January 21, 2026

Continuous System Simulations

- "time" drives the simulation
- "time" is the continuous variable
- Differential equations
    - Ordinary Differential Equations (ODEs)

Three set of variables:

(i) state variables

(ii) input variables ← human/operator

(iii) output variables

ODE

$$F\left(t, x, \frac{dx}{dt}, \dots, \frac{d^{(n-1)}x}{dt^{(n-1)}}\right) = \frac{d^n x}{dt^n}$$

time (independent variable)

state variables

derivatives of x

Notations: $\frac{dx}{dt}$ derivative of x w.r.t. t

x'

Dx

Example:

① $mx'' = F$

$\Rightarrow m\frac{d^2x}{dt^2} = F$ order = 2

② $x' + 32x'' + \underline{\underline{x'''}} = 0$ order = 3

③ $x' + 34x = 32$ order = 1

Solving Differential Equations

E.g. $x'' = 2$

$$\Rightarrow \frac{d^2x}{dt^2} = 2$$

Solution is the value of $\underline{\underline{x = x(t)}}$ that satisfies this equation.

x depends upon t.

Key idea: integration.

(i) $x'' = 2$

(ii) Integrate once: $\underline{\underline{x'}} = 2t + c_1$

(iii) Integrate twice: $x = t^2 + tc_1 + c_2$

$$x = t^2 + \underbrace{tc_1}_{\uparrow} + \underbrace{c_2}_{\uparrow}$$

constants of integration

Q. How to solve for c_1 and c_2?

(i) initial conditions

(ii) boundary conditions

Initial Conditions: $x'(0) = 3$ $\quad x(0) = 2$

$$3 = (2)(0) + c_1 \qquad c_1 = 3$$

$$2 = (0)^2 + 0 + c_2 \qquad c_2 = 2$$

Solution: $\boxed{x = t^2 + 3t + 2}$

N^{th} order ODE Reducibility

$$x^{(n)} = F(t, x, x', \ldots, x^{(n-1)})$$

We can re-write this n^{th}-order ODE as a system of first-order differential equations by defining a new family of unknown functions.

$$x^{(i-1)} = x_i$$

Example:

2nd order ODE: $x'' = -g$

$$\Rightarrow \frac{d^2x}{dt^2} = -g$$ order = 2 state variable : x

This can be reduced to :

$$\frac{d}{dt}\left(\frac{dx}{dt}\right) = -g$$

$$\left.\begin{aligned} \frac{dx}{dt} &= \textcircled{v} \qquad \text{order} = 1 \\ \frac{dv}{dt} &= -g \qquad \text{order} = 1 \end{aligned}\right\} \textcircled{2}$$

state variables: x, v

Takeaways:

(i) All of our simulations only involve first order ODEs.

(ii) Even if we have higher order equations : we apply reducibility to get a system of first order equations.

(iv) Easier to solve numerically.

Very few numerical solvers exist for higher order ODEs.

Newton's Second Law of Motion

$$\underset{\substack{\uparrow \\ \text{accelaration}}}{a} \propto \underset{\substack{\uparrow \\ \text{force}}}{F} \quad \text{and} \quad a \propto \underset{\substack{\uparrow \\ \text{mass}}}{\frac{1}{m}}$$

$$\boxed{F = ma}$$

a : rate of change of velocity.

$\underset{\substack{\uparrow \\ \text{velocity}}}{v}$: rate of change of $\underset{\substack{\downarrow \\ x}}{\text{position}}$

$$v = \frac{dx}{dt}$$

$$a = \frac{dv}{dt} = \frac{d^2x}{dt^2}$$

$F = ma$ is a second-order differential equation

$$\Rightarrow F = m\frac{d^2x}{dt^2}$$

$$\Rightarrow \frac{d^2x}{dt^2} = \frac{F}{m} \xrightarrow{\text{Reduce}} \left.\begin{aligned} \frac{dx}{dt} &= v \\ \frac{dv}{dt} &= \frac{F}{m} \end{aligned}\right\} \text{2 first order equations}$$

state variable

x

state variables:

x, v

Solving ODEs numerically:

$$\frac{dx}{dt} = v \longrightarrow \Delta x = v\Delta t$$ — updates x

$$\frac{dv}{dt} = \frac{F}{m} \longrightarrow \underbrace{\Delta v = \frac{F}{m}\Delta t}_{\text{update rules}}$$ — updates v

update rules ⟵ require the current values of x and v.

Model: $$v(t+\Delta t) = v(t) + \frac{\overset{\text{input}}{F}}{\underset{\text{input / constant}}{m}}\Delta t$$

$$x(t+\Delta t) = x(t) + v\Delta t$$

state: x, v you need an update rule for each state variable.

