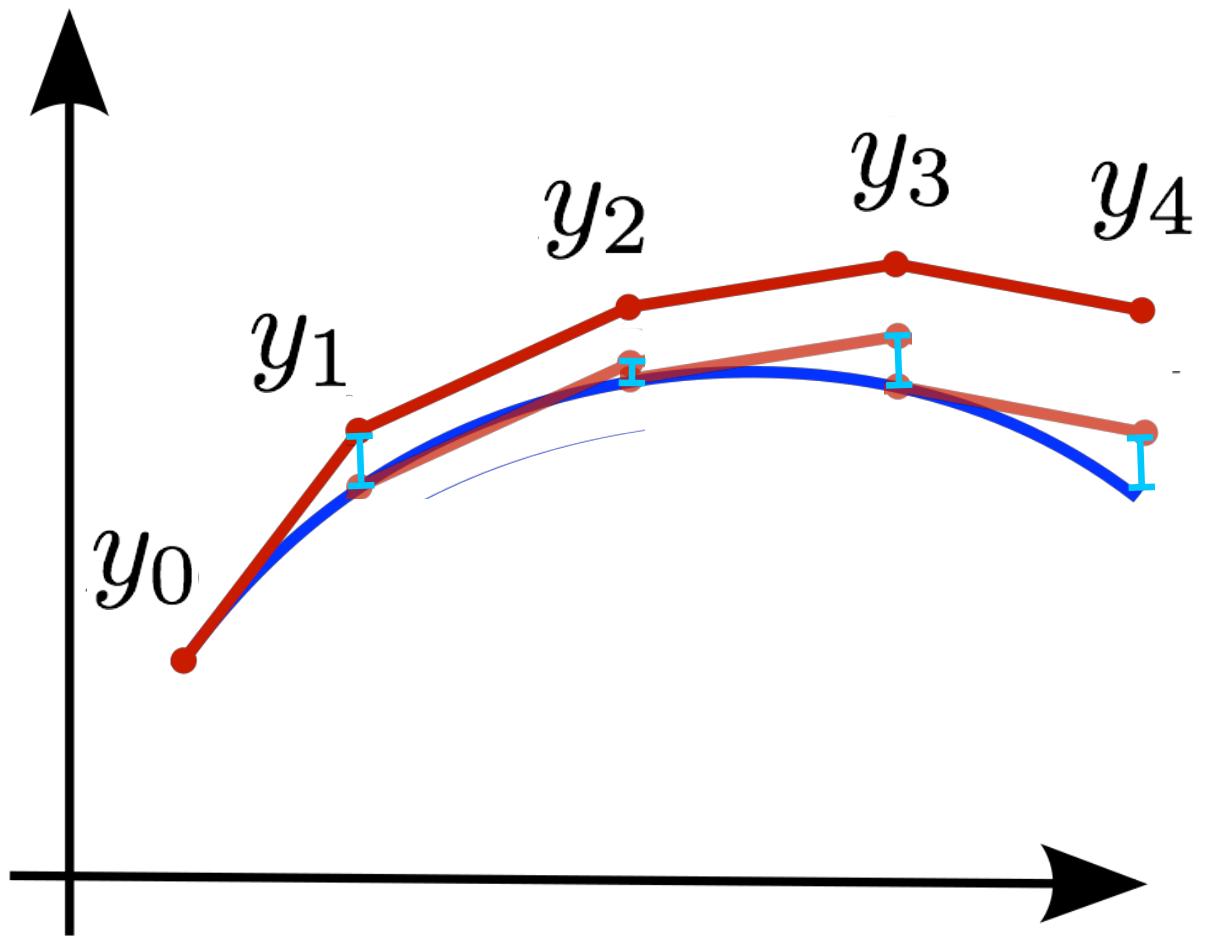


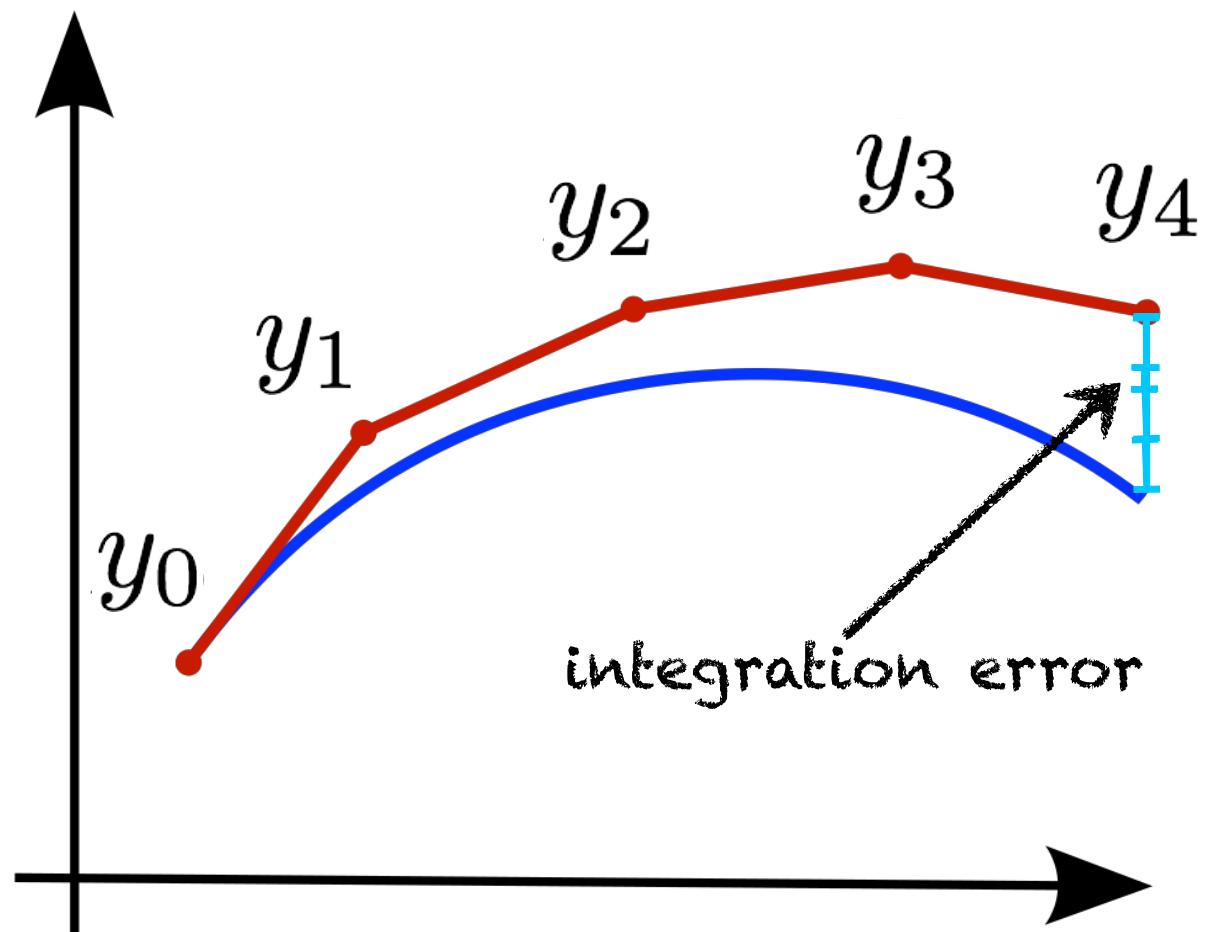
Euler integration



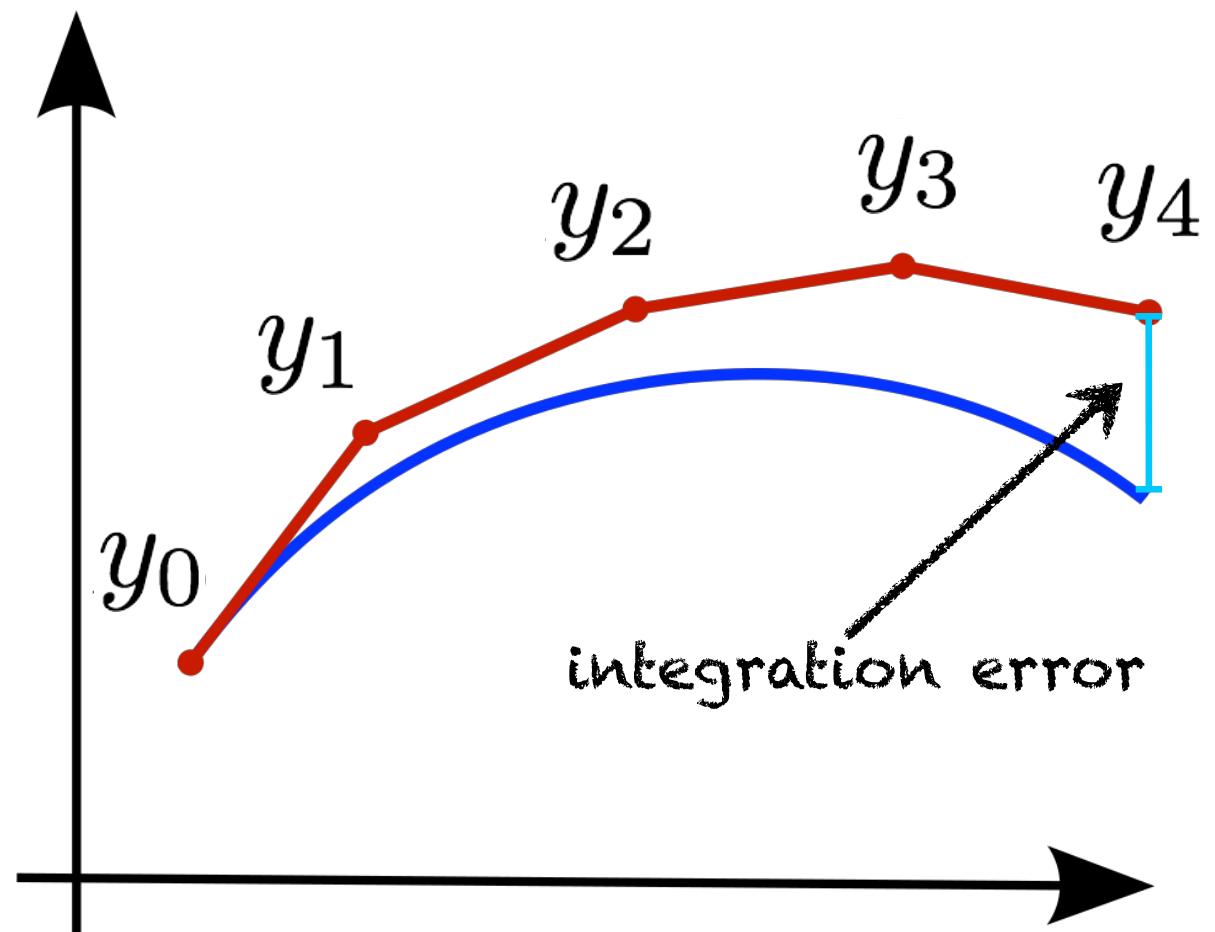








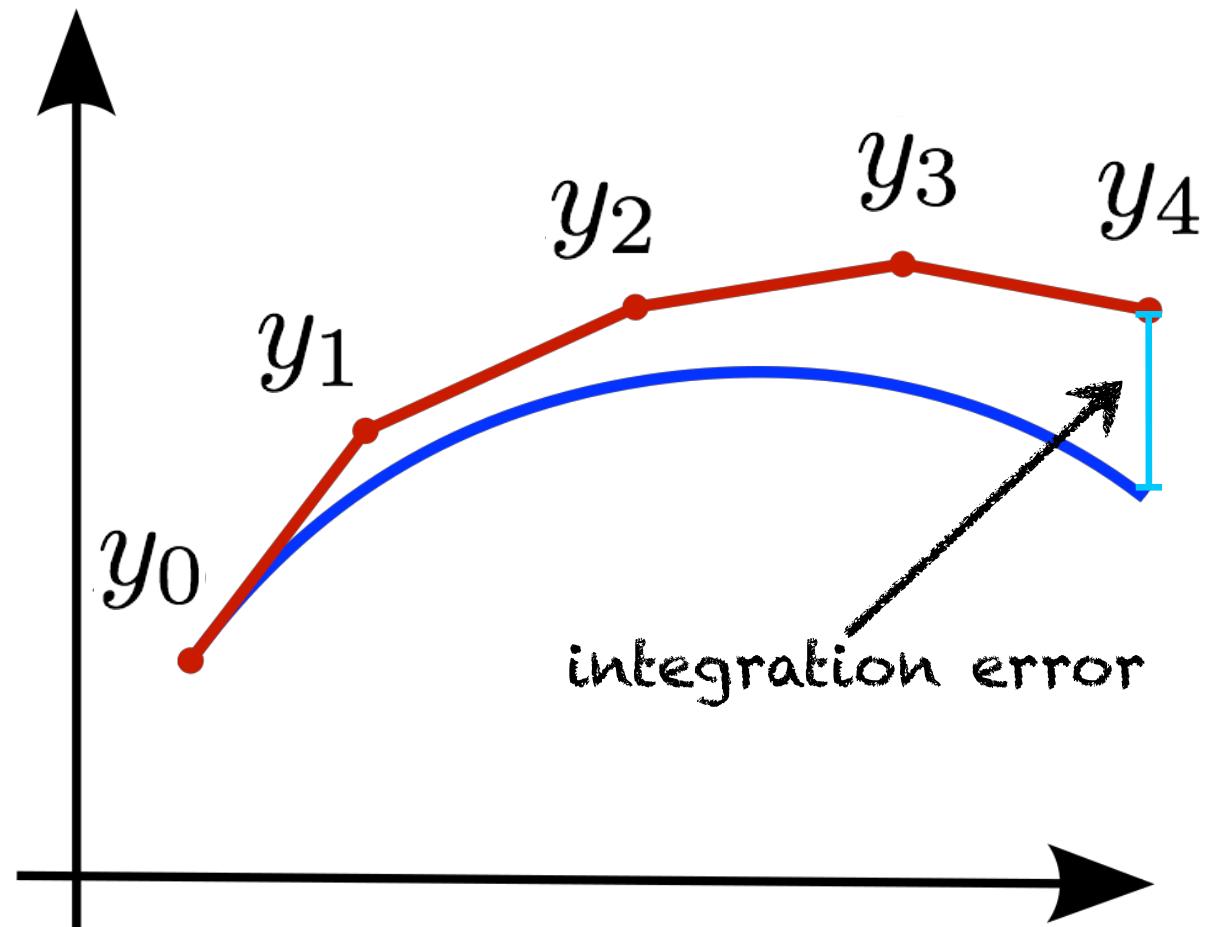
Can we improve
this integration
over time?



Can we improve
this integration
over time?

Option 1:

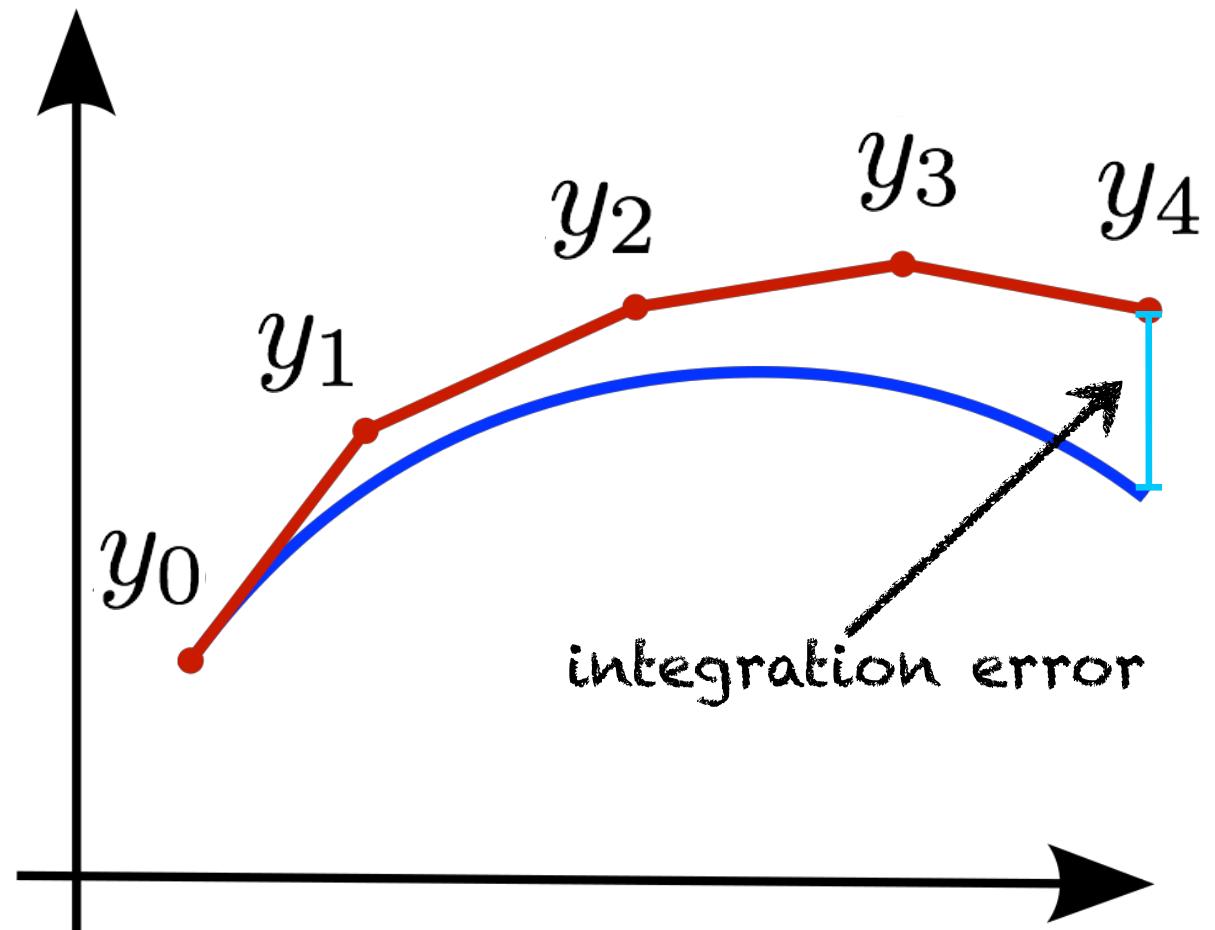
Option 2:



Can we improve
this integration
over time?

Option 1:
Reduce timestep

Option 2:
Use a better
integrator



Verlet Integration

For a [differential equation](#) of second order of the type $\ddot{\vec{x}}(t) = A(\vec{x}(t))$ with initial conditions $\vec{x}(t_0) = \vec{x}_0$ and $\dot{\vec{x}}(t_0) = \vec{v}_0$, an approximate numerical solution $\vec{x}_n \approx \vec{x}(t_n)$ at the times $t_n = t_0 + n \Delta t$ with step size $\Delta t > 0$ can be obtained by the following method:

Iterate position $\vec{x}_{n+1} = 2\vec{x}_n - \vec{x}_{n-1} + A(\vec{x}_n) \Delta t^2$

Verlet Integration

$$A(\vec{x}_n) = \frac{\Delta^2 \vec{x}_n}{\Delta t^2} = \frac{v_n - v_{n-1}}{\Delta t}$$

Verlet Integration

$$\begin{aligned} A(\vec{x}_n) &= \frac{\Delta^2 \vec{x}_n}{\Delta t^2} = \frac{v_n - v_{n-1}}{\Delta t} = \frac{\frac{\vec{x}_{n+1} - \vec{x}_n}{\Delta t} - \frac{\vec{x}_n - \vec{x}_{n-1}}{\Delta t}}{\Delta t} \\ &= \frac{\vec{x}_{n+1} - 2\vec{x}_n + \vec{x}_{n-1}}{\Delta t^2} \end{aligned}$$

Verlet Integration

$$A(\vec{x}_n) = \frac{\Delta^2 \vec{x}_n}{\Delta t^2} = \frac{v_n - v_{n-1}}{\Delta t} = \frac{\frac{\vec{x}_{n+1} - \vec{x}_n}{\Delta t} - \frac{\vec{x}_n - \vec{x}_{n-1}}{\Delta t}}{\Delta t}$$

