Animation for Computer Games
COMP 477/6311

Prof. Tiberiu Popa

Time integration
A lot of math

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)
- Just need to solve this equation… **easy peasy lemon squeezy**
- Not quite → but can appreciate that nearly everything in physics animation starts with one equation

The Basics

- $\ddot{x} = M^{-1}F$ (eq 1)
- x
  - variable that we solve for
  - function of time
- F
  - Assumed known → this is how we control the animation → will talk about types of forces in great detail
  - Also a function of time
- M
  - Generaly constant
  - Assumed known
  - Depends on the geometry

The Basics

- $\ddot{x} = M^{-1}F$ (eq 1)
- What kind of equation is this?
  - Differential equations
  - ODE vs. PDE
  - Our equation is an ODE
  - Second order
  - Initial value problem (IVP)
    - We know the position and velocity at the beginning

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

- $\ddot{x} = M^{-1}F$ (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?**

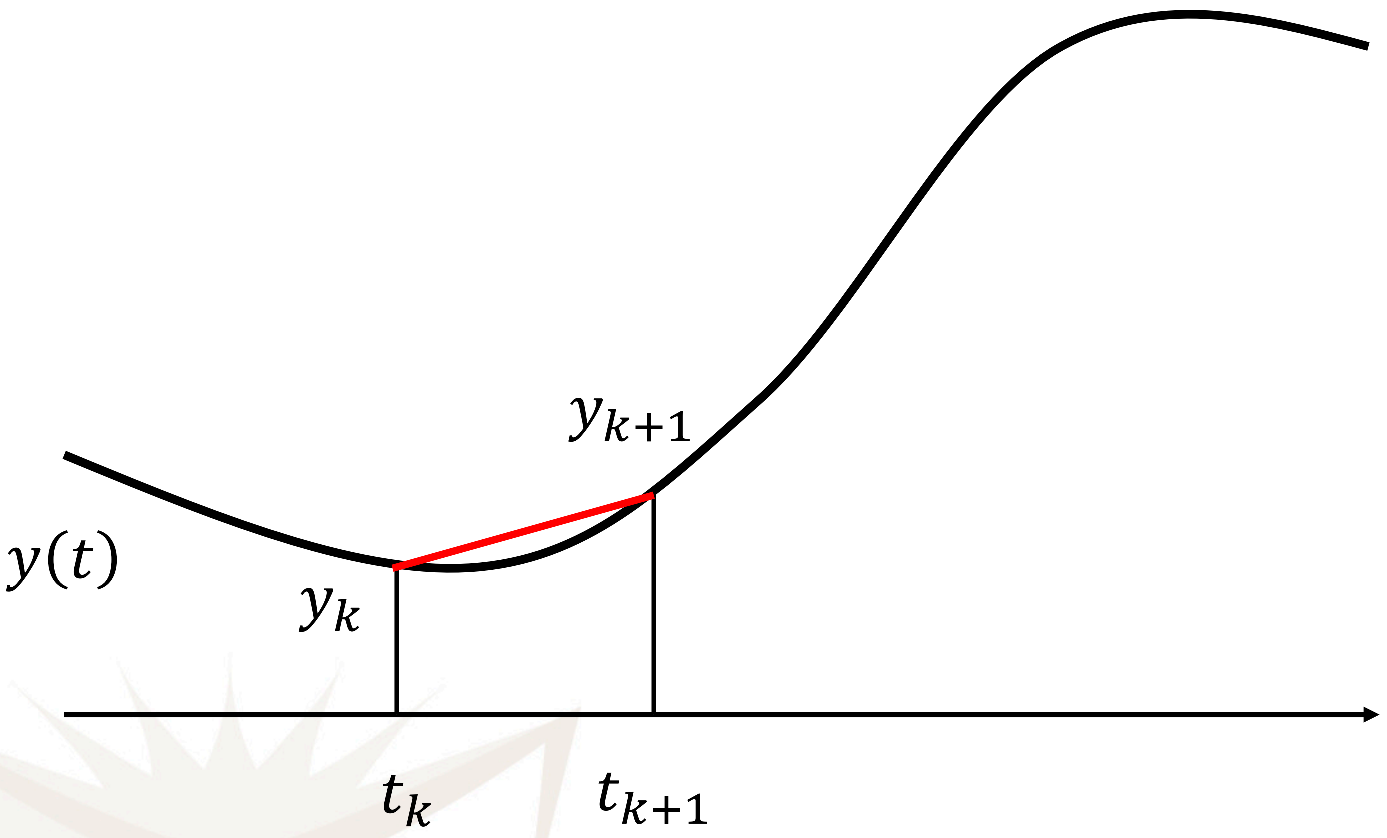
The Basics

- $y = \binom{x}{v}$
- $\ddot{x} = M^{-1}F$ (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



The Basics

- Taylor series:
- $y(t+\Delta t) = \sum_{i=0}^{\infty} \frac{y^{(i)}(t)(\Delta t^{i})}{i!} = y(t) + \dot{y}(t)\Delta t + \frac{\ddot{y}(t)(\Delta t^{2})}{2} + \cdots$
- $y_k = \binom{x_k}{v_k} \approx y(t_0 + k \cdot \Delta t)$