$x(3) = ?$

$v(3) = ?$

$x(0), v(0)$ is given.

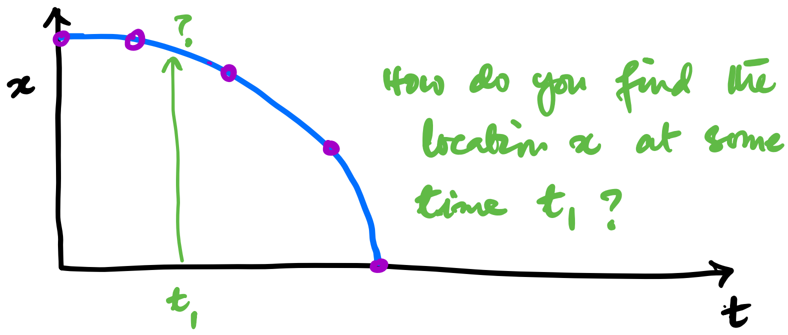
F, m are given.

Δt is also provided. Eg. $\Delta t = 1$

$$\underline{x(0), v(0)} \longrightarrow \underline{x(1), v(1)} \longrightarrow \underline{x(2), v(2)} \longrightarrow \underline{x(3), v(3)}$$

Choice of Δt

- if Δt is too small, the simulation can become very slow
- if Δt is too large, the simulation can become very inaccurate.
  - Advanced techniques can dynamically adjust Δt.



How do you find the exact location x at some time t_1?

- Δt has a relationship to the length (timescale) of simulation

  - molecular activity
  - echosystems
  - galaxy formation

- Δt that is used to advance the simulation is much smaller than the Δt that is used to display results

  output variables.

* Update rules are constructed from differential equations that model the dynamics of the system. Update rules are then used to see how state variables change over time.

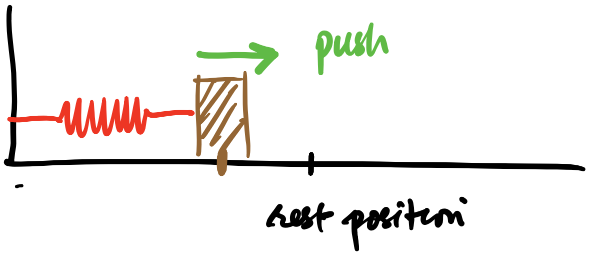
Mass spring system

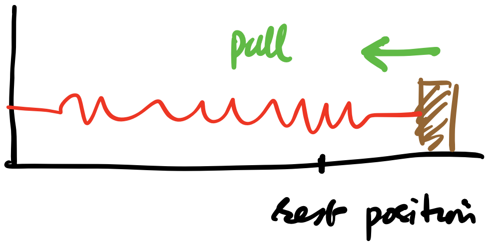
- Hook's law: "the extension is proportional to force"

$$F = -kx$$

displacement from the rest position.

spring constant





Modelling a mass-spring system

(i) How do I model the motion of mass that is attached to a spring?

$F = ma$

$F = -kx$

(a) Differential Equation (ODE) that captures dynamics

$$ma = -kx$$

$$\Rightarrow \boxed{m\frac{d^2x}{dt^2} = -kx}$$

$$\Rightarrow \frac{d^2x}{dt^2} = -\frac{k}{m}x \longrightarrow \left.\begin{aligned} \frac{dx}{dt} &= v \\ \frac{dv}{dt} &= -\frac{k}{m}x \end{aligned}\right\} \text{2 1st order ODEs.}$$

(b) Update rules.

$$\Delta x = v\Delta t \quad —①$$

$$\Delta v = -\frac{k}{m}x\Delta t \quad —②$$

1. Δt
2. k
3. m

(c)

$$\left.\begin{aligned} x(t+\Delta t) &= x(t) + v\Delta t \\ v(t+\Delta t) &= v(t) - \frac{k}{m}\underline{x(t)}\,\Delta t \end{aligned}\right] \text{Euler Integration}$$

* Mind your variables.