$$a_n = \frac{\vec{x}_{n+1} - 2\vec{x}_n + \vec{x}_{n-1}}{\Delta t^2}$$

$$\vec{x}_{n+1} = 2\vec{x}_n - \vec{x}_{n-1} + a_n \Delta t^2$$

Verlet Integration

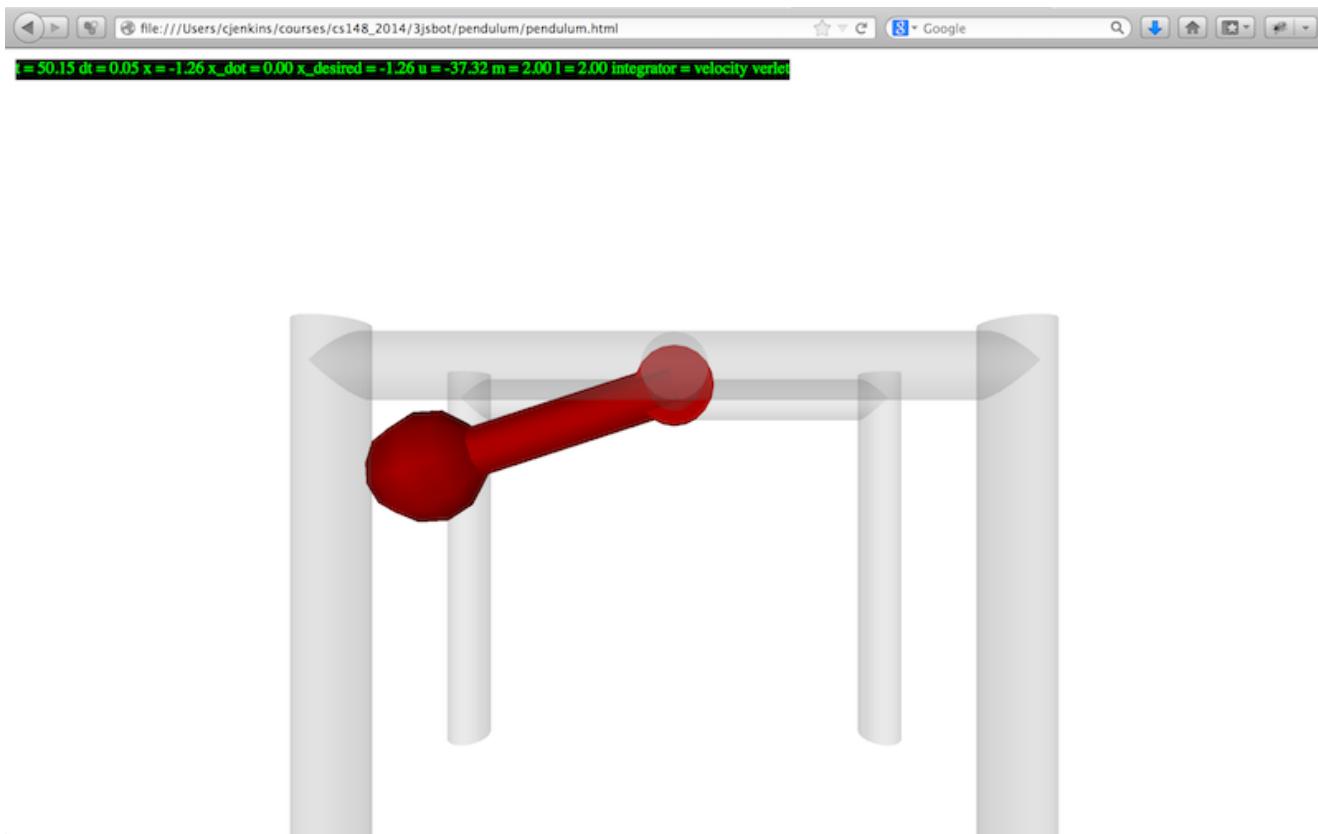
For a differential equation of second order of the type $\ddot{\vec{x}}(t) = A(\vec{x}(t))$ with initial conditions $\vec{x}(t_0) = \vec{x}_0$ and $\dot{\vec{x}}(t_0) = \vec{v}_0$, an approximate numerical solution $\vec{x}_n \approx \vec{x}(t_n)$ at the times $t_n = t_0 + n \Delta t$ with step size $\Delta t > 0$ can be obtained by the following method:

Initialize $\vec{x}_1 = \vec{x}_0 + \vec{v}_0 \Delta t + \frac{1}{2} A(\vec{x}_0) \Delta t^2$

Iterate position $\vec{x}_{n+1} = 2\vec{x}_n - \vec{x}_{n-1} + A(\vec{x}_n) \Delta t^2$

Do not forget to initialize

Let's see what happens



Verlet Integration

For a differential equation of second order of the type $\ddot{\vec{x}}(t) = A(\vec{x}(t))$ with initial conditions $\vec{x}(t_0) = \vec{x}_0$ and $\dot{\vec{x}}(t_0) = \vec{v}_0$, an approximate numerical solution $\vec{x}_n \approx \vec{x}(t_n)$ at the times $t_n = t_0 + n \Delta t$ with step size $\Delta t > 0$ can be obtained by the following method:

Initialize $\vec{x}_1 = \vec{x}_0 + \vec{v}_0 \Delta t + \frac{1}{2} A(\vec{x}_0) \Delta t^2$

Iterate position $\vec{x}_{n+1} = 2\vec{x}_n - \vec{x}_{n-1} + A(\vec{x}_n) \Delta t^2$

**Iterate velocity
(optional)** $\vec{v}(t) = \frac{\vec{x}(t + \Delta t) - \vec{x}(t - \Delta t)}{2\Delta t} + \mathcal{O}(\Delta t^2)$

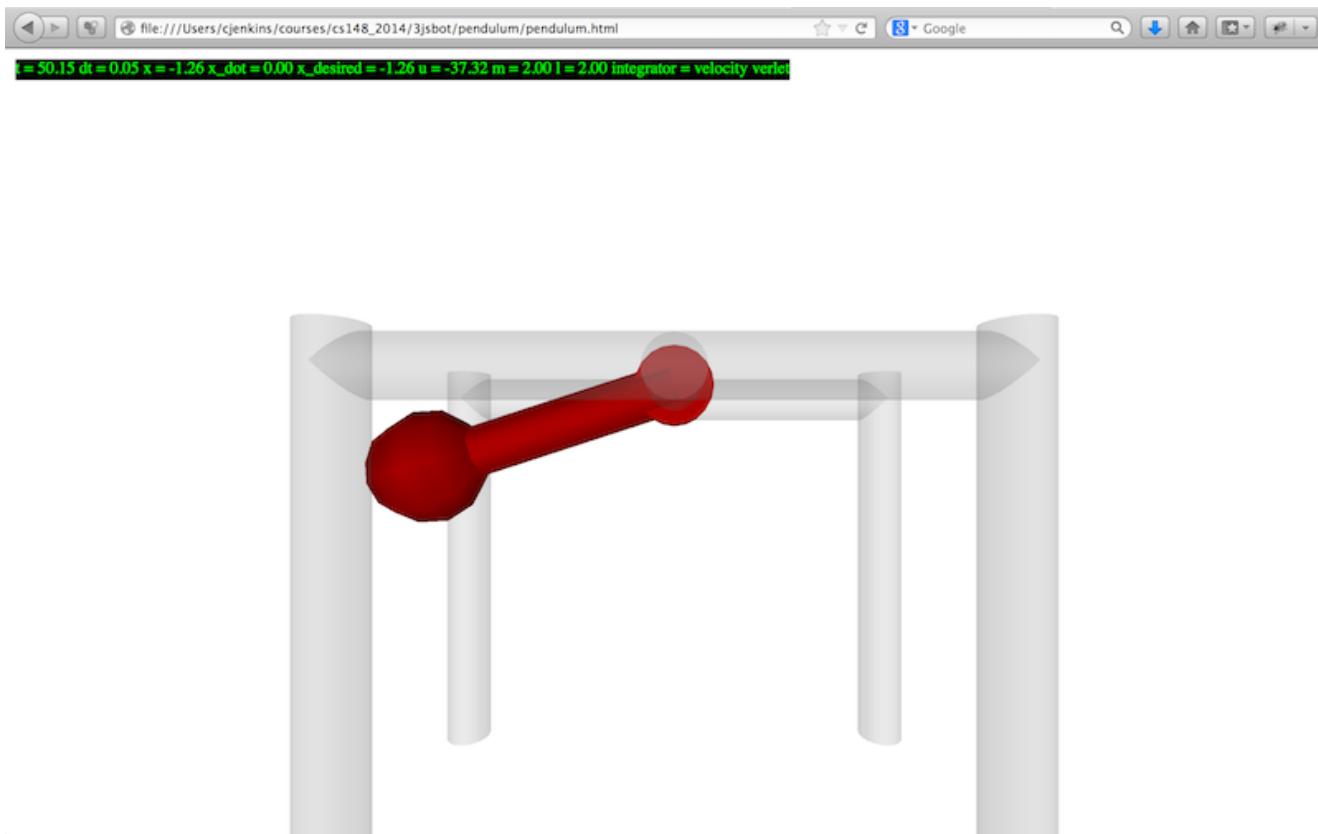
Velocity Verlet

$$\vec{x}(t + \Delta t) = \vec{x}(t) + \vec{v}(t) \Delta t + \frac{1}{2} \vec{a}(t) \Delta t^2$$

$$\vec{v}(t + \Delta t) = \vec{v}(t) + \frac{\vec{a}(t) + \vec{a}(t + \Delta t)}{2} \Delta t$$

assumes that acceleration $\vec{a}(t + \Delta t)$ only depends on position $\vec{x}(t + \Delta t)$

Let's see what happens



Runge-Kutta

$$y_{n+1} = y_n + \sum_{i=1}^s b_i k_i,$$

where

think of $f()$ as callable function
for dy/dt

$$k_1 = h f(t_n, y_n),$$

$$k_2 = h f(t_n + c_2 h, y_n + a_{21} k_1),$$

$$k_3 = h f(t_n + c_3 h, y_n + a_{31} k_1 + a_{32} k_2),$$

:

$$k_s = h f(t_n + c_s h, y_n + a_{s1} k_1 + a_{s2} k_2 + \cdots + a_{s,s-1} k_{s-1})$$

Runge-Kutta

Butcher tableau

$$y_{n+1} = y_n + \sum_{i=1}^s b_i k_i,$$

where

$$k_1 = h f(t_n, y_n),$$

$$k_2 = h f(t_n + c_2 h, y_n + a_{21} k_1),$$

$$k_3 = h f(t_n + c_3 h, y_n + a_{31} k_1 + a_{32} k_2),$$

:

$$k_s = h f(t_n + c_s h, y_n + a_{s1} k_1 + a_{s2} k_2 + \cdots + a_{s,s-1} k_{s-1})$$

↖ s : order of RK formula

0				
c_2	a_{21}			
c_3	$a_{31} \ a_{32}$			
:	:			
c_s	$a_{s1} \ a_{s2} \ \dots \ a_{s,s-1}$			
<hr/>				
b_1	b_2	\dots	b_{s-1}	b_s

describes family
of RK methods

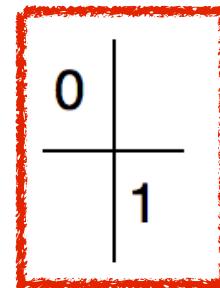
Runge-Kutta I

$$y_{n+1} = y_n + \sum_{i=1}^s b_i k_i,$$

Euler's Method when $s=1$ and $b_1=1$

where

$$k_1 = h f(t_n, y_n),$$



An RK 2: The Midpoint Method

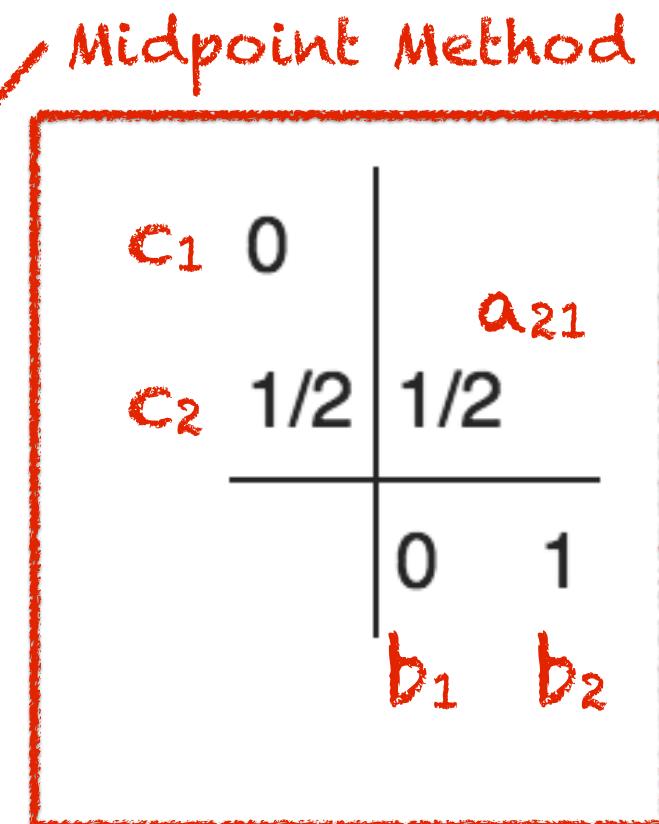
$$y_{n+1} = y_n + \sum_{i=1}^s b_i k_i,$$

