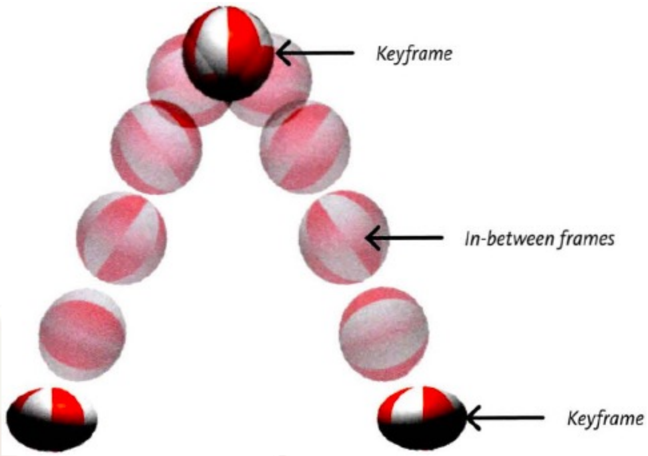
Animation for Computer Games
COMP 477/6311

Prof. Tiberiu Popa

Physics-based Animation
A lot of math and a little bit of physics

The Basics

- Keyframe animation is v. tedious
- Alternatives are:
  - Performance capture
    - look at real-life performances
    - Record
    - Retarget
  - Physics-based animation
    - Compute the animation as the result of a physics simulation



The Basics

- Newton's Law: $F = ma$
- $\ddot{x} = M^{-1}F$ (eq 1)
- $x \in \mathbb{R}^{3\times n}$
- $M \in \mathbb{R}^{n\times n}$, diagonal with the mass of each particle/vertex on diagonal
- $F \in \mathbb{R}^{3\times n}$

The Basics

- $\ddot{x} = M^{-1}F$ (eq 1)
- IVP, second order ODE
- Solving differential equations is a field in itself as they are very very popular in physics
- Lots of tools available → we will explore some of them
- Second order ODE are difficult?
- **What can we do to simplify it?**

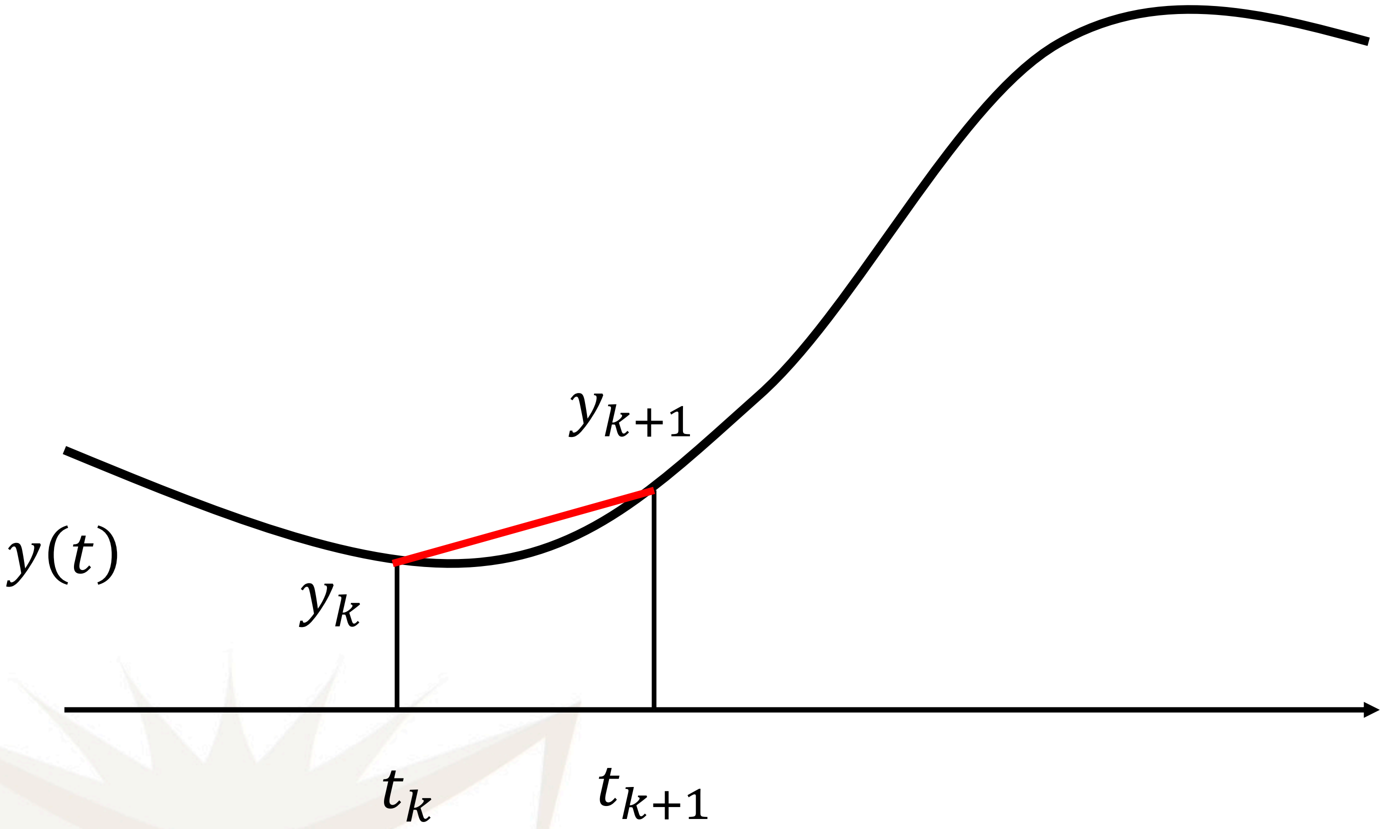
The Basics

- $y = \binom{x}{v}$
- $\ddot{x} = M^{-1}F$ (eq 1) becomes
- $\dot{y} = \binom{v}{M^{-1}F}$ (eq 2)
- IVP, 1st order PDE
- **What next?**