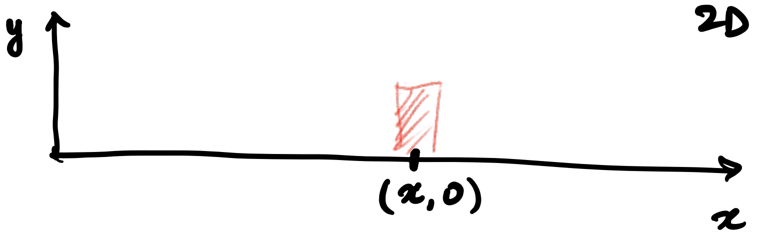
(i) $F = -k\underline{x}$ → x: displacement of the spring

(ii) $F = ma \rightarrow \frac{d^2x}{dt^2} = \frac{F}{m}$ → x: position of mass

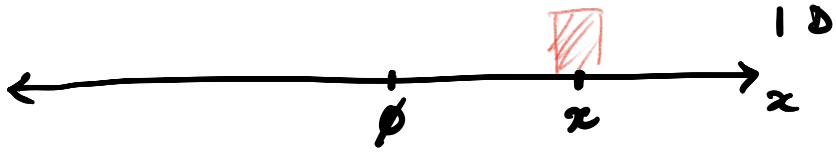
Displacement of the spring:

= | rest length − current length |

Position of the mass:

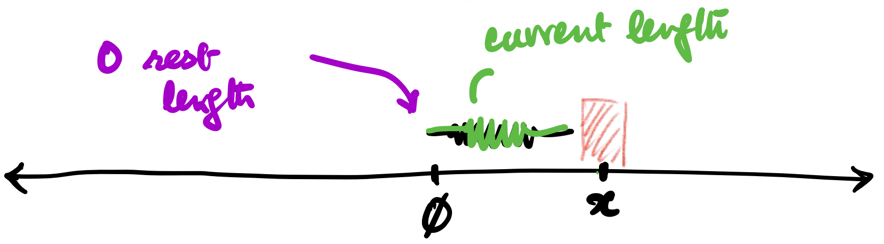


2D coordinate system



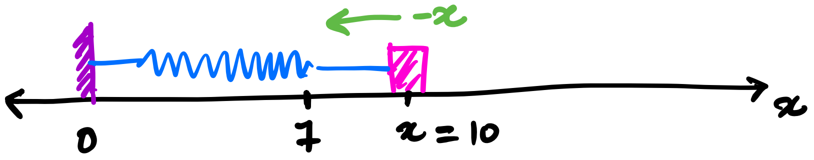
1 D coordinate system.

Q. Under what conditions would the update rules described above work.



Exercise :

Ⓐ

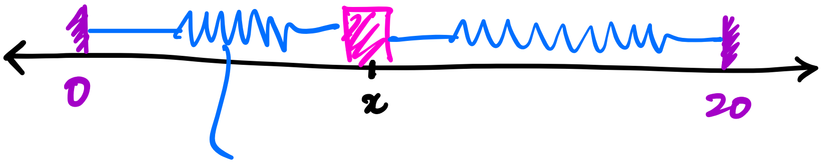


rest length = $\ell_{rest} = 7$

spring constant = $k = 0.3$

compute the magnitude and the direction of the force that this spring exerts on the mass m.

Ⓑ



$\ell^{1}_{rest} = 5$

$\ell^{2}_{rest} = 12$

$x = 9$

Mass Spring Damper

Mass experiences a damping force that is proportional to its velocity (friction).

velocity

① $F = -kx - cv$

spring displacement

damping constant

spring constant

We also have $F = ma$. — ②

Putting ① and ② together

$$ma = -kx - cv$$

$$\Rightarrow \boxed{m\frac{d^2x}{dt^2} = -kx - cv}$$

↓

(i) $\frac{dx}{dt} = v$ (ii) $m\frac{dv}{dt} = -kx - cv$ 1st order equation

Update rules:

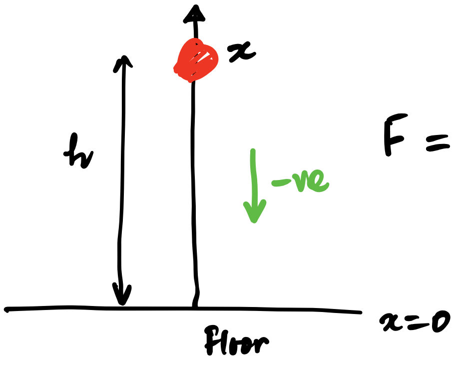
$$\Delta x = v \Delta t$$

$$\Delta v = -\frac{k}{m} x \Delta t - \frac{c}{m} v \Delta t$$

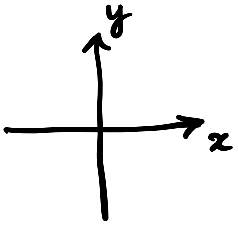
State variables:

x, v

Bouncing Ball



$F = -mg$ ①



We also have: $F = ma$ ②

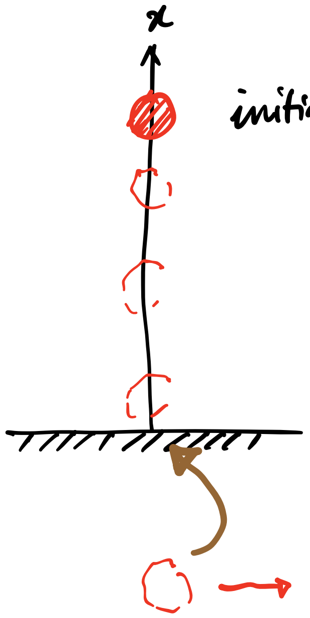
From ① and ② $\cancel{m}a = -\cancel{m}g$