where

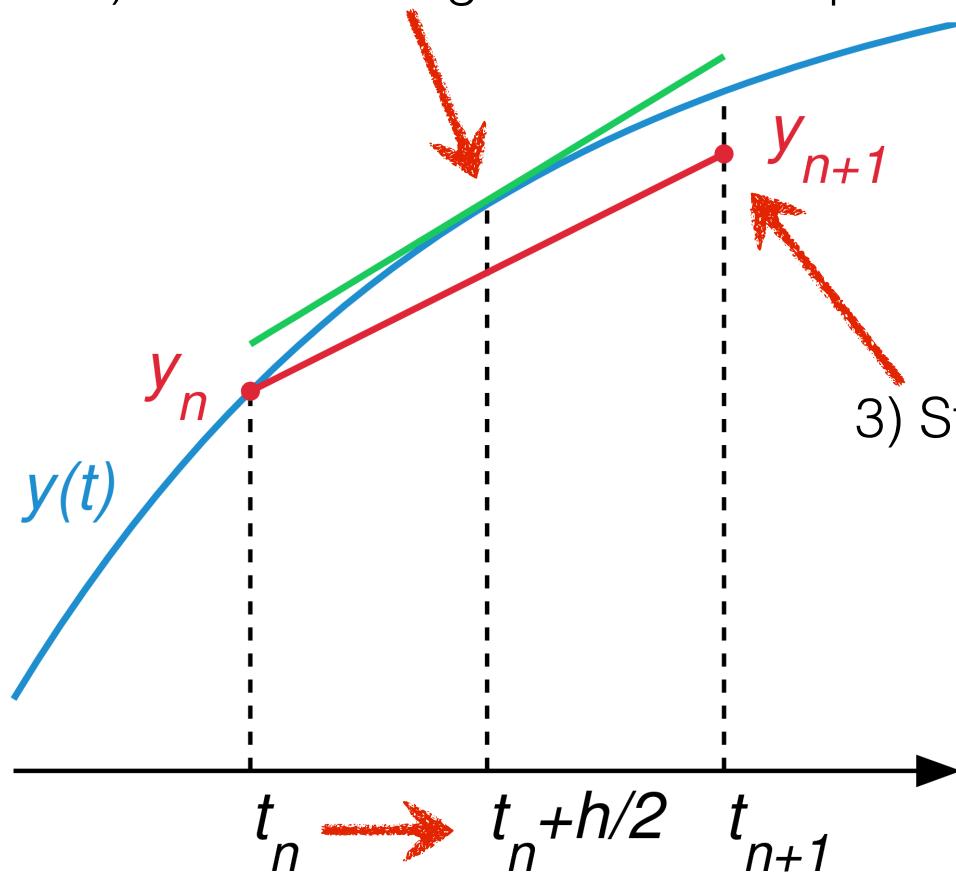
$$s=2 \quad k_1 = h f(t_n, y_n), \quad b_1=0$$

$$k_2 = h f(t_n + c_2 h, y_n + a_{21} k_1), \quad b_2=1$$

$$c_2=0.5 \quad a_{21}=0.5$$



2) Evaluate tangent at trial midpoint



1) Take “trial” step to midpoint

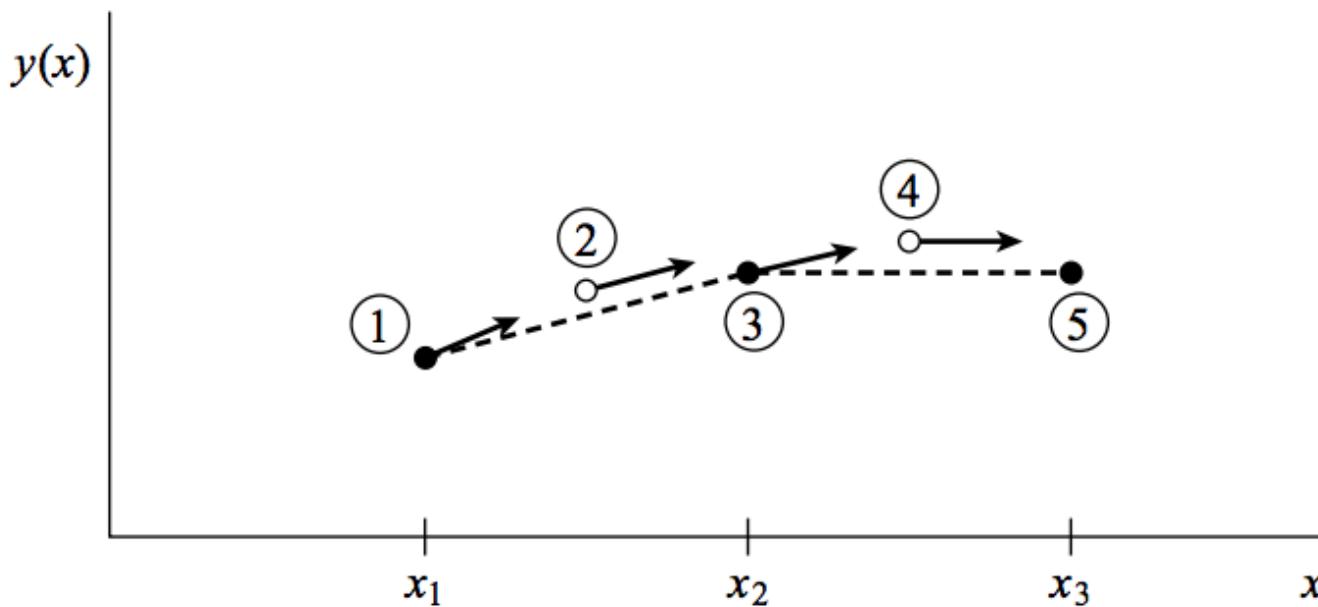
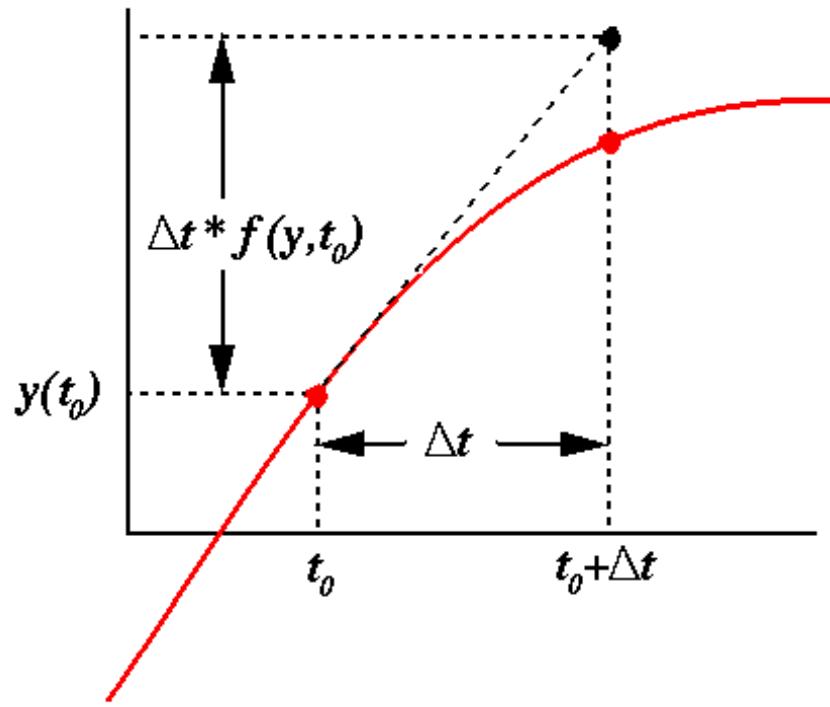
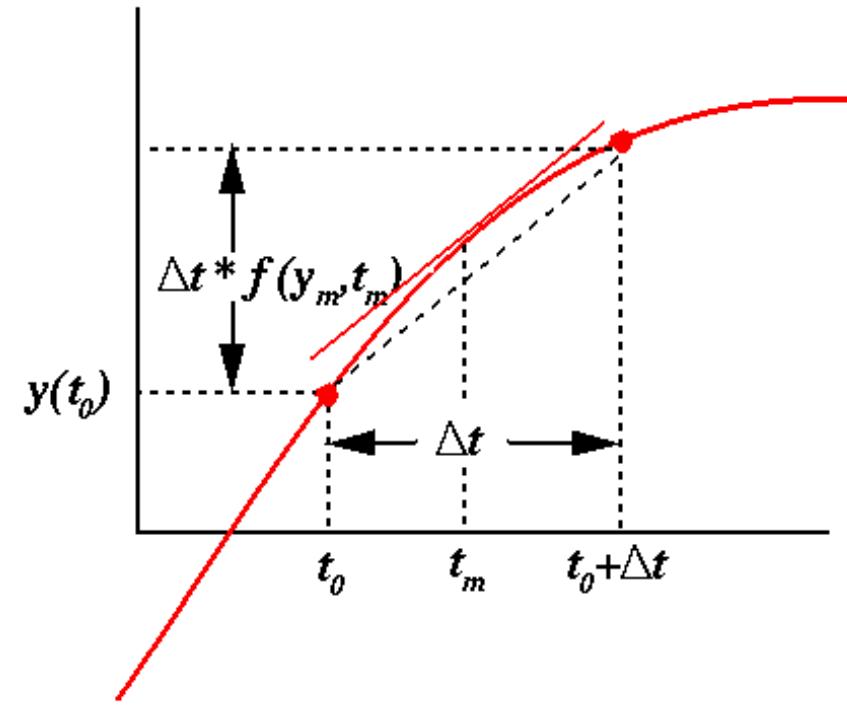


Figure 16.1.2. Midpoint method. Second-order accuracy is obtained by using the initial derivative at each step to find a point halfway across the interval, then using the midpoint derivative across the full width of the interval. In the figure, filled dots represent final function values, while open dots represent function values that are discarded once their derivatives have been calculated and used.