The Basics

- Taylor series:
- $y(t+\Delta t) = \sum_{i=0}^{\infty} \frac{y^{(i)}(t)(\Delta t^{i})}{i!} = y(t) + \dot{y}(t)\Delta t + \frac{\ddot{y}(t)(\Delta t^{2})}{2} + \cdots$
- If Δt is small, terms of the series decrease and are → to 0
- If Δt is small we can approximate the series by truncating it
- We cannot compute the true function
- Estimate it at discrete numerical intervals:
- $y_k \approx y(t_k)$
- $t_k = t_0 + k \cdot \Delta t$
- $k \geq 0$

Intuition

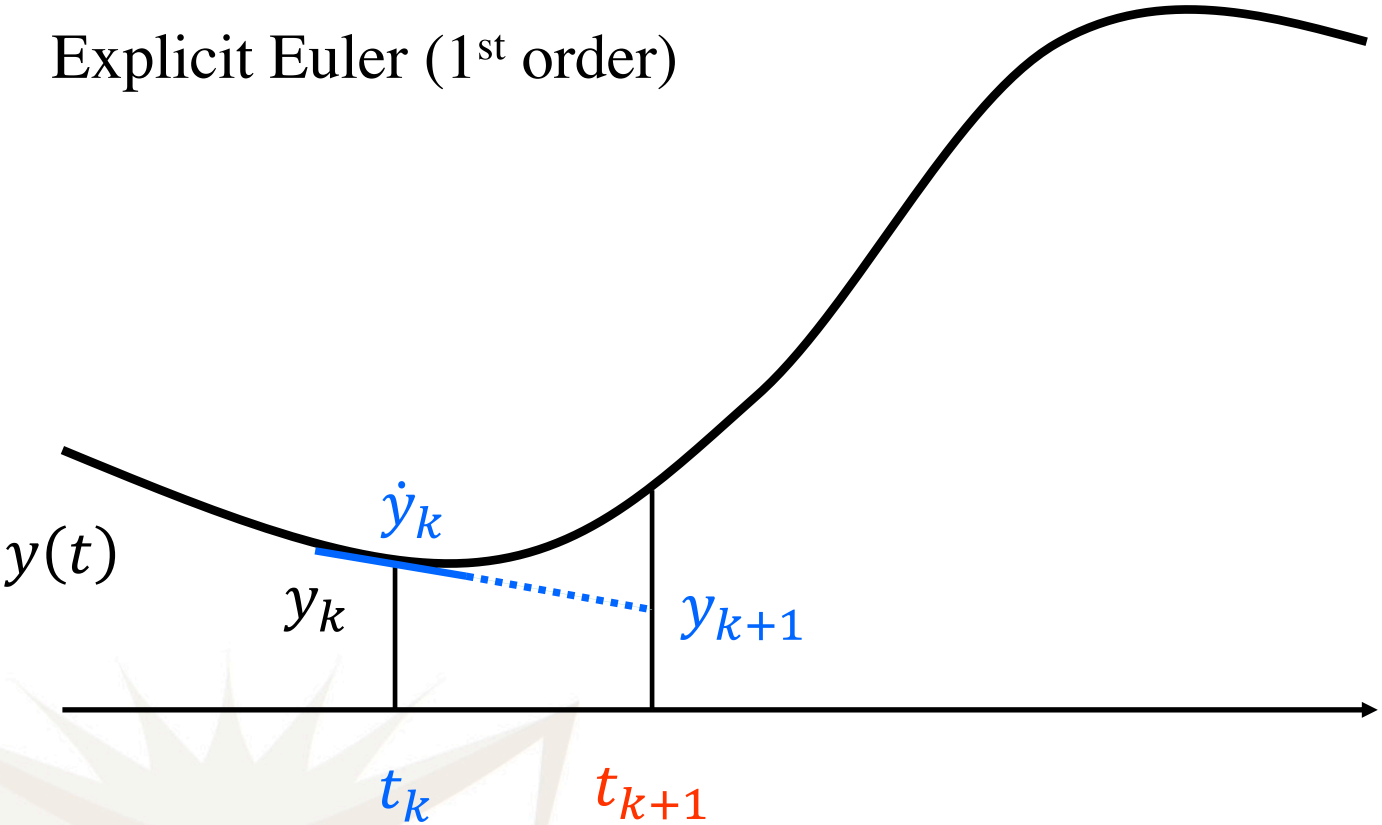


Explicit/Forward Euler (1st order)

- $\dot{y} = \begin{pmatrix} v \\ M^{-1}F \end{pmatrix}$ (eq 2)
- $y_{k+1} = y_k + \dot{y}_k \Delta t$ (eq 3)
- $\begin{pmatrix} x_{k+1} \\ v_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ v_k \end{pmatrix} + \Delta t \begin{pmatrix} v_k \\ M^{-1}F_k \end{pmatrix}$ (eq 4)
- Easy?
- YES → iterate from t_0 in steps of size Δt computing x, v → everything to the right of the eq3 or eq4 are known!!!!
- Don't forget → IVP
- $y(t_0) = \begin{pmatrix} x(t_0) \\ v(t_0) \end{pmatrix}$ assumed known