$\Rightarrow a = -g$

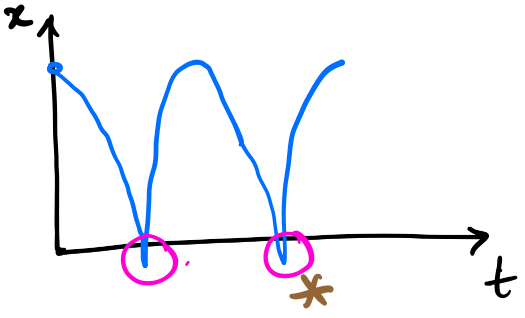
$\Rightarrow \frac{d^2x}{dt^2} = -g \longrightarrow$

$$\frac{dx}{dt} = v \quad \frac{dv}{dt} = -g$$

Model Dynamics.



initial conditions: $x = h, \; v = 0$



→ collision detected $\because x < 0$

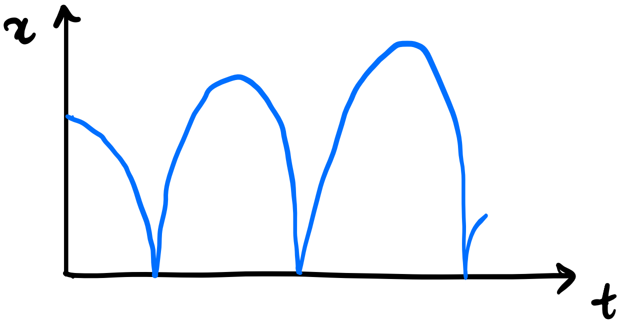
we are asked to compute a response.

Re-initialize the differential equations using the following initial conditions.

$x = x, \quad v = -v$

* How about we re-initialize the differential equations as follows

$x = 0, \; v = -v$

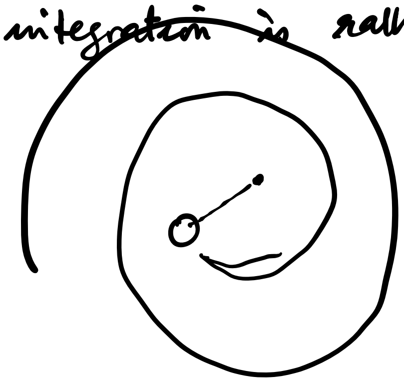


* Collision detection and response needs care. Since these affect the overall accuracy of your simulation.
* Exact collision times ?

$\Delta t \rightarrow 0$

Numerical Solver:

(i) Euler integration is rather poor. The system gains energy.



spiraling out of control.

(ii) Runge-kutta (RK4) method.

Much more numerically stable.

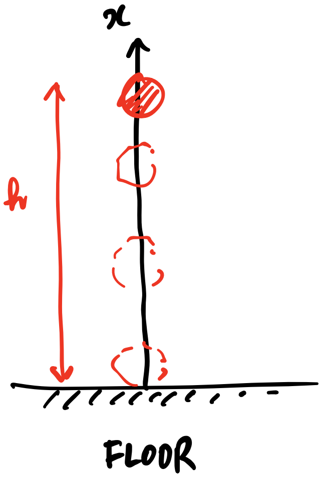
Error is proportional to Δt^4

Where as error for Euler integration is proportional to Δt.

Recall $\Delta t < 1$

* Using laws of physics to your advantage

Law of Conservation of Energy.



$x = h, v = 0$

Re initialize ODE as follows

$x = 0, \; v = \underline{\underline{-v}} \rightsquigarrow v = -\sqrt{2gh}$

↓↓ This $|v|$ is much larger than the true v at floor height.

⟶ collision is detected. We need a response!

Potential Energy: mgh

K.E. $\frac{1}{2}mv^2$

Total energy: $mgh + \frac{1}{2}mv^2$

Q. What is the K.E. at height h: 0

What is the P.E. at floor: 0

① − $mgh + 0 = mgh$

② − $0 + \frac{1}{2}mv^2 = \frac{1}{2}mv^2$

$$\cancel{m}gh = \frac{1}{2}\cancel{m}v^2$$

$$\Rightarrow \quad v = \sqrt{2gh} \quad *$$

Q. When will this approach fail?

A. Damping / friction.