Euler



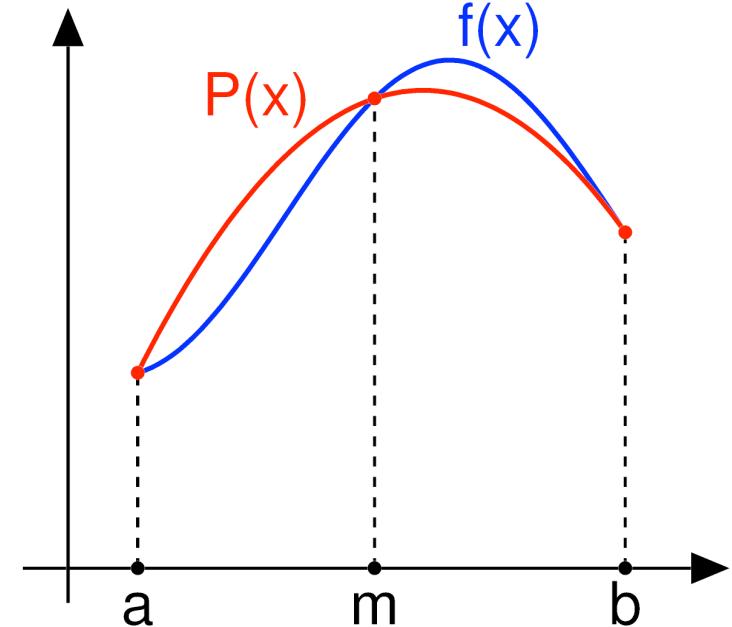
Midpoint

The “RK4”: Simpson’s Rule

a widely used version of fourth order Runge-Kutta
(known as “RK4”) implements Simpson’s rule:

$$\int_a^b f(x) dx \approx \frac{b-a}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right]$$

numerical approximation of definite integrals by
evaluating integral extents and midpoint



$$y_{n+1} = y_n + \sum_{i=1}^s b_i k_i,$$

where

$$k_1 = h f(t_n, y_n),$$

$$k_2 = h f(t_n + c_2 h, y_n + a_{21} k_1),$$

$$k_3 = h f(t_n + c_3 h, y_n + a_{31} k_1 + a_{32} k_2),$$

⋮

$$k_s = h f(t_n + c_s h, y_n + a_{s1} k_1 + a_{s2} k_2 + \cdots + a_{s,s-1} k_{s-1})$$

RK4 uses s=4

$$a_{21}=1/2, a_{32}=1/2, a_{43}=1,$$

$$c_1=0, c_2=1/2, c_3=1/2, c_4=1$$

$$b_1=1/6, b_2=1/3, b_3=1/3, b_4=1/6$$

0				
1/2	1/2			
1/2	0	1/2		
1	0	0	1	
	1/6	1/3	1/3	1/6

Runge-Kutta: RK4

$$y_{n+1} = y_n + \frac{1}{6}h (k_1 + 2k_2 + 2k_3 + k_4)$$

$$t_{n+1} = t_n + h$$

for $n = 0, 1, 2, 3, \dots$, using

$$k_1 = f(t_n, y_n), \text{ Slope at beginning of interval}$$

$$k_2 = f(t_n + \frac{1}{2}h, y_n + \frac{h}{2}k_1), \text{ Slope at trial midpoint } hk_1/2$$

$$k_3 = f(t_n + \frac{1}{2}h, y_n + \frac{h}{2}k_2), \text{ Slope at trial midpoint } hk_2/2$$

$$k_4 = f(t_n + h, y_n + hk_3). \text{ Slope at trial end of interval}$$

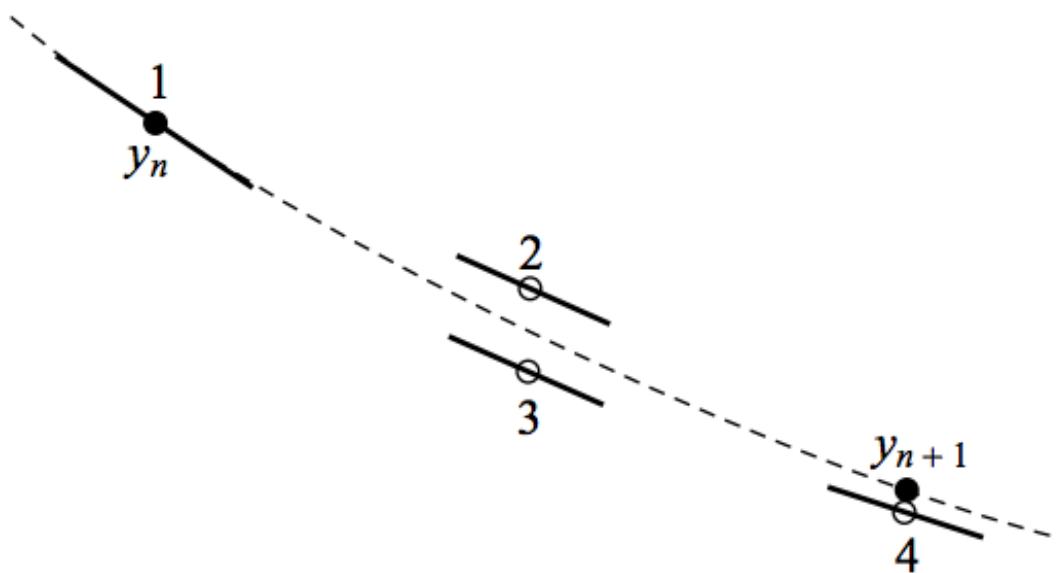


Figure 16.1.3. Fourth-order Runge-Kutta method. In each step the derivative is evaluated four times: once at the initial point, twice at trial midpoints, and once at a trial endpoint. From these derivatives the final function value (shown as a filled dot) is calculated. (See text for details.)

$$k(x_1) = x_t + 0$$

$$k(v_1) = v_t + 0$$

$c_1=0$

remember: 2nd order ODE
can represented as 1st order

$$y_{n+1} = y_n + \dot{y}_n \Delta t$$

$$\dot{y}_{n+1} = \dot{y}_n + \ddot{y}_n \Delta t$$

$$K(x_1) = x_t + 0$$

integrate position from t to trial
midpoint at $t+c_2h$ using $K(v_1)$ ($c_2=1/2$)

$$K(v_1) = v_t + 0$$
$$K(x_2) = K(x_1) + (a_{21} * K(v_1) * h)$$

$a_{21}=1/2$

$$k(x_1) = x_t + 0$$

$$k(v_1) = v_t + 0$$

$$k(x_2) = k(x_1) + (a_{21} * k(v_1) * h)$$

$$k(v_2) = k(v_1) + (a_{21} * \text{accel}(k(x_1)) * h)$$

integrate velocity from t to trial midpoint at $t+c_2h$

acceleration depends on system state
(i.e., position of pendulum)

$$k(x_1) = x_t + 0$$

$$k(v_1) = v_t + 0$$

$$k(x_2) = k(x_1) + (a_{21} * k(v_1) * h)$$

$$k(v_2) = k(v_1) + (a_{21} * \text{accel}(k(x_1)) * h)$$

$$k(x_3) = k(x_1) + (a_{32} * k(v_2) * h)$$

$a_{32}=1/2$

integrate position from t to second trial midpoint at $t+c_3h$ using $k(v_2)$ ($c_3=1/2$)

$$k(x_1) = x_t + 0$$

$$k(v_1) = v_t + 0$$

$$k(x_2) = k(x_1) + (a_{21} * k(v_1) * h)$$

$$k(v_2) = k(v_1) + (a_{21} * \text{accel}(k(x_1)) * h)$$

$$k(x_3) = k(x_1) + (a_{32} * k(v_2) * h)$$

$$k(v_3) = k(v_1) + (a_{32} * \text{accel}(k(x_2)) * h)$$



acceleration depends
on system state

$$k(x_1) = x_t + 0$$

$$k(v_1) = v_t + 0$$

$$k(x_2) = k(x_1) + (a_{21} * k(v_1) * h)$$

$$k(v_2) = k(v_1) + (a_{21} * \text{accel}(k(x_1)) * h)$$

$$k(x_3) = k(x_1) + (a_{32} * k(v_2) * h)$$

$$k(v_3) = k(v_1) + (a_{32} * \text{accel}(k(x_2)) * h)$$

$$k(x_4) = k(x_1) + (a_{43} * k(v_3) * h)$$

$a_{43}=1$

integrate position from t
to trial endpoint at $t+c_4h$
using $k(v_3)$ ($c_4=1$)

$$k(x_1) = x_t + 0$$

$$k(v_1) = v_t + 0$$

$$k(x_2) = k(x_1) + (a_{21} * k(v_1) * h)$$

$$k(v_2) = k(v_1) + (a_{21} * \text{accel}(k(x_1)) * h)$$

$$k(x_3) = k(x_1) + (a_{32} * k(v_2) * h)$$

$$k(v_3) = k(v_1) + (a_{32} * \text{accel}(k(x_2)) * h)$$

$$k(x_4) = k(x_1) + (a_{43} * k(v_3) * h)$$

$$k(v_4) = k(v_1) + (a_{43} * \text{accel}(k(x_3)) * h)$$

$$k(x_1) = x_t + 0$$

$$k(v_1) = v_t + 0$$

$$k(x_2) = k(x_1) + (a_{21} * k(v_1) * h)$$

$$k(v_2) = k(v_1) + (a_{21} * \text{accel}(k(x_1)) * h)$$

$$k(x_3) = k(x_1) + (a_{32} * k(v_2) * h)$$

$$k(v_3) = k(v_1) + (a_{32} * \text{accel}(k(x_2)) * h)$$

$$k(x_4) = k(x_1) + (a_{43} * k(v_3) * h)$$

$$k(v_4) = k(v_1) + (a_{43} * \text{accel}(k(x_3)) * h)$$

$$b_1=1/6, b_2=1/3, b_3=1/3, b_4=1/6$$

$$x_{t+h} = x_t + h * (b_1 * k(v_1) + b_2 * k(v_2) + b_3 * k(v_3) + b_4 * k(v_4))$$