Intuition

Explicit Euler (1st order)

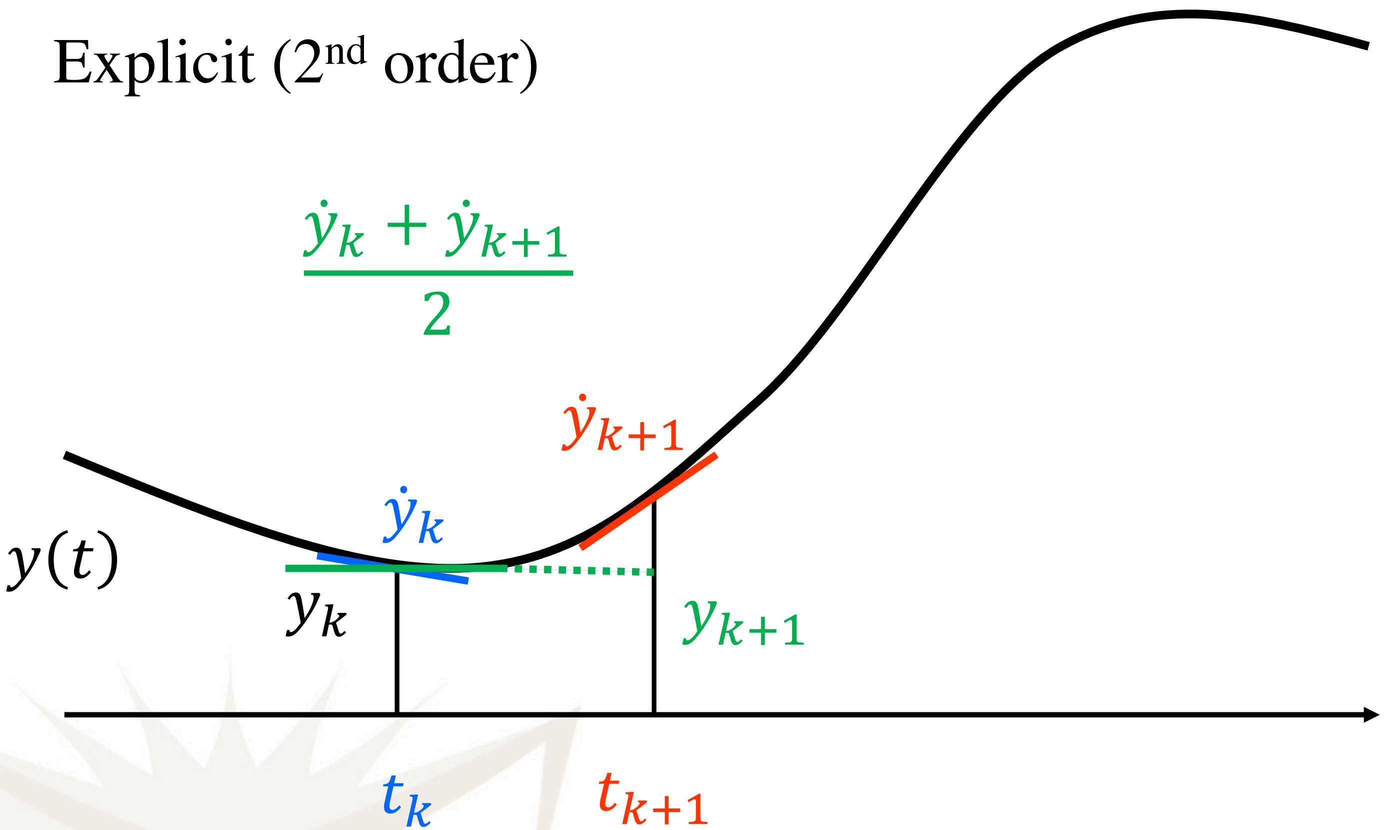


Explicit Euler (1st order)

- $y_{k+1} = y_k + \dot{y}_k \Delta t$ (eq 3)
- Easiest scheme but
  - Error accumulates
  - Unstable unless tiny time steps

Intuition

Explicit (2nd order)