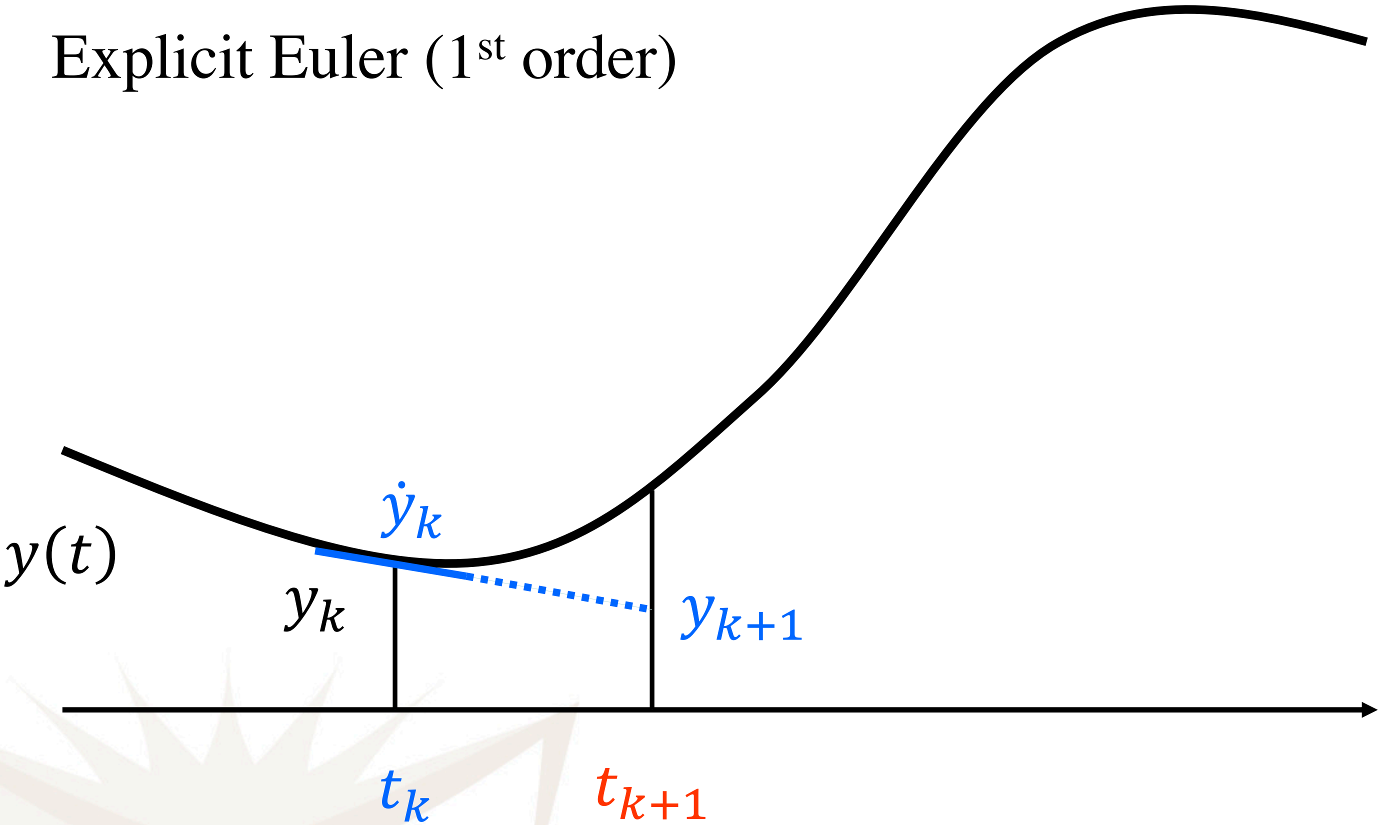
- Explicit schemes:
  - 1st order – truncate after the second term $\dot{y}$,
  - 2nd order – truncate after third term etc.

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 (1^{st} order)



Explicit Euler (1st order)

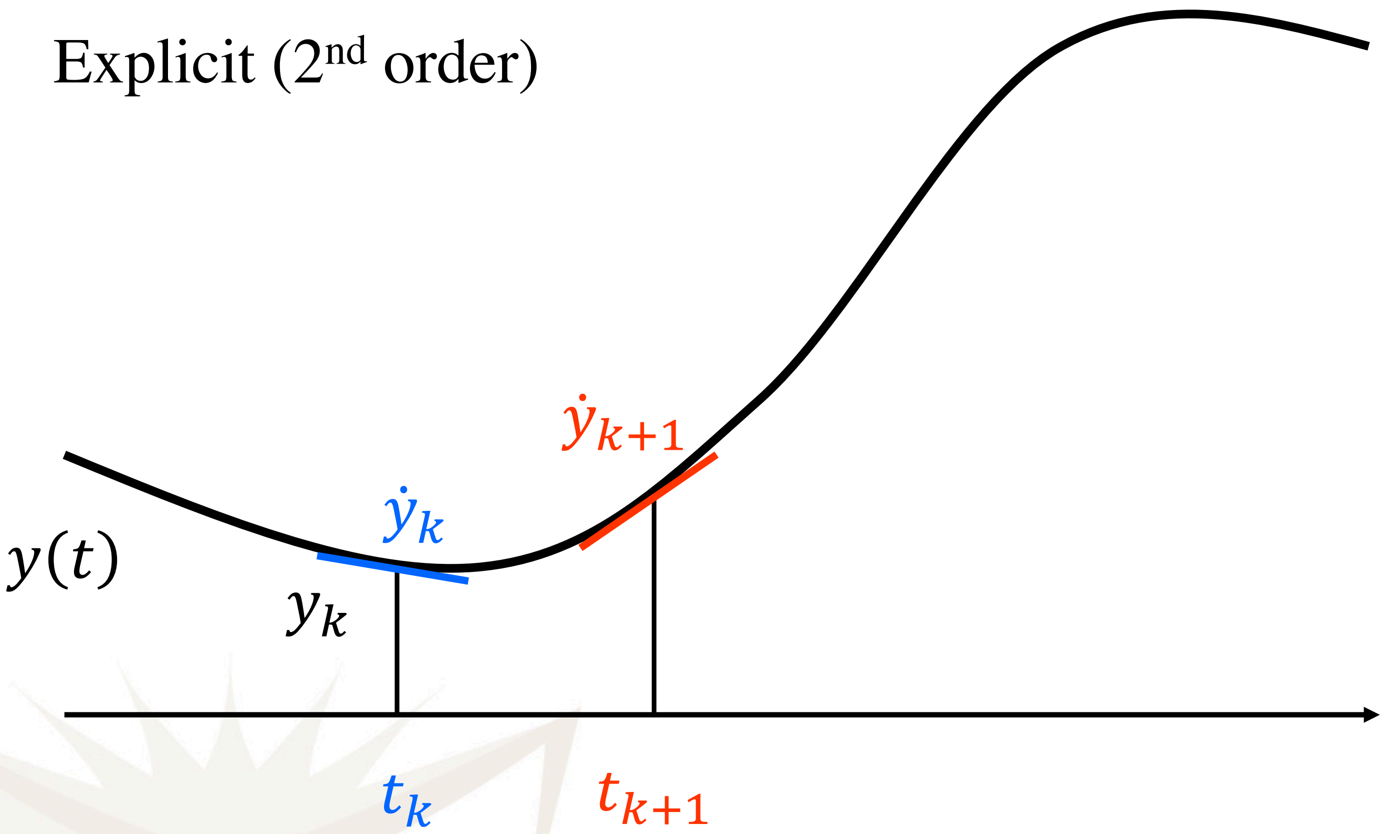
- $y_{k+1} = y_k + \dot{y}_k \Delta t$ (eq 3)
- Easiest scheme but
  - Error accumulates
  - Unstable unless tiny time steps