$$k(x_1) = x_t + 0 \quad \text{velocity depends on acceleration } a(x)$$

$$k(v_1) = v_t + 0$$

$$k(x_2) = k(x_1) + (a_{21} * k(v_1) * h)$$

$$k(v_2) = k(v_1) + (a_{21} * \text{accel}(k(x_1)) * h)$$

$$k(x_3) = k(x_1) + (a_{32} * k(v_2) * h)$$

$$k(v_3) = k(v_1) + (a_{32} * \text{accel}(k(x_2)) * h)$$

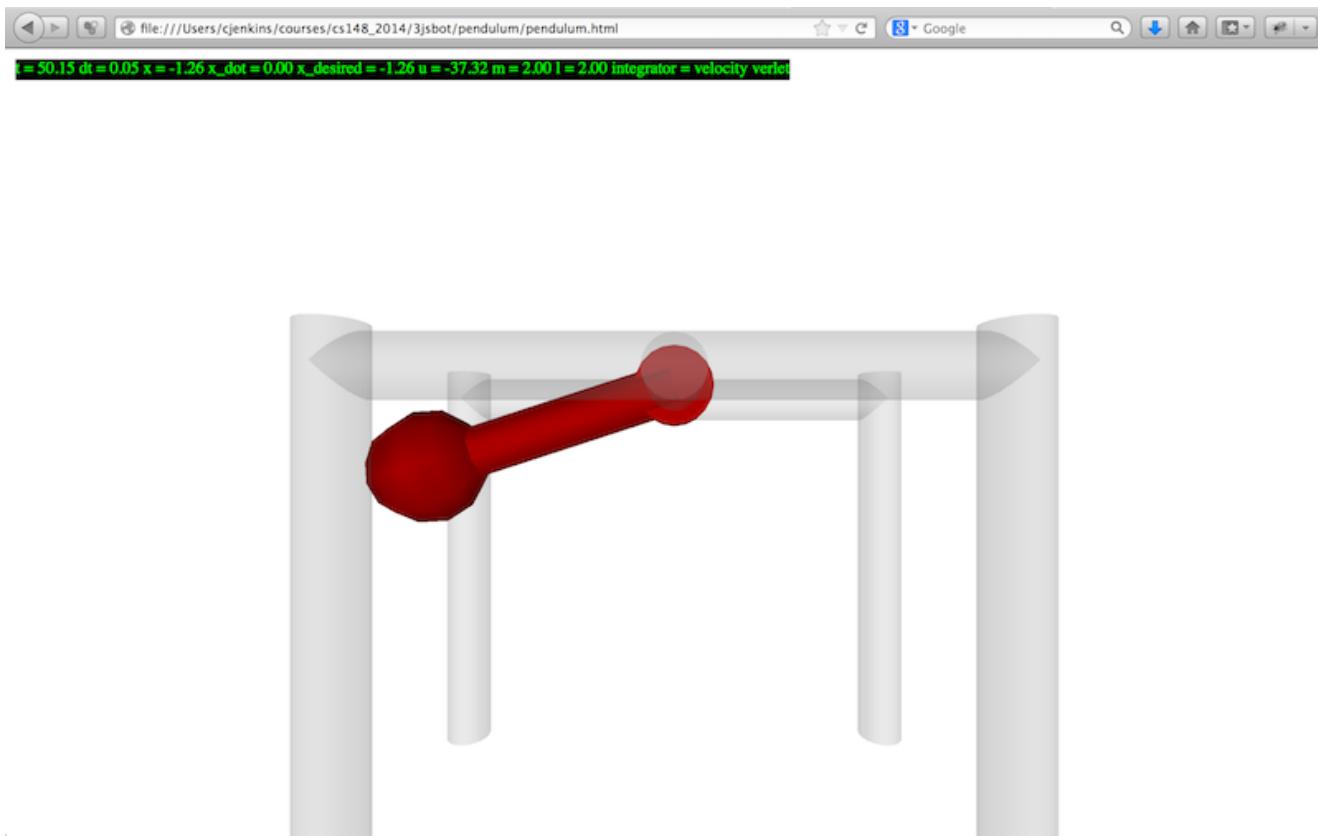
$$k(x_4) = k(x_1) + (a_{43} * k(v_3) * h)$$

$$k(v_4) = k(v_1) + (a_{43} * \text{accel}(k(x_3)) * h)$$

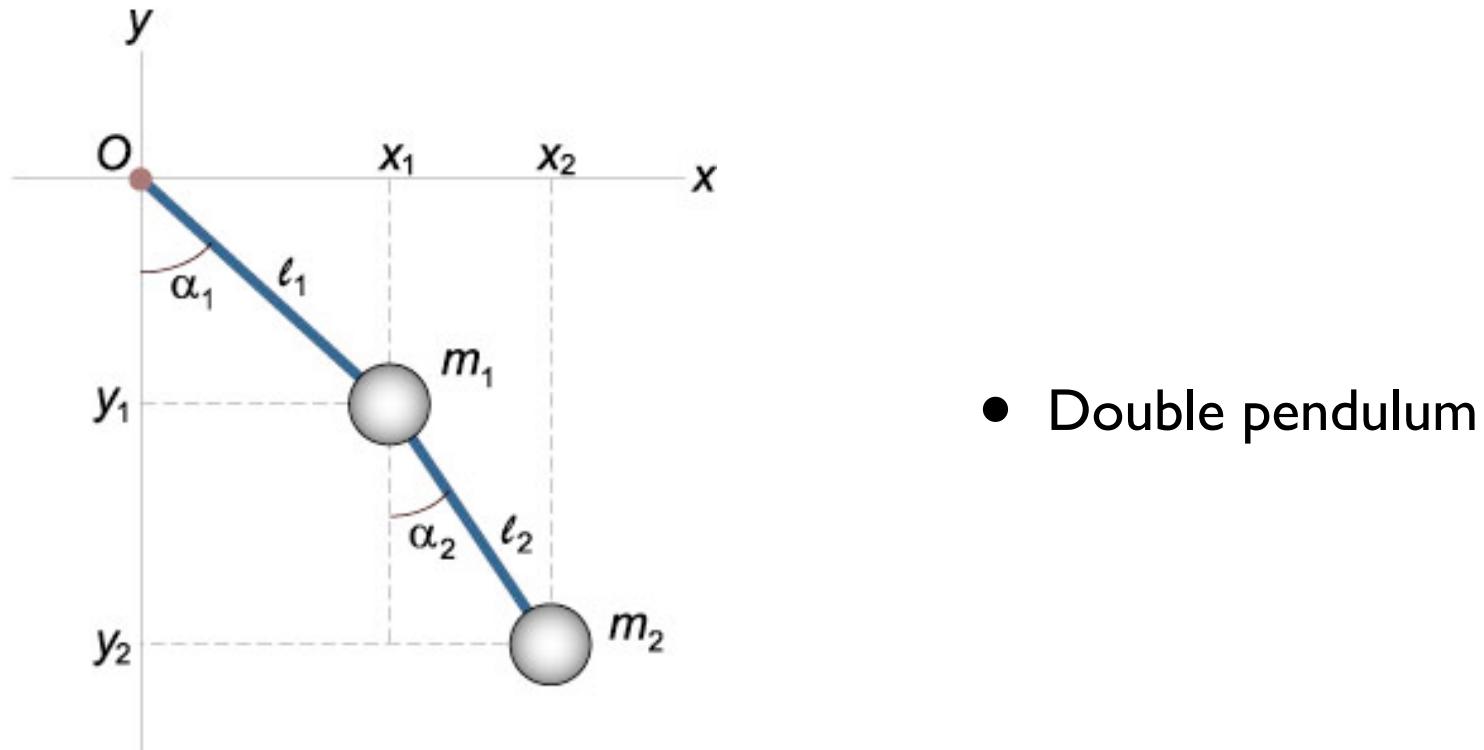


$$v_{t+h} = v_t + h * [b_1 * a(k(x_1)) + b_2 * a(k(x_2)) + b_3 * a(k(x_3)) + b_4 * a(k(x_4))]$$

Let's see what happens

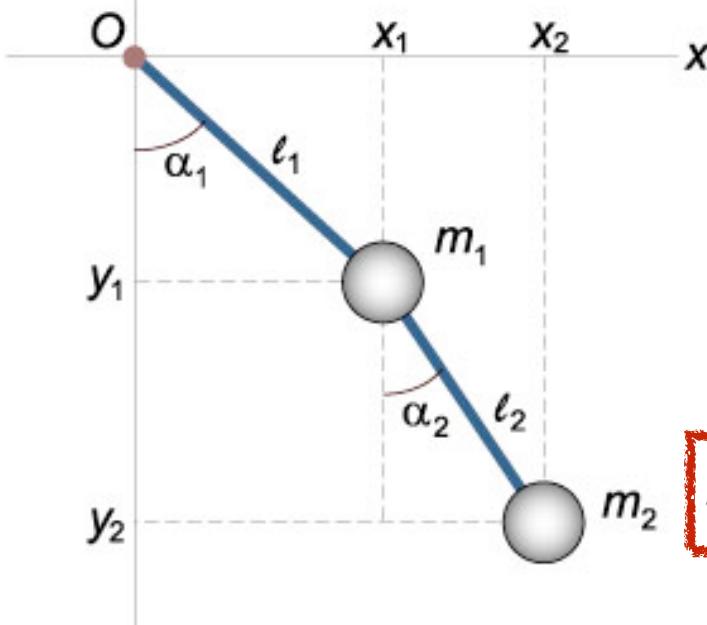


Can we add another link?



Locations of pendulum bobs

$$x_1 = l_1 \sin \alpha_1, \quad x_2 = l_1 \sin \alpha_1 + l_2 \sin \alpha_2, \quad y_1 = -l_1 \cos \alpha_1, \quad y_2 = -l_1 \cos \alpha_1 - l_2 \cos \alpha_2.$$



Lagrangian of pendulum bob positions

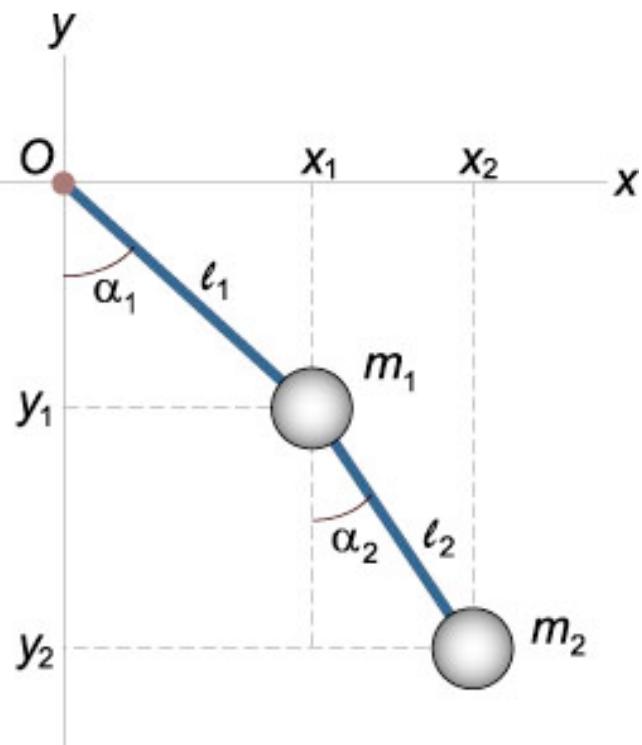
$$T = \frac{m_1 v_1^2}{2} + \frac{m_2 v_2^2}{2} = \frac{m_1 (\dot{x}_1^2 + \dot{y}_1^2)}{2} + \frac{m_2 (\dot{x}_2^2 + \dot{y}_2^2)}{2}, \quad V = m_1 g y_1 + m_2 g y_2.$$

$$L = T - V = T_1 + T_2 - (V_1 + V_2) = \frac{m_1}{2} (\dot{x}_1^2 + \dot{y}_1^2) + \frac{m_2}{2} (\dot{x}_2^2 + \dot{y}_2^2) - m_1 g y_1 - m_2 g y_2$$