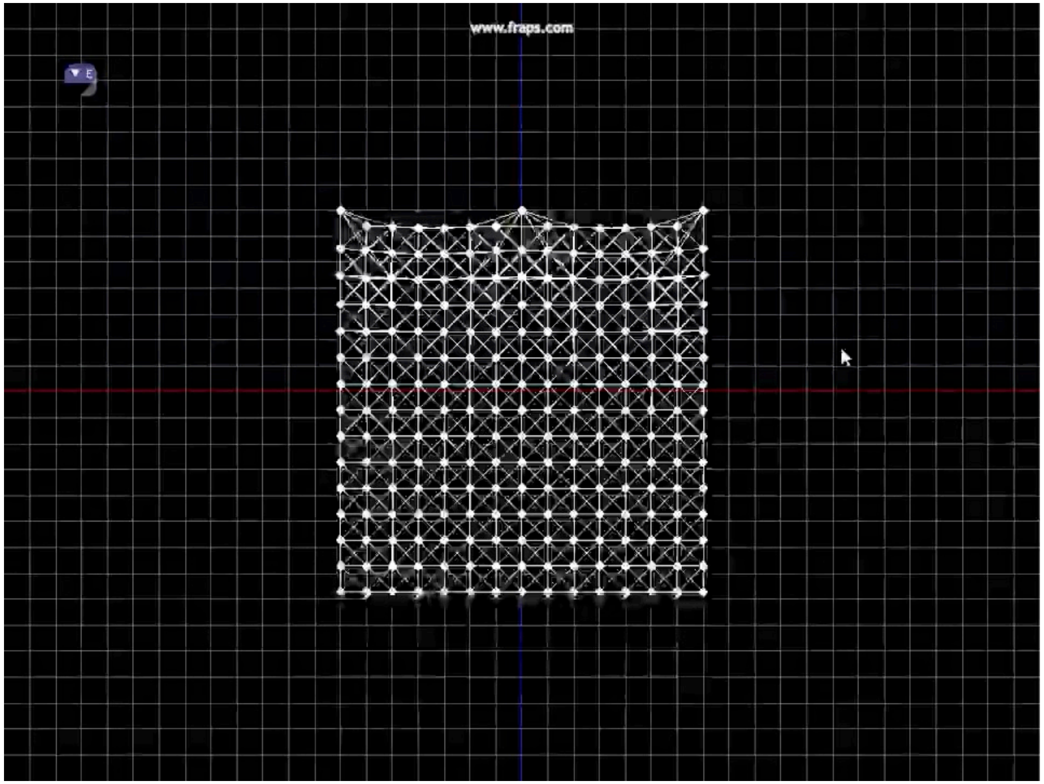


Explicit (2nd order)

- $y = \binom{x}{v}$
- $\dot{y} = \binom{v}{M^{-1}F}$ (eq 2)
- $y_{k+1} = y_k + \dot{y}_k \Delta t + \frac{\ddot{y}_k}{2} \Delta t^2$ (eq 5)
- Use definition of derivative:
- $\ddot{y}(t) = \lim_{\Delta t \to 0} \frac{\dot{y}(t+\Delta t) - \dot{y}(t)}{\Delta t}$ (eq 6)
- Combine eq 5 and eq 6:
- $y_{k+1} = y_k + \dot{y}_k \Delta t + \frac{\dot{y}_{k+1} - \dot{y}_k}{2} \Delta t$
- $y_{k+1} = y_k + \frac{\dot{y}_{k+1} + \dot{y}_k}{2} \Delta t$ (eq 7)
- We don't know yet $\dot{y}_{k+1}$

Implicit Euler

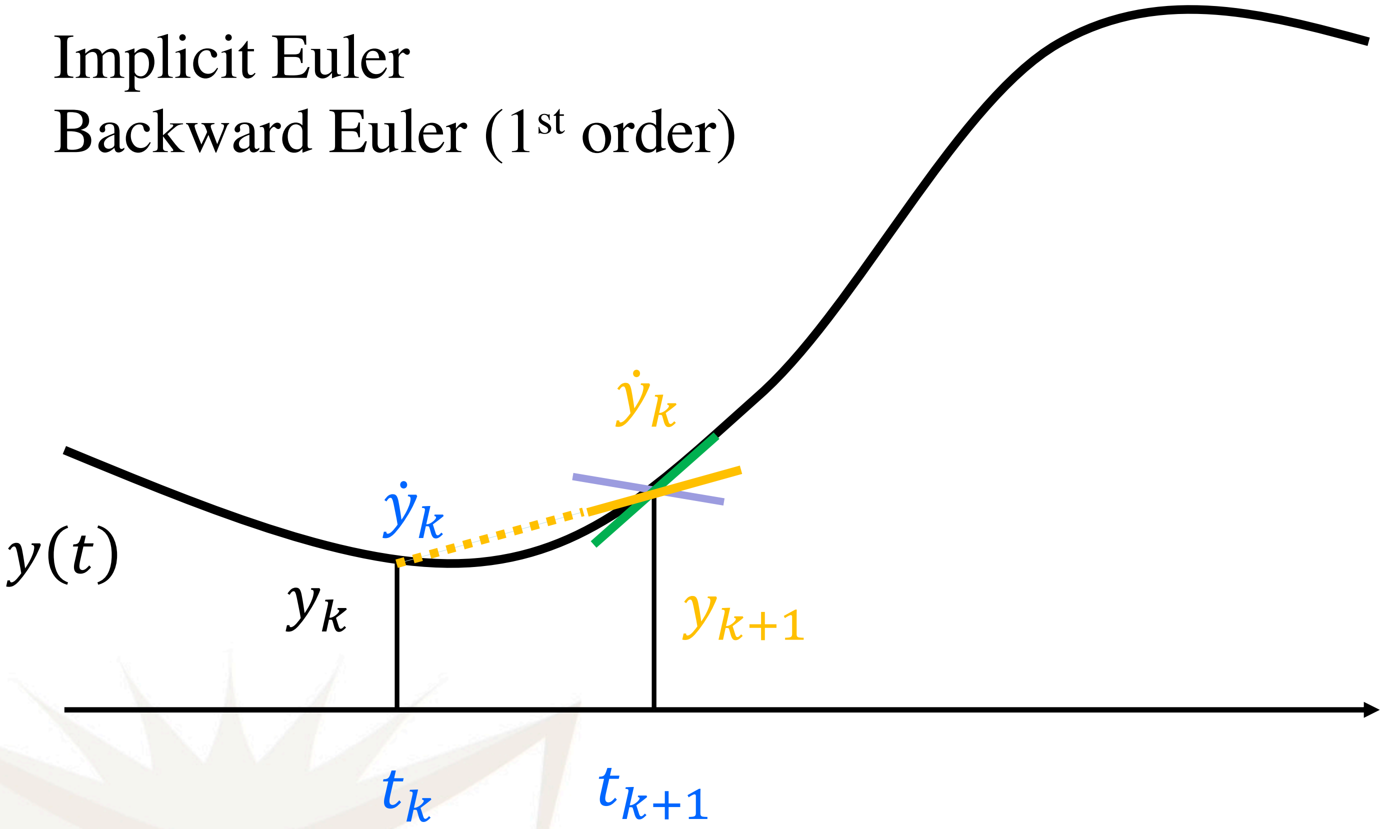
- Review explicit Euler: $y_{k+1} = y_k + \dot{y}_k \Delta t$ (eq 3)
- Easiest scheme but
  - Not accurate
  - Unstable