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
  - How to improve?
    - Higher order
      - Heun
      - Midpoint
      - Runge-Kutta
    - Implicit methods
      - Implicit/Backward Euler

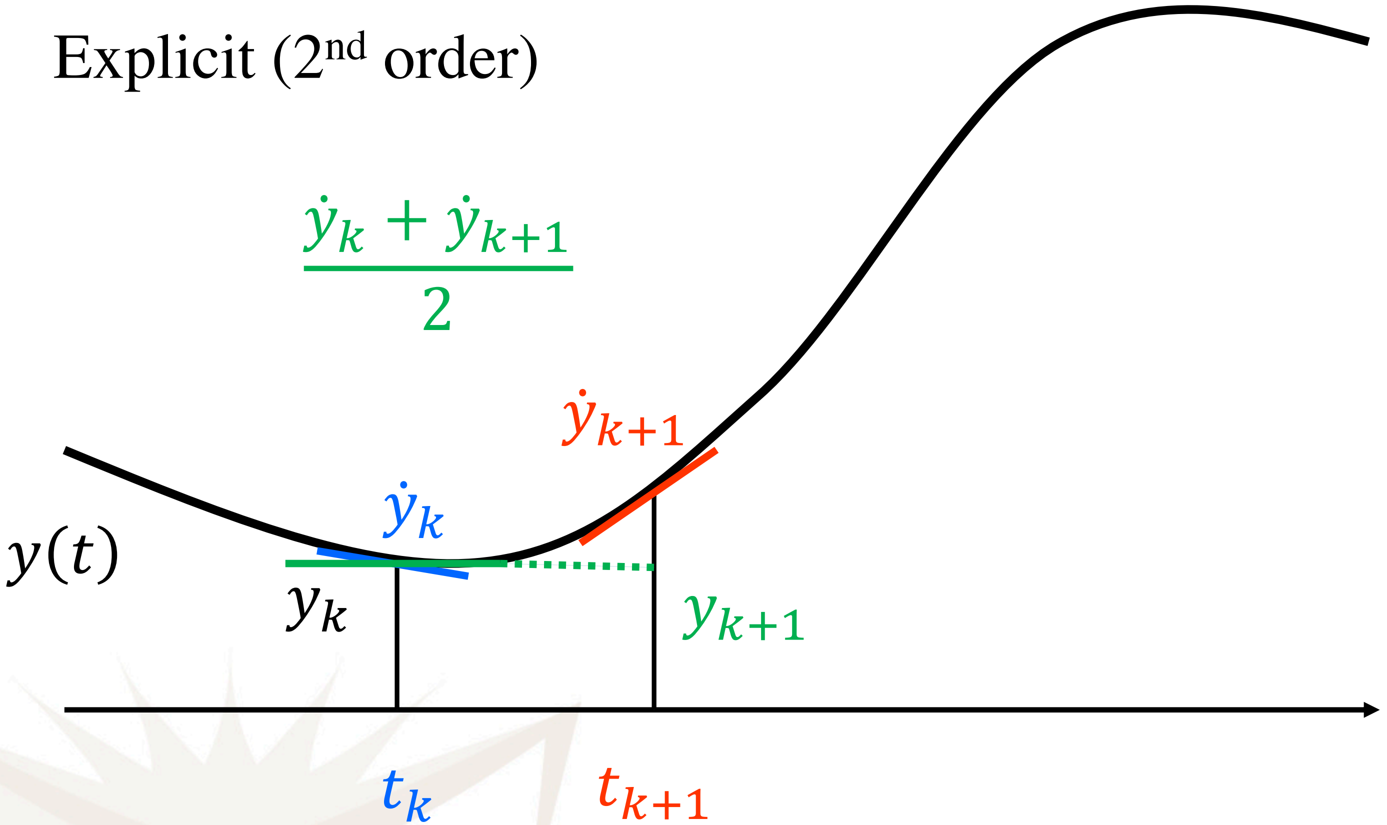
Intuition

Explicit (2nd order)



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 $\dot{y}_{k+1}$

Explicit (2nd order)