Lagrangian in generalized coordinates (joint angles)

$$\dot{x}_1 = l_1 \cos \alpha_1 \cdot \dot{\alpha}_1, \quad \dot{x}_2 = l_1 \cos \alpha_1 \cdot \dot{\alpha}_1 + l_2 \cos \alpha_2 \cdot \dot{\alpha}_2,$$

$$\dot{y}_1 = l_1 \sin \alpha_1 \cdot \dot{\alpha}_1, \quad \dot{y}_2 = l_1 \sin \alpha_1 \cdot \dot{\alpha}_1 + l_2 \sin \alpha_2 \cdot \dot{\alpha}_2.$$



$$T_1 = \frac{m_1}{2} (\dot{x}_1^2 + \dot{y}_1^2) = \frac{m_1}{2} (l_1^2 \dot{\alpha}_1^2 \cos^2 \alpha_1 + l_1^2 \dot{\alpha}_1^2 \sin^2 \alpha_1) = \frac{m_1}{2} l_1^2 \dot{\alpha}_1^2,$$

$$T_2 = \frac{m_2}{2} (\dot{x}_2^2 + \dot{y}_2^2) = \frac{m_2}{2} [(l_1 \dot{\alpha}_1 \cos \alpha_1 + l_2 \dot{\alpha}_2 \cos \alpha_2)^2 + (l_1 \dot{\alpha}_1 \sin \alpha_1 + l_2 \dot{\alpha}_2 \sin \alpha_2)^2]$$

$$= \frac{m_2}{2} [l_1^2 \dot{\alpha}_1^2 \cos^2 \alpha_1 + l_2^2 \dot{\alpha}_2^2 \cos^2 \alpha_2 + 2l_1 l_2 \dot{\alpha}_1 \dot{\alpha}_2 \cos \alpha_1 \cos \alpha_2 + l_1^2 \dot{\alpha}_1^2 \sin^2 \alpha_1 + l_2^2 \dot{\alpha}_2^2 \sin^2 \alpha_2 + 2l_1 l_2 \dot{\alpha}_1 \dot{\alpha}_2 \sin \alpha_1 \sin \alpha_2]$$

$$= \frac{m_2}{2} [l_1^2 \dot{\alpha}_1^2 + l_2^2 \dot{\alpha}_2^2 + 2l_1 l_2 \dot{\alpha}_1 \dot{\alpha}_2 \cos(\alpha_1 - \alpha_2)],$$

$$V_1 = m_1 g y_1 = -m_1 g l_1 \cos \alpha_1,$$

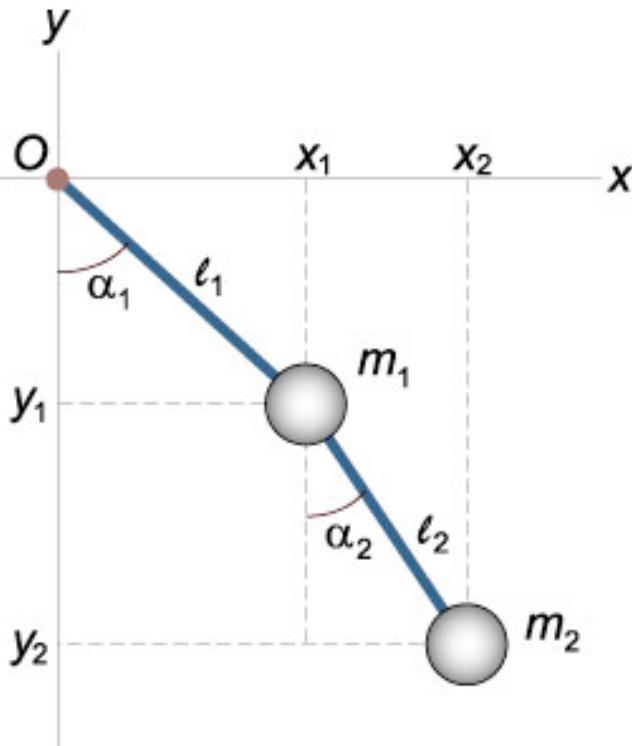
$$V_2 = m_2 g y_2 = -m_2 g (l_1 \cos \alpha_1 + l_2 \cos \alpha_2).$$

$$L = T - V = T_1 + T_2 - (V_1 + V_2) =$$

$$= \left(\frac{m_1}{2} + \frac{m_2}{2} \right) l_1^2 \dot{\alpha}_1^2 + \frac{m_2}{2} l_2^2 \dot{\alpha}_2^2 + m_2 l_1 l_2 \dot{\alpha}_1 \dot{\alpha}_2 \cos(\alpha_1 - \alpha_2) + (m_1 + m_2) g l_1 \cos \alpha_1 + m_2 g l_2 \cos \alpha_2.$$

Lagrangian equations of motion

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\alpha}_i} - \frac{\partial L}{\partial \alpha_i} = 0, \quad i = 1, 2.$$



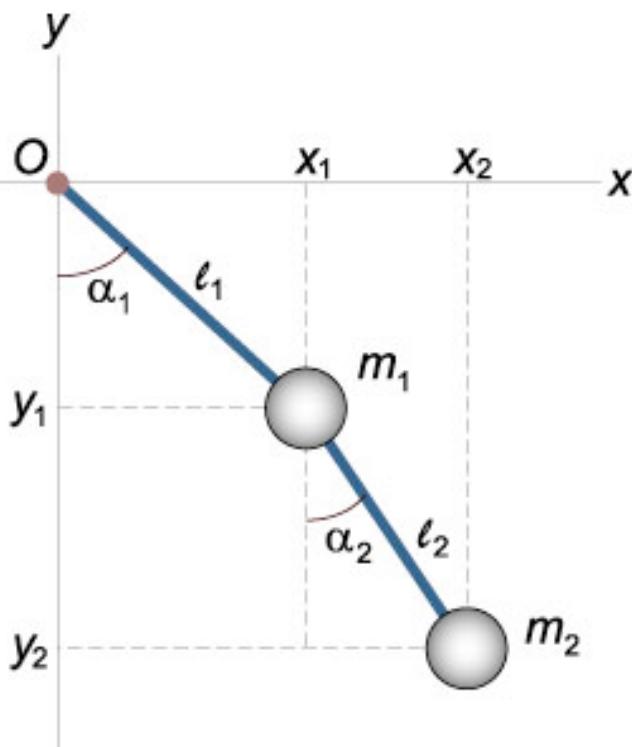
$$\begin{aligned}\frac{\partial L}{\partial \dot{\alpha}_1} &= (m_1 + m_2) l_1^2 \dot{\alpha}_1 + m_2 l_1 l_2 \dot{\alpha}_2 \cos(\alpha_1 - \alpha_2), & \frac{\partial L}{\partial \alpha_1} &= -m_2 l_1 l_2 \dot{\alpha}_1 \dot{\alpha}_2 \sin(\alpha_1 - \alpha_2) - (m_1 + m_2) g l_1 \sin \alpha_1, \\ \frac{\partial L}{\partial \dot{\alpha}_2} &= m_2 l_2^2 \dot{\alpha}_2 + m_2 l_1 l_2 \dot{\alpha}_1 \cos(\alpha_1 - \alpha_2), & \frac{\partial L}{\partial \alpha_2} &= m_2 l_1 l_2 \dot{\alpha}_1 \dot{\alpha}_2 \sin(\alpha_1 - \alpha_2) - m_2 g l_2 \sin \alpha_2.\end{aligned}$$

Lagrangian EOM for $i=1$

$$\begin{aligned}\frac{d}{dt} \left[(m_1 + m_2) l_1^2 \dot{\alpha}_1 + m_2 l_1 l_2 \dot{\alpha}_2 \cos(\alpha_1 - \alpha_2) \right] + m_2 l_1 l_2 \dot{\alpha}_1 \dot{\alpha}_2 \sin(\alpha_1 - \alpha_2) + (m_1 + m_2) g l_1 \sin \alpha_1 &= 0, \\ \Rightarrow (m_1 + m_2) l_1^2 \ddot{\alpha}_1 + m_2 l_1 l_2 \left[\ddot{\alpha}_2 \cos(\alpha_1 - \alpha_2) - \dot{\alpha}_2 \sin(\alpha_1 - \alpha_2) \cdot (\dot{\alpha}_1 - \dot{\alpha}_2) \right] \\ &\quad + m_2 l_1 l_2 \dot{\alpha}_1 \dot{\alpha}_2 \sin(\alpha_1 - \alpha_2) + (m_1 + m_2) g l_1 \sin \alpha_1 = 0, \\ \Rightarrow (m_1 + m_2) l_1^2 \ddot{\alpha}_1 + m_2 l_1 l_2 \ddot{\alpha}_2 \cos(\alpha_1 - \alpha_2) - \cancel{m_2 l_1 l_2 \dot{\alpha}_1 \dot{\alpha}_2 \sin(\alpha_1 - \alpha_2)} + m_2 l_1 l_2 \dot{\alpha}_2^2 \sin(\alpha_1 - \alpha_2) \\ &\quad + \cancel{m_2 l_1 l_2 \dot{\alpha}_1 \dot{\alpha}_2 \sin(\alpha_1 - \alpha_2)} + (m_1 + m_2) g l_1 \sin \alpha_1 = 0, \\ \Rightarrow (m_1 + m_2) l_1^2 \ddot{\alpha}_1 + m_2 l_1 l_2 \ddot{\alpha}_2 \cos(\alpha_1 - \alpha_2) + m_2 l_1 l_2 \dot{\alpha}_2^2 \sin(\alpha_1 - \alpha_2) + (m_1 + m_2) g l_1 \sin \alpha_1 &= 0. \\ (m_1 + m_2) l_1 \ddot{\alpha}_1 + m_2 l_1 l_2 \ddot{\alpha}_2 \cos(\alpha_1 - \alpha_2) + m_2 l_1 l_2 \dot{\alpha}_2^2 \sin(\alpha_1 - \alpha_2) + (m_1 + m_2) g \sin \alpha_1 &= 0.\end{aligned}$$

Lagrangian equations of motion

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\alpha}_i} - \frac{\partial L}{\partial \alpha_i} = 0, \quad i = 1, 2.$$



$$\begin{aligned}\frac{\partial L}{\partial \dot{\alpha}_1} &= (m_1 + m_2) l_1^2 \dot{\alpha}_1 + m_2 l_1 l_2 \dot{\alpha}_2 \cos(\alpha_1 - \alpha_2), & \frac{\partial L}{\partial \alpha_1} &= -m_2 l_1 l_2 \dot{\alpha}_1 \dot{\alpha}_2 \sin(\alpha_1 - \alpha_2) - (m_1 + m_2) g l_1 \sin \alpha_1, \\ \frac{\partial L}{\partial \dot{\alpha}_2} &= m_2 l_2^2 \dot{\alpha}_2 + m_2 l_1 l_2 \dot{\alpha}_1 \cos(\alpha_1 - \alpha_2), & \frac{\partial L}{\partial \alpha_2} &= m_2 l_1 l_2 \dot{\alpha}_1 \dot{\alpha}_2 \sin(\alpha_1 - \alpha_2) - m_2 g l_2 \sin \alpha_2.\end{aligned}$$