https://www.youtube.com/watch?v=rN6XUM4KOYo

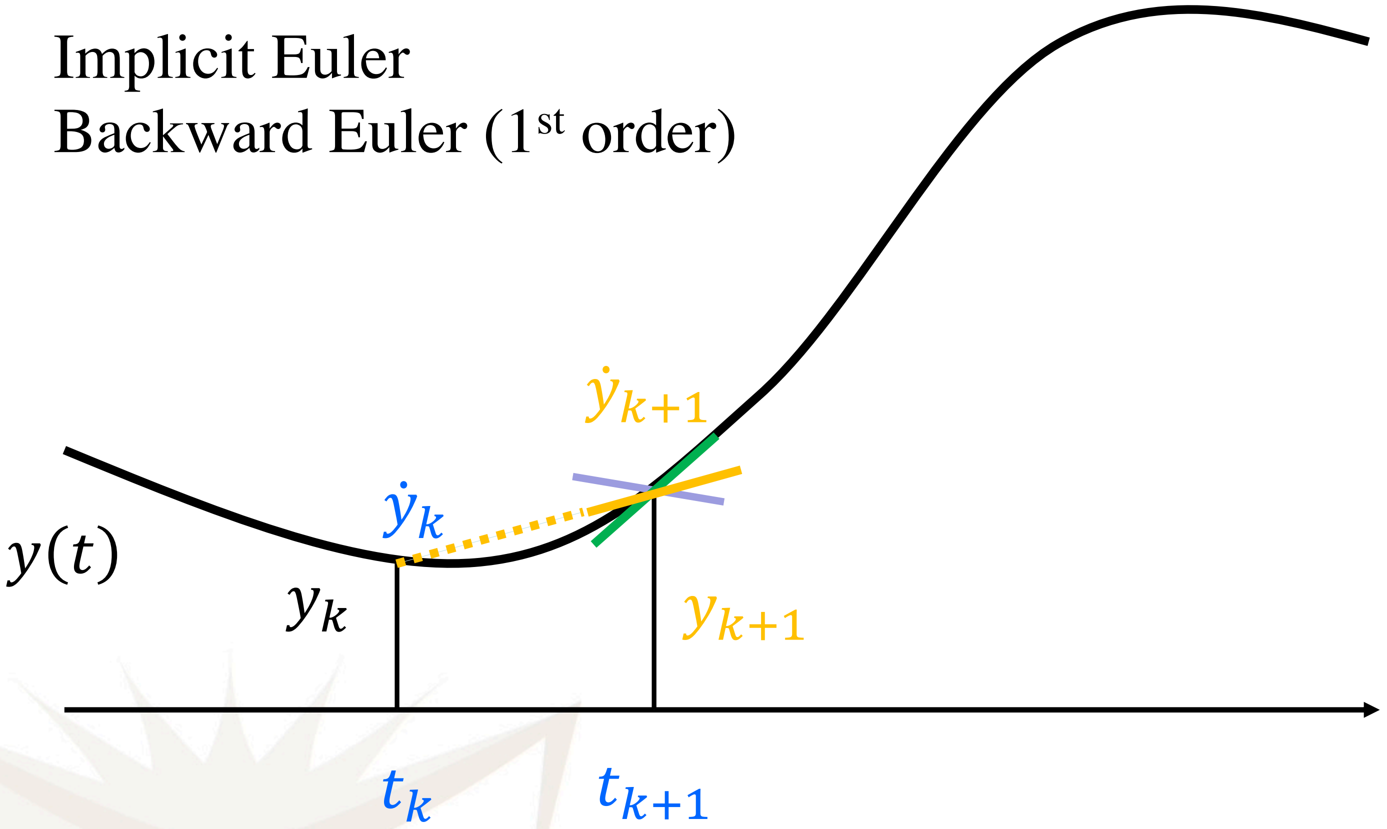
Intuition

Implicit Euler
Backward Euler (1st order)