- $y_{k+1} = y_k + \frac{\dot{y}_{k+1}+\dot{y}_k}{2}\Delta t$ (eq 7)
- We don't know yet $\dot{y}_{k+1}$
- We can estimate using Explicit Euler → need to clean up notation
- $y = \binom{x}{v}, \dot{y} = \binom{v}{M^{-1}F}$ (eq 2) → rewrite
- Solve for y s.t.
- $\begin{cases} \dot{y}(t) = f(t, y) \\ \dot{y}(t_0) = y_0 \end{cases}$ (eq 8)

where $f(\mathrm{t}, y) = f\left(t, \binom{x(t)}{v(t)}\right) = \binom{v(t)}{M^{-1}F(t)}$ (eq 9)

Explicit (2nd order)

- $\begin{cases} \dot{y}(t) = f(t, y) \\ \dot{y}(t_0) = y_0 \end{cases}$ (eq 8)

where $f(\mathrm{t}, y) = f\left(t, \begin{pmatrix} x(t) \\ v(t) \end{pmatrix}\right) = \begin{pmatrix} v(t) \\ M^{-1}F(t) \end{pmatrix}$ (eq 9)

With this new notation, Explicit Euler becomes:

$y_{k+1} = y_k + f(t_k, y_k)\Delta t$ (eq 10)

Explicit (2nd order)

- $y_{k+1} = y_k + \frac{\dot{y}_{k+1} + \dot{y}_k}{2} \Delta t$ (eq 7)
- $\dot{y}(t) = f(t, y)$ by definition in eq 8
- $\hat{y}_{k+1} = y_k + \Delta t \cdot f(t_k, y_k)$ (eq 10) by applying explicit Euler
- Rewrite eq 7: $y_{k+1} = y_k + \frac{f(t_{k+1}, \hat{y}_{k+1}) + f(t_k, y_k)}{2} \Delta t$ (eq 11)
- Everything on the right-hand side is known once again
- $\binom{x_{k+1}}{v_{k+1}} = \binom{x_k}{v_k} + \frac{\Delta t}{2}\left(\begin{pmatrix} v_k + \Delta t \cdot M^{-1} \cdot F_k \\ M^{-1} \cdot \hat{F}_k \end{pmatrix} + \begin{pmatrix} v_k \\ M^{-1} \cdot F_k \end{pmatrix}\right)$

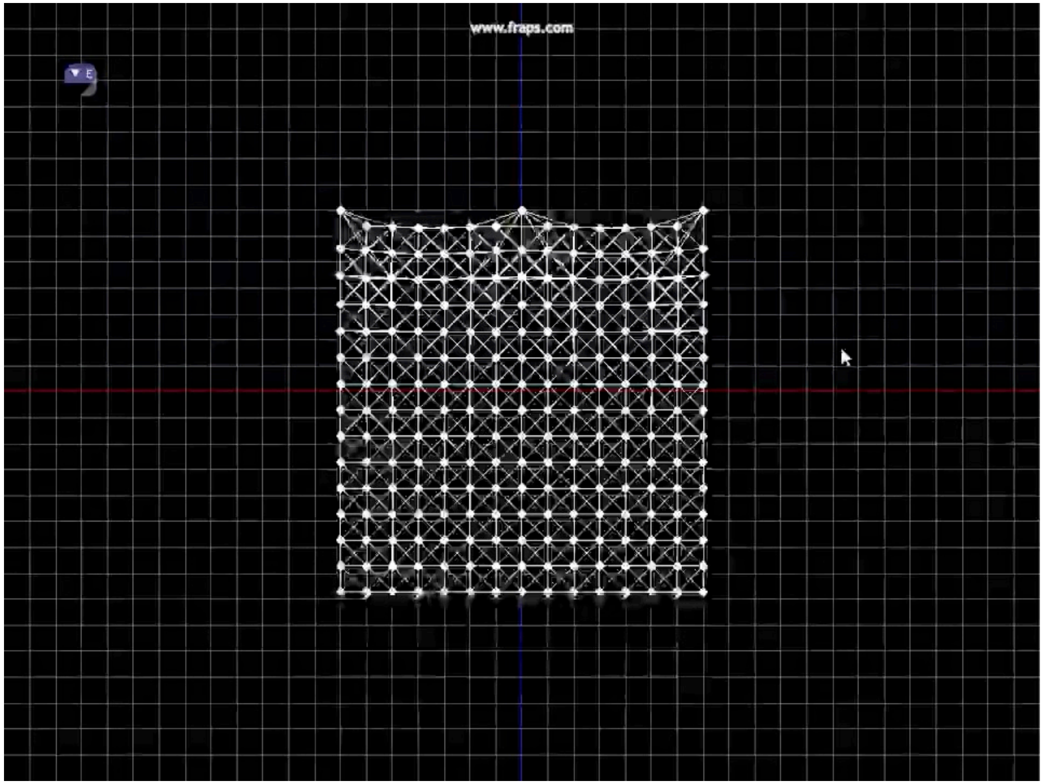
Heun's method

Explicit (2nd order)

- $y_{k+1} = y_k + \frac{\dot{y}_{k+1}+\dot{y}_k}{2}\Delta t$ (eq 7)
- $\dot{y}(t) = f(t,y)$ by definition in eq 8
- $\hat{y}_{k+1} = y_k + \frac{\Delta t}{2} \cdot f(t_k, y_k)$ (eq 10) by applying explicit Euler
- Rewrite eq 7: $y_{k+1} = y_k + \frac{f(t_{k+1},\hat{y}_{k+1})+f(t_k,y_k)}{2}\Delta t$ (eq 11)
- Everything on the right-hand side is known once again
- $\binom{x_{k+1}}{v_{k+1}} = \binom{x_k}{v_k} + \frac{\Delta t}{2}\left(\begin{pmatrix} v_k + \frac{\Delta t}{2} \cdot M^{-1} \cdot F_k \\ M^{-1} \cdot \hat{F}_k \end{pmatrix} + \begin{pmatrix} v_k \\ M^{-1} \cdot F_k \end{pmatrix}\right)$
- Mid-point method