Lagrangian EOM for $i=2$

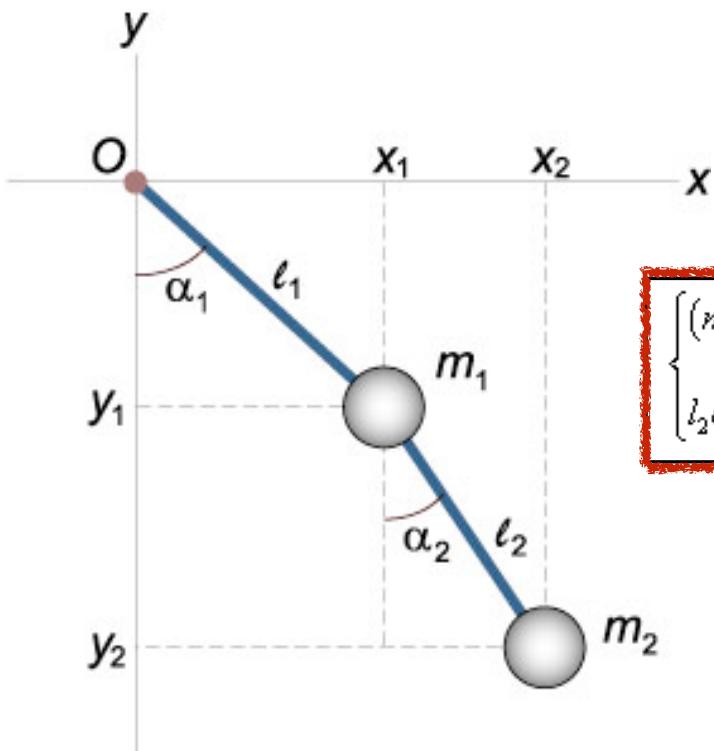
$$\frac{d}{dt} [m_2 l_2^2 \dot{\alpha}_2 + m_2 l_1 l_2 \dot{\alpha}_1 \cos(\alpha_1 - \alpha_2)] - m_2 l_1 l_2 \dot{\alpha}_1 \dot{\alpha}_2 \sin(\alpha_1 - \alpha_2) + m_2 g l_2 \sin \alpha_2 = 0,$$

$$\Rightarrow m_2 l_2^2 \ddot{\alpha}_2 + m_2 l_1 l_2 \ddot{\alpha}_1 \cos(\alpha_1 - \alpha_2) - m_2 l_1 l_2 \dot{\alpha}_1 \sin(\alpha_1 - \alpha_2) \cdot (\dot{\alpha}_1 - \dot{\alpha}_2) - m_2 l_1 l_2 \dot{\alpha}_1 \dot{\alpha}_2 \sin(\alpha_1 - \alpha_2) + m_2 g l_2 \sin \alpha_2 = 0,$$

$$\Rightarrow m_2 l_2^2 \ddot{\alpha}_2 + m_2 l_1 l_2 \ddot{\alpha}_1 \cos(\alpha_1 - \alpha_2) - m_2 l_1 l_2 \dot{\alpha}_1^2 \sin(\alpha_1 - \alpha_2) + \underline{m_2 l_1 l_2 \dot{\alpha}_1 \dot{\alpha}_2 \sin(\alpha_1 - \alpha_2)} - \underline{m_2 l_1 l_2 \dot{\alpha}_1 \dot{\alpha}_2 \sin(\alpha_1 - \alpha_2)} + m_2 g l_2 \sin \alpha_2 = 0,$$

$$\Rightarrow m_2 l_2^2 \ddot{\alpha}_2 + m_2 l_1 l_2 \ddot{\alpha}_1 \cos(\alpha_1 - \alpha_2) - m_2 l_1 l_2 \dot{\alpha}_1^2 \sin(\alpha_1 - \alpha_2) + m_2 g l_2 \sin \alpha_2 = 0.$$

$$l_2 \ddot{\alpha}_2 + l_1 \ddot{\alpha}_1 \cos(\alpha_1 - \alpha_2) - l_1 \dot{\alpha}_1^2 \sin(\alpha_1 - \alpha_2) + g \sin \alpha_2 = 0.$$



The 2 equations of motion
for double pendulum

$$\begin{cases} (m_1 + m_2)l_1\ddot{\alpha}_1 + m_2l_2\ddot{\alpha}_2 \cos(\alpha_1 - \alpha_2) + m_2l_2\dot{\alpha}_2^2 \sin(\alpha_1 - \alpha_2) + (m_1 + m_2)g \sin \alpha_1 = 0 \\ l_2\ddot{\alpha}_2 + l_1\ddot{\alpha}_1 \cos(\alpha_1 - \alpha_2) - l_1\dot{\alpha}_1^2 \sin(\alpha_1 - \alpha_2) + g \sin \alpha_2 = 0 \end{cases}$$

Substitute equations into each other to
solve for $\ddot{\alpha}_1$ and $\ddot{\alpha}_2$

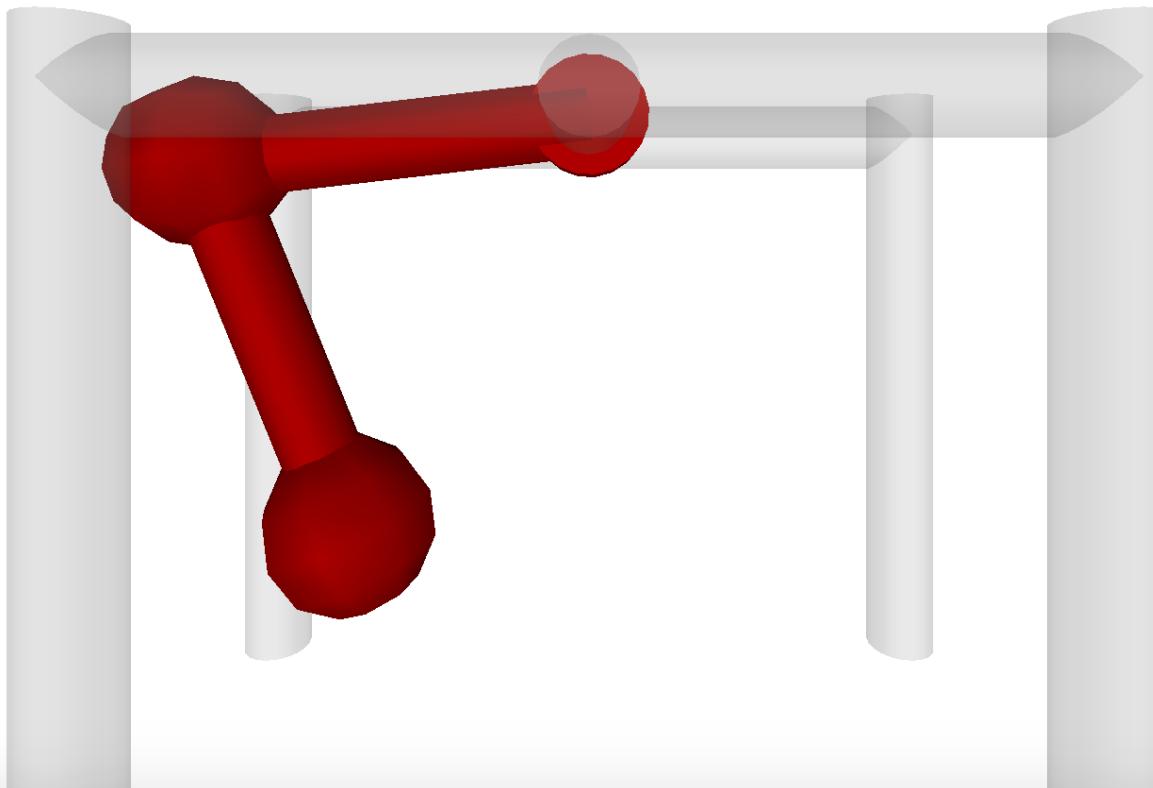
What happened to the torques?

```
System
t = 6119.20 dt = 0.05
integrator = runge-kutta
x1 = -1.46
x1_dot = -0.00
x2 = 1.84
x2_dot = -0.00
x1_desired = -1.46
x2_desired = 1.84
```

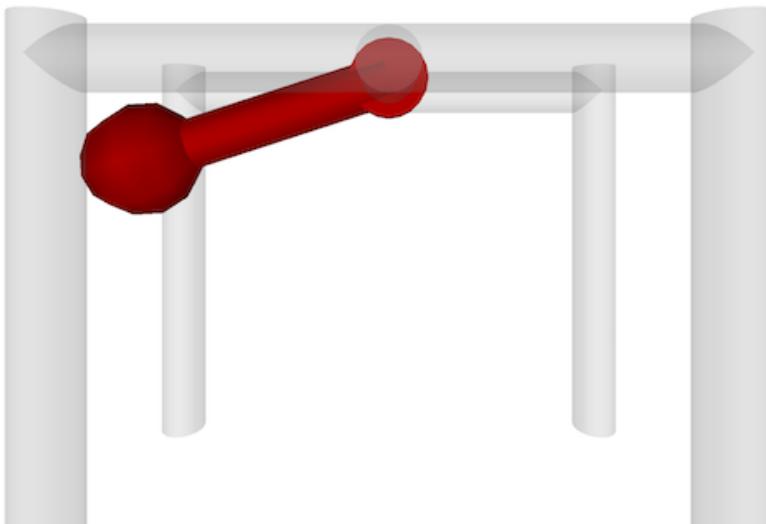
Pendularm2 by oharib

```
Pendulum
mass = 2.00
length = 2.00
gravity = 9.81
```

```
Keys
a/d - apply user force
1/2 - adjust desired angle 1
3/4 - adjust desired angle 2
c/x - toggle servo
s - disable servo
```



Project 2: Pendularm



[https://raw.githubusercontent.com/ohseejay/
kineval-stencil-f16/master/
project_pendularm/pendularm1.html](https://raw.githubusercontent.com/ohseejay/kineval-stencil-f16/master/project_pendularm/pendularm1.html)

```
function init() {...  
  pendulum = { // pendulum object  
    length:2.0,  
    mass:2.0,  
    angle:Math.PI/2,  
    angle_dot:0.0};  
  
  gravity = 9.81; // Earth gravity  
  t = 0; dt = 0.05; // init time  
  pendulum.control = 0; // motor  
  
  // next lecture: PID control  
  pendulum.desired = -Math.PI/2.5;  
  pendulum.desired_dot = 0;  
  pendulum.servo =  
    {kp:0, kd:0, ki:0};  
  ... }
```

```
function animate() { ...
if (numerical_integrator === "euler") {
    // STENCIL: Euler integrator }
else if (numerical_integrator === "verlet") {
    // STENCIL: basic Verlet integration }
else if (numerical_integrator === "velocity verlet") {
    // STENCIL: velocity Verlet }
else if (numerical_integrator === "runge-kutta") {
    // STENCIL: Runge-Kutta 4 integrator }
else { }
// set the angle of the pendulum
pendulum.geom.rotation.y = pendulum.angle;
t = t + dt; // advance time
...
}

function pendulum_acceleration(p,g) {
    // STENCIL: return acceleration from equations of motion }
```



Next up:
Control

(although no Janet Jackson)

UM EECS 398/598 - autorob.github.io