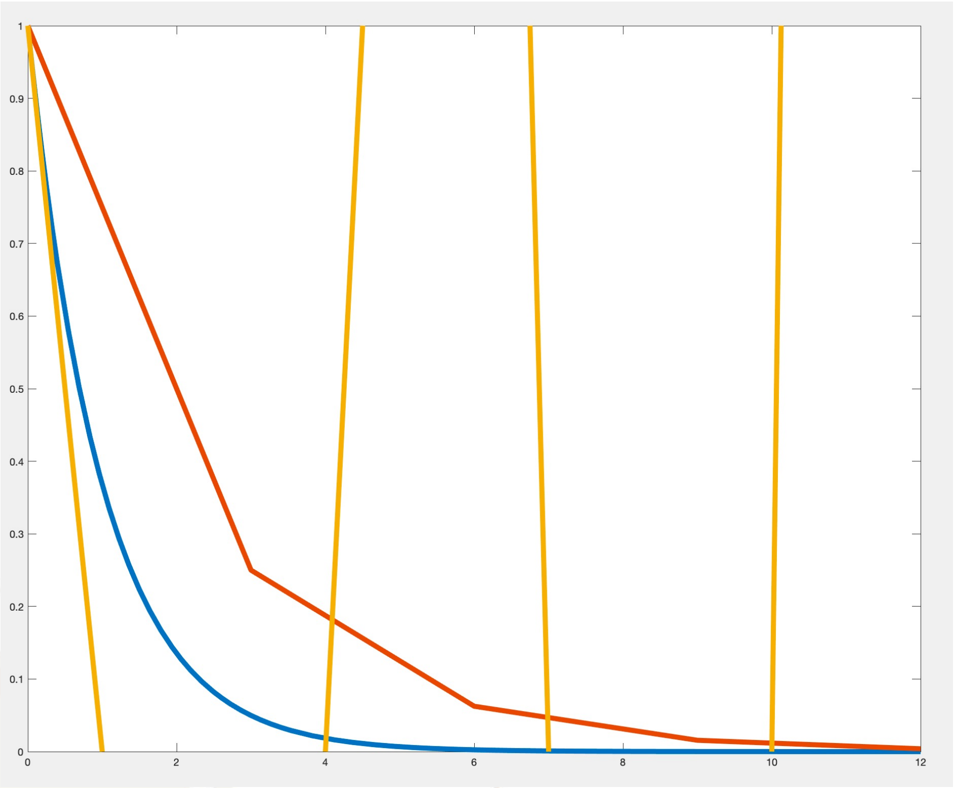
Intuition

Implicit Euler
Backward Euler (1st order)



Time integration example

- On the whiteboard



The Basics

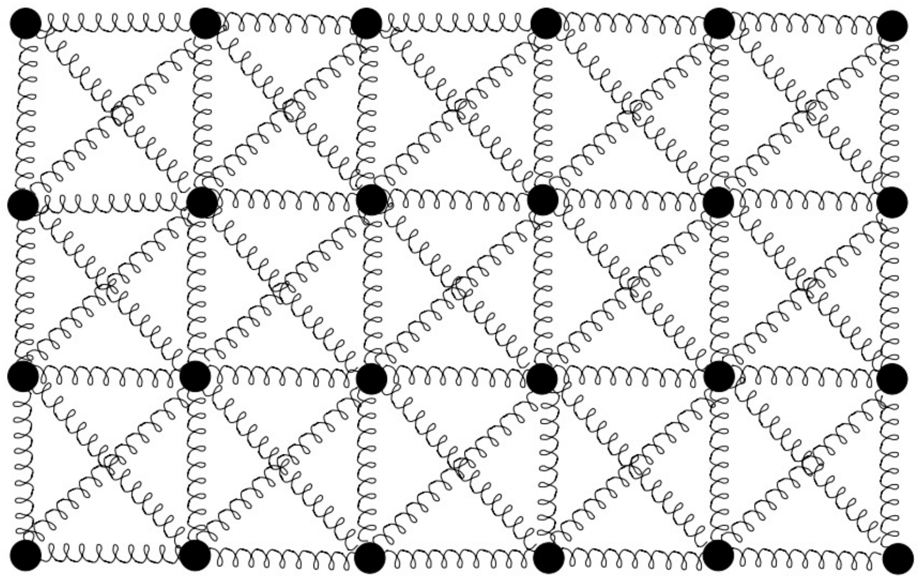
- Integration techniques are applied to all physics simulations
- Differences are in: forces, discretization
  - Rigid bodies
    - gravity
    - friction
  - Elastic bodies (e.g. cloth)
    - gravity
    - elastic forces
    - friction
  - Fluids
    - Navier-Stokes equations

Elastic objects

- Elastic objects when deformed tend to return to return to the rest pose
- Internal forces push them towards the rest pose (stress)
- Deformation from rest-pose (strain)
- Hook's law: linear relationship between stress and strain
- A little abstract because we need a discretization