Implicit Euler

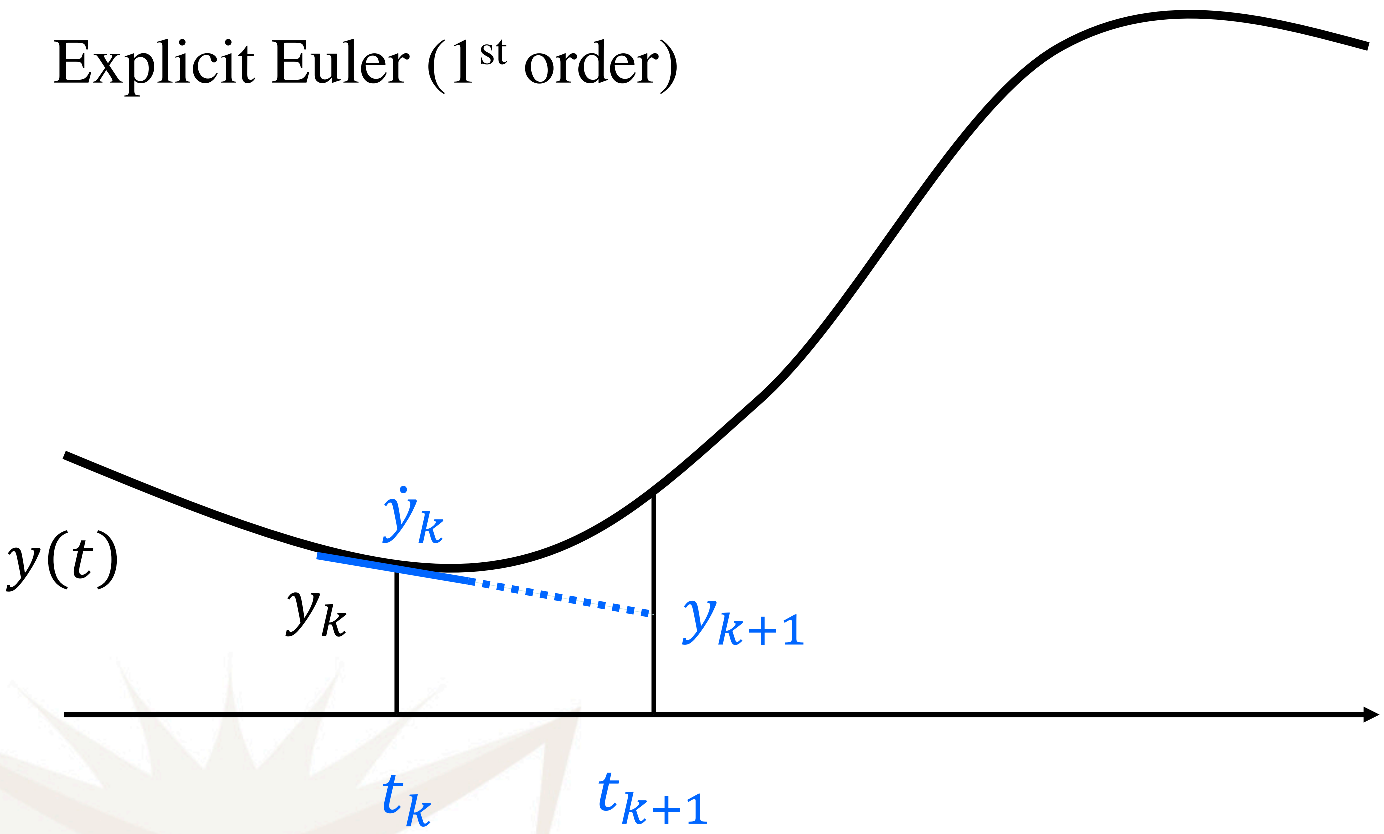
- Review explicit Euler: $y_{k+1} = y_k + \dot{y}_k \Delta t$ (eq 3)
- Easiest scheme but
  - Not accurate
  - Unstable