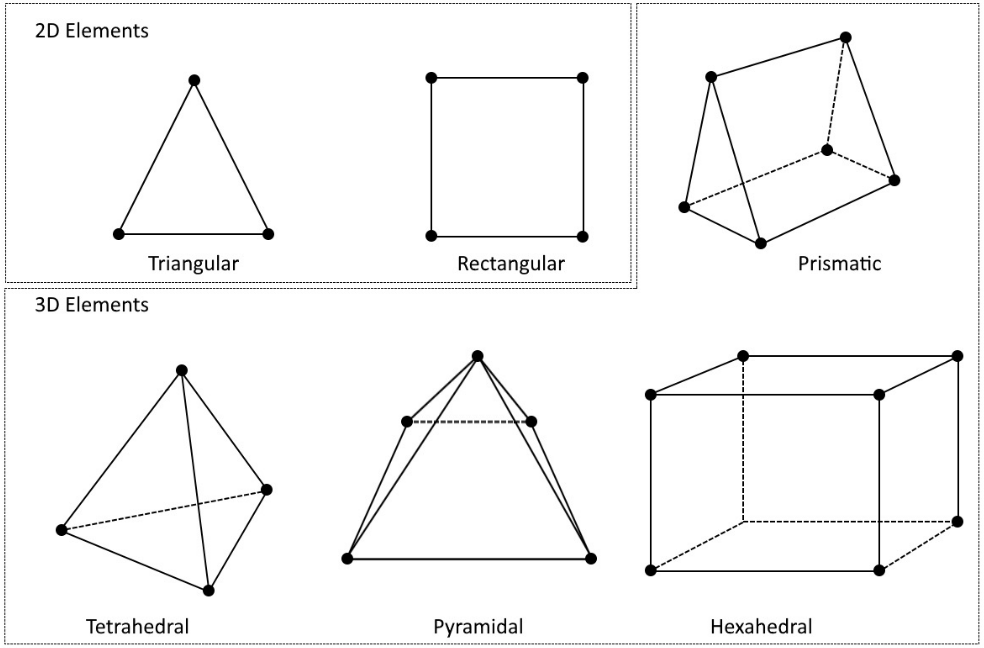
Elastic objects

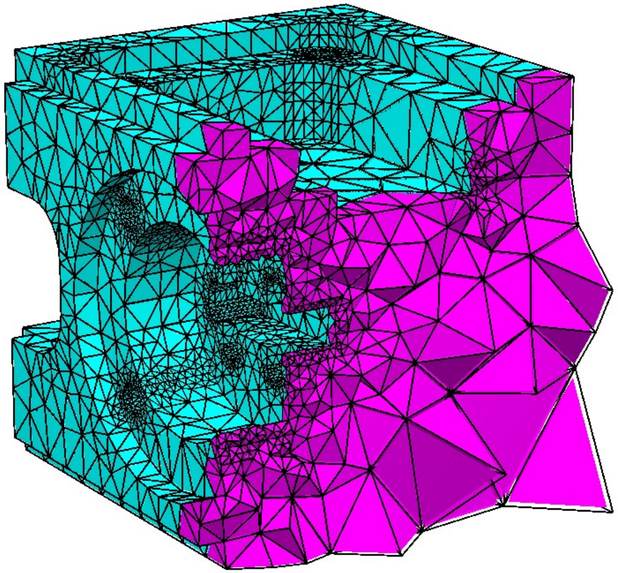
- Elastic objects are discretized in 2 ways:
  - Spring systems



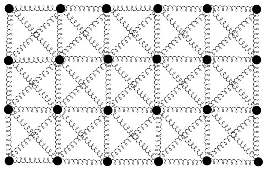
Elastic objects

- Elastic objects are discretized in 2 ways:
  - Finite Element Methods (FEM)

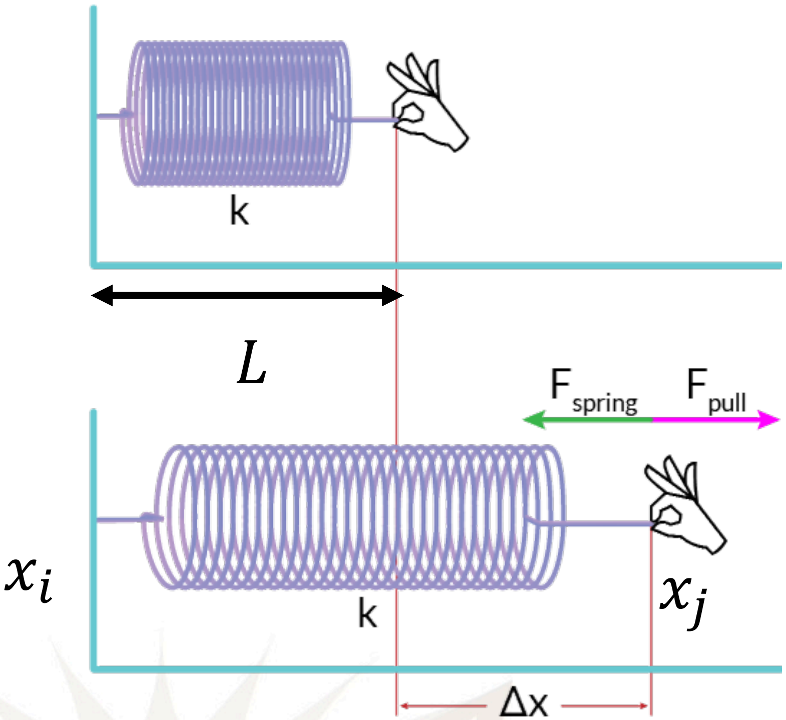




Springs



- 1D models embedded in 2D or 3D



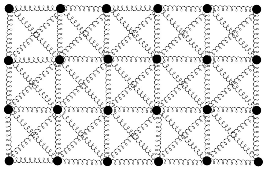
$$F^{spring} = F^{int} = -k \cdot \Delta x$$

(Hooke's law)

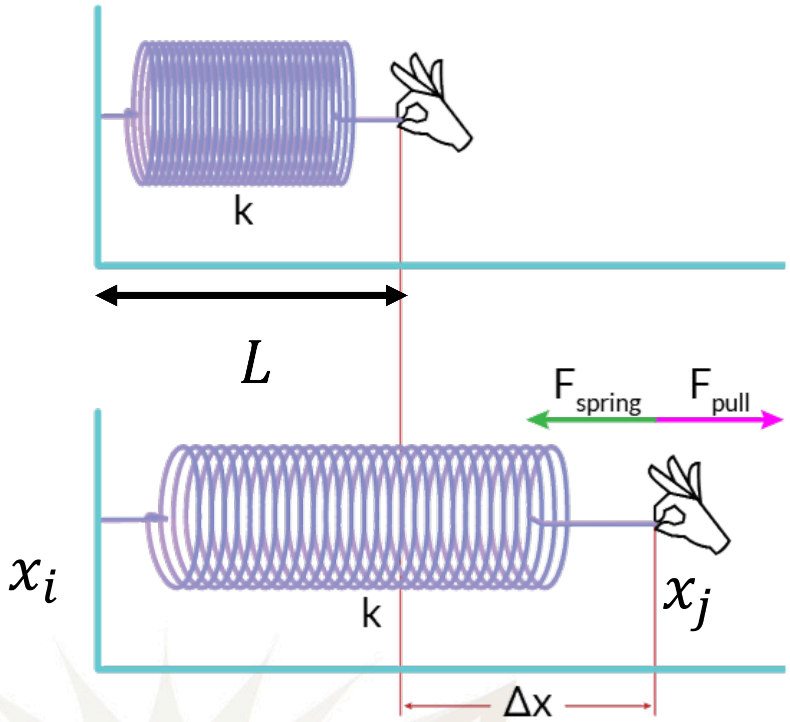
$$F^{int} = -k \cdot (\|x_i - x_j\| - L) \cdot \frac{x_j - x_i}{\|x_i - x_j\|}$$

https://sciencenotes.org/hookes-law-example-problem/

Springs

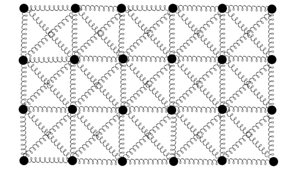


- Other forces?
- Infinite motion?
- Internal friction



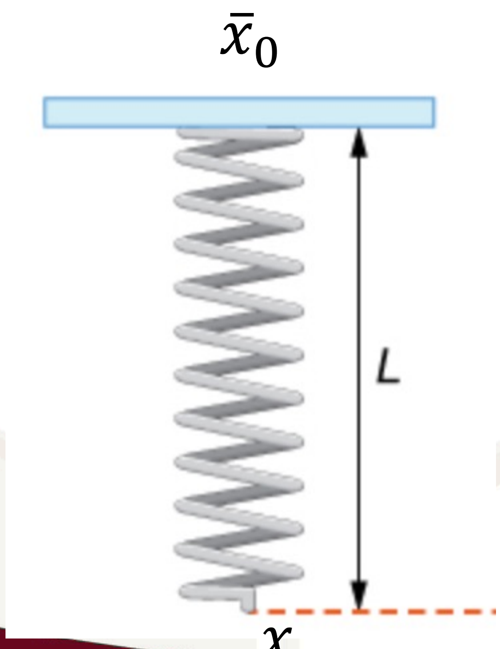
$$F^{damp} = F^{dp} = -\gamma \cdot v$$

https://sciencenotes.org/hookes-law-example-problem/



An example

- Implicit Euler: $y_{k+1} = y_k + f(t_{k+1}, y_{k+1})\Delta t$ (eq 12)
- $f(\mathrm{t}, y) = f\left(t, \begin{pmatrix} x(t) \\ v(t) \end{pmatrix}\right) = \begin{pmatrix} v(t) \\ M^{-1}F(t) \end{pmatrix}$ (eq 9)



$$y_0 = \begin{pmatrix} \bar{x}_0 + \begin{pmatrix} 0 \\ -L \end{pmatrix} \\ 0 \end{pmatrix},$$

$$F^{dp} = -\gamma \cdot v$$

$$F^{int} = -k \cdot (\|x - \bar{x}_0\| - L) \cdot \frac{x - \bar{x}_0}{\|x - \bar{x}_0\|}$$

An example

$$y_{k+1} = y_k + f(t_{k+1}, y_{k+1})\Delta t$$
$$f(\mathrm{t}, y) = f\left(t, \begin{pmatrix} x(t) \\ v(t) \end{pmatrix}\right) = \begin{pmatrix} v(t) \\ M^{-1}F(t) \end{pmatrix}$$

Implicit Euler Scheme

$$F_{damp} = F_{dp} = -\gamma \cdot v$$
$$F^{int} = -k \cdot (\|x - \bar{x}_0\| - L) \cdot \frac{x - \bar{x}_0}{\|x - \bar{x}_0\|}$$
$$G = -gM \text{ (gravity force)}$$

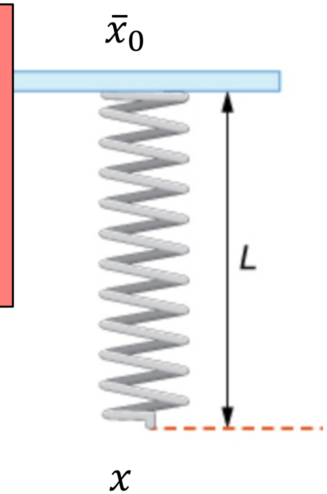
Forces

$$y_0 = \begin{pmatrix} \bar{x}_0 + \begin{pmatrix} 0 \\ -L \end{pmatrix} \\ 0 \end{pmatrix}$$

Initial values

$$\begin{pmatrix} x_{k+1} \\ v_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ v_k \end{pmatrix} + \begin{pmatrix} v_{k+1} \\ M^{-1}(F^{dp}_{k+1} + F^{int}_{k+1} - gM) \end{pmatrix} \Delta t =$$
$$= \begin{pmatrix} x_k \\ v_k \end{pmatrix} + \begin{pmatrix} v_{k+1} \\ M^{-1}\left(-\gamma \cdot v_{k+1} + -k \cdot (\|x_{k+1} - \bar{x}_0\| - L) \cdot \frac{x_{k+1} - \bar{x}_0}{\|x_{k+1} - \bar{x}_0\|} - gM\right) \end{pmatrix} \Delta t$$

Equation to solve



Non-linear equation

Newton-Raphson method (among others)

Practical Aspects

- Multiple connected Springs (i.e. spring system)
- Some end-points are fixed, some can move (i.e. variable)
- $x, v \in \mathbb{R}^{N \times k}$ where k is the dimension (i.e. 2 or 3) and N is the number of variables
- If a spring is between a variable and a fixed end-point, force are contributing to the motion of the variable end-point
- If a spring is between 2 variable –end-points, internal force force is split between the 2 end-points
- This logic makes the differentiation quite difficult