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

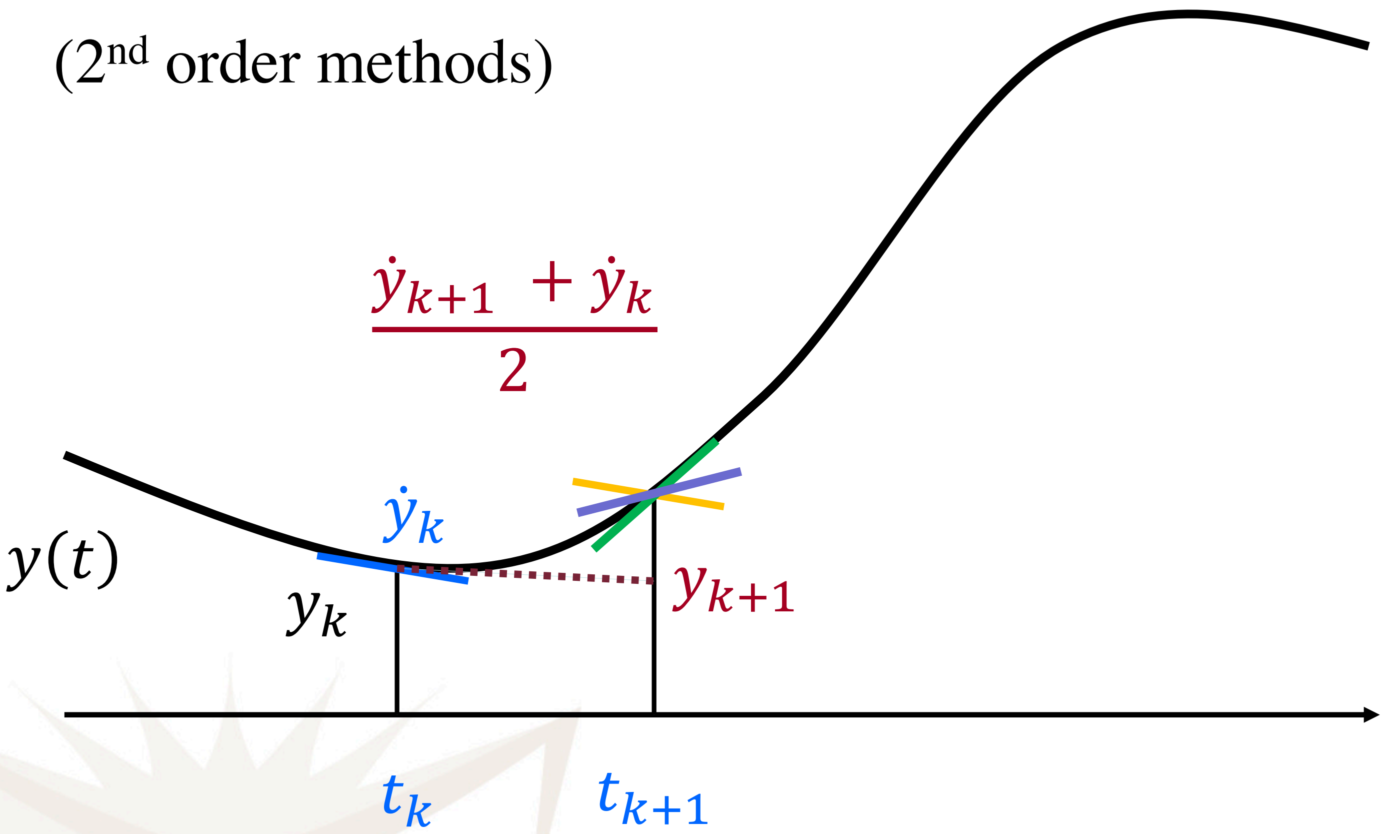
Intuition

Explicit Euler (1st order)



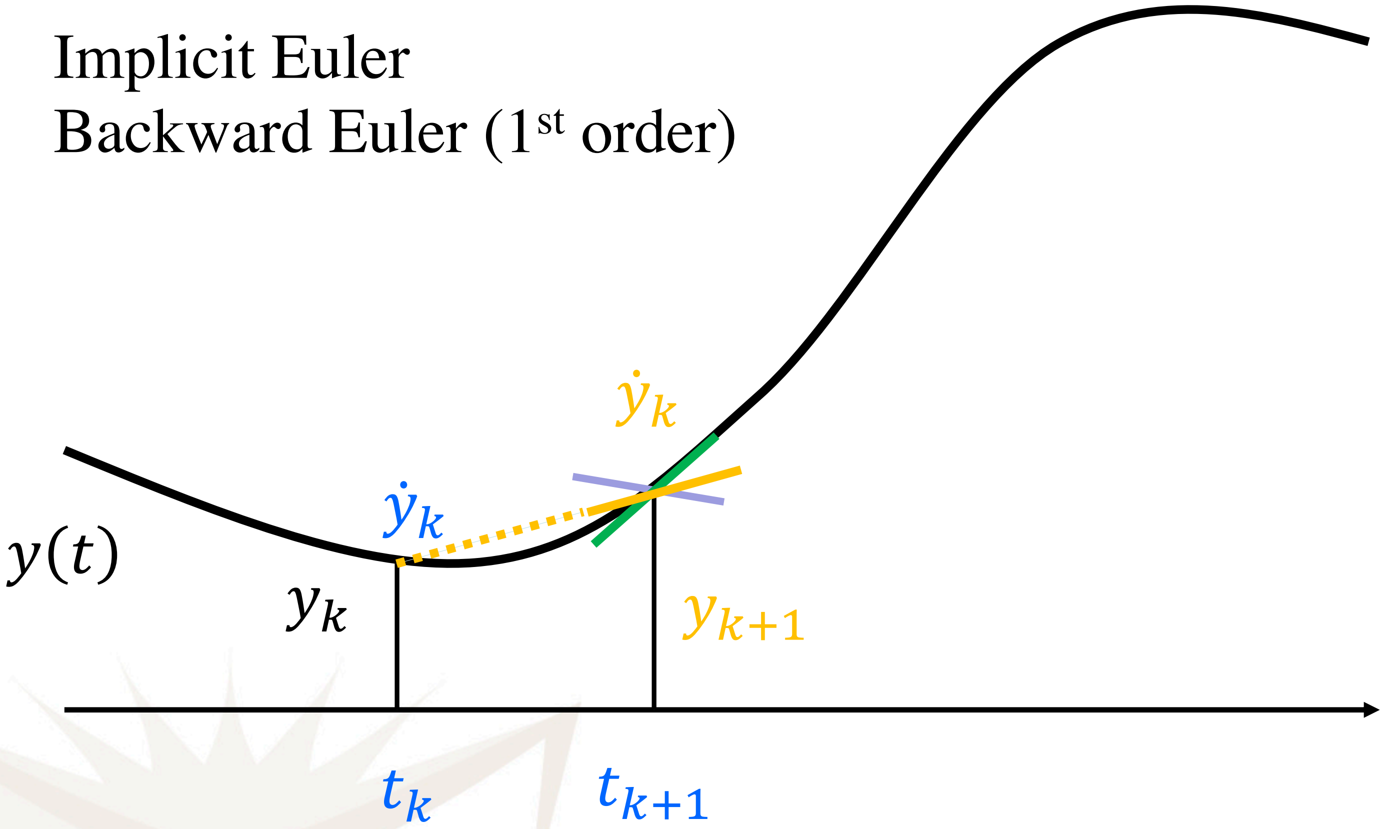
Intuition

(2nd order methods)



Intuition

Implicit Euler
Backward Euler (1st order)

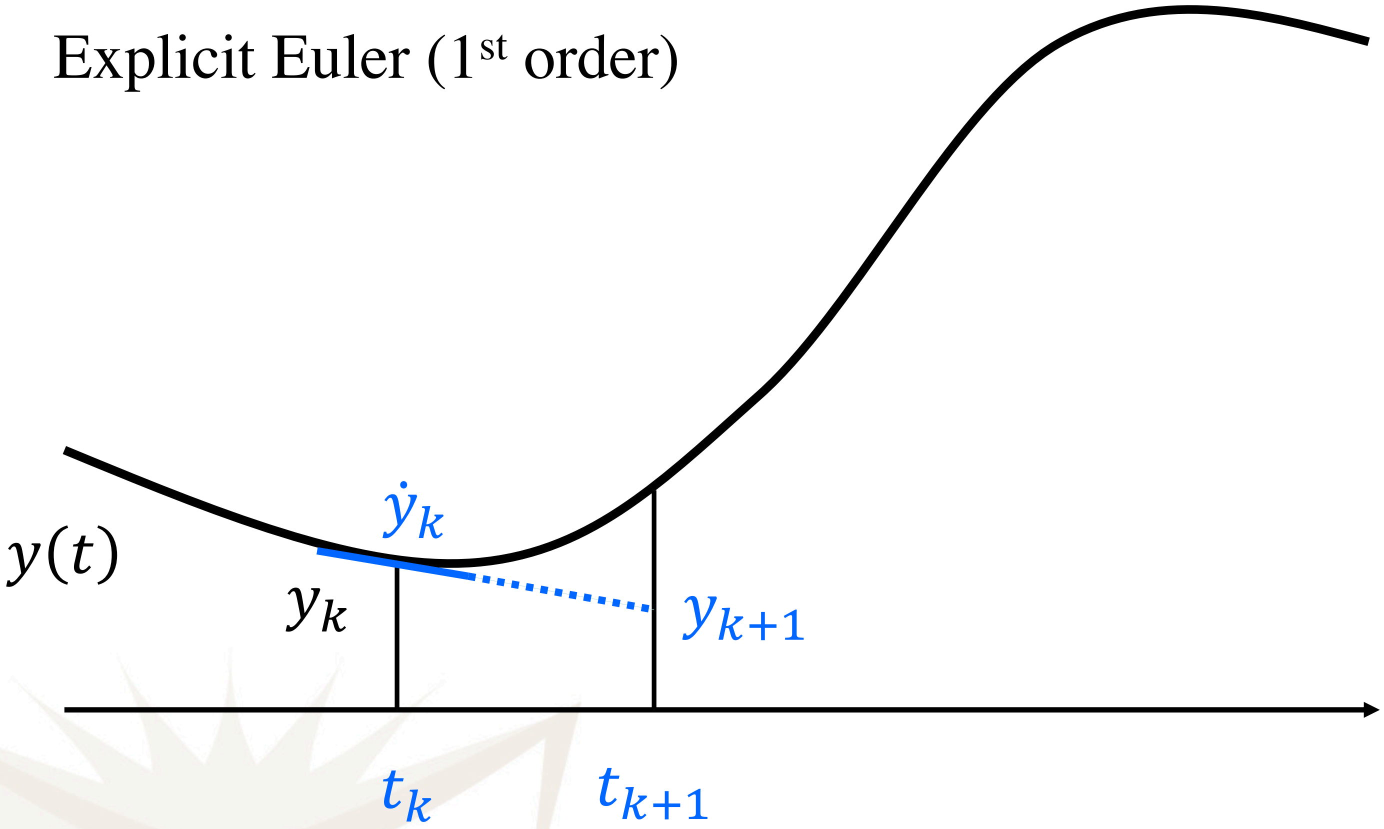


Implicit Euler (1st order)

- Explicit Euler: $y_{k+1} = y_k + f(t_k, y_k)\Delta t$ (eq 10)
- Implicit Euler: $y_{k+1} = y_k + f(t_{k+1}, y_{k+1})\Delta t$ (eq 12)
- Also called Backward Euler
- Small change → unconditionally stable
- Why?

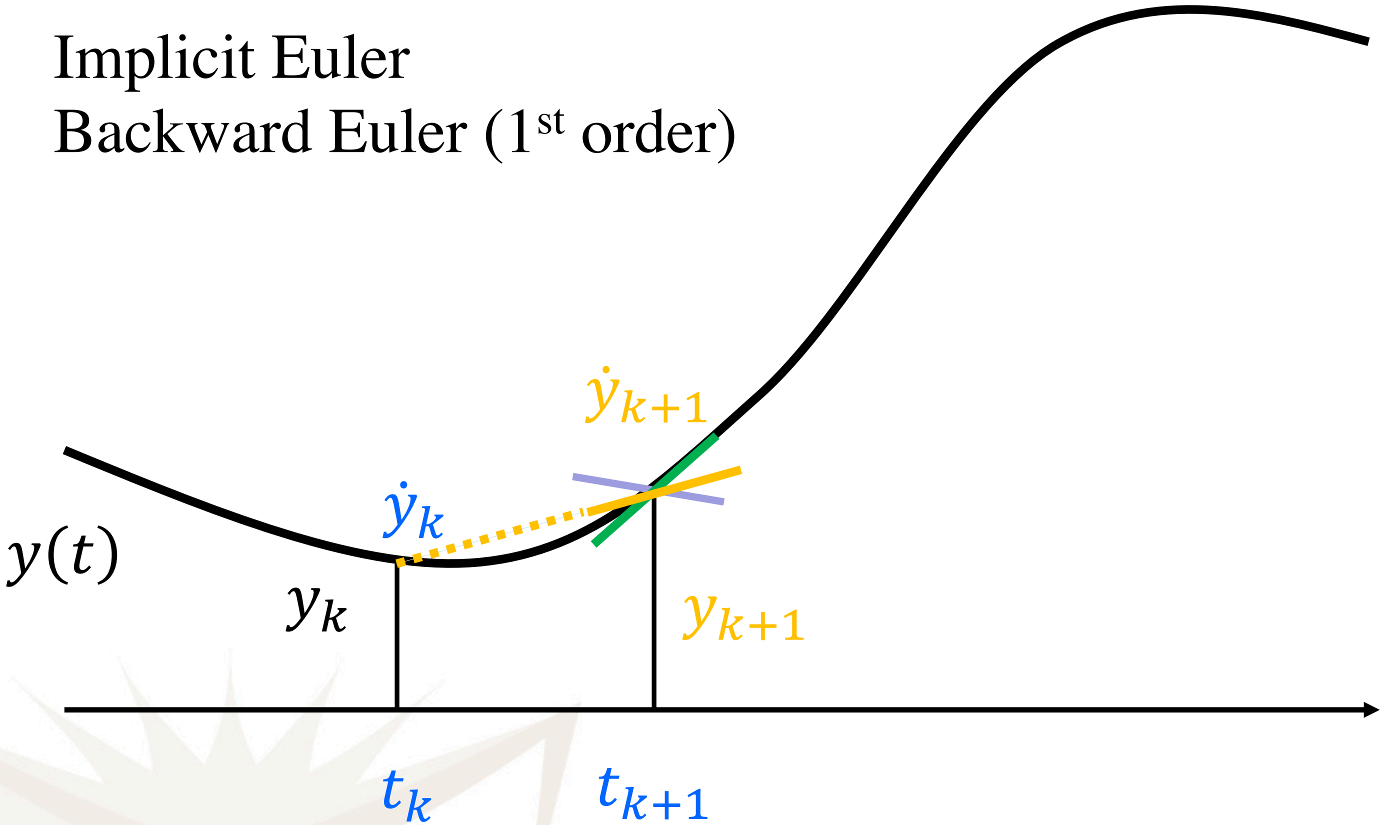
Intuition

Explicit Euler (1st order)



Intuition

Implicit Euler
Backward Euler (1st order)



Implicit Euler (1st order)

- Implicit Euler: $y_{k+1} = y_k + f(t_{k+1}, y_{k+1})\Delta t$ (eq 12)
- No free lunch
- RHS unknown
- The idea is to not estimate it from current state
- Rather to add it as a variable in the system
- Solve for y_{k+1}
- Typically a non-linear system
- Newton-Raphson method to solve (will be shown in the lab)

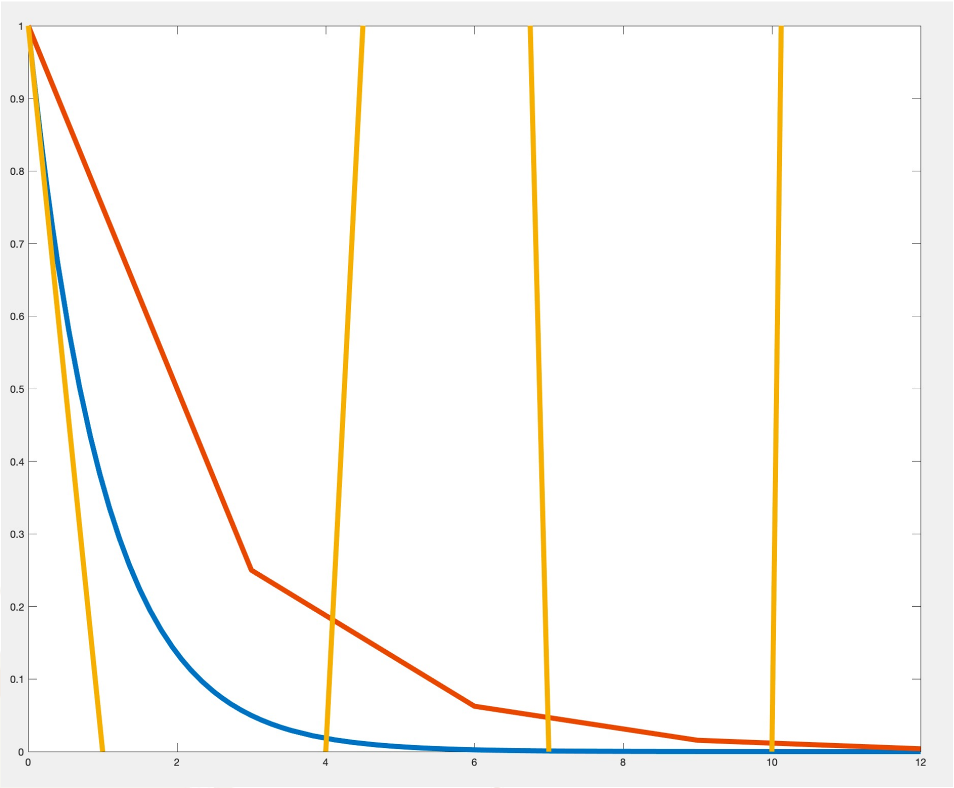
Implicit Euler (1st order)

- 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)

- $\begin{pmatrix} x_{k+1} \\ v_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ v_k \end{pmatrix} + \begin{pmatrix} v_{k+1} \\ M^{-1}F_{k+1} \end{pmatrix}\Delta t$ (eq 13)
- Looks easier than it is: forces can be non-linear in the variables

Time integration example

- On the whiteboard



Intuition