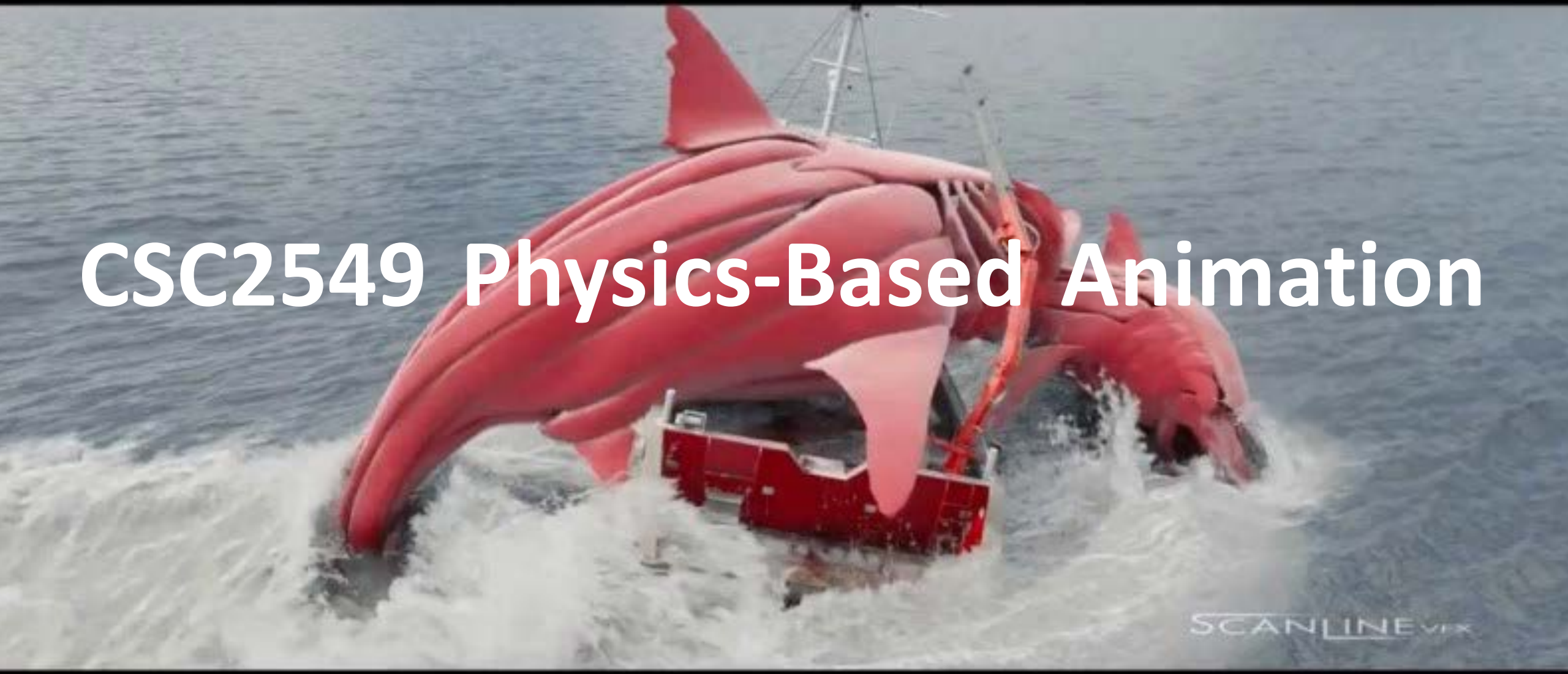


CSC2549 Physics-Based Animation

The Meg | Scanline VFX

Last Week: Introduction and Springs

Today: Time Integration

Reminders

Website:

https://github.com/dilevin/CSC2549-physics-based-animation

Bulletin Board:

https://bb-2019-09.teach.cs.toronto.edu/c/csc2549

MarkUs:

https://markus.teach.cs.toronto.edu/csc2549-2019-09/

Contact:

diwlevin@cs.toronto.edu

More Reminders

Assignment #1 is due next Friday

https://github.com/dilevin/CSC2549-a1-mass-spring-1d

Assignment #2 will be posted tomorrow

Graphics Reading Group

Seminar Room in BA5166 (Dynamic Graphics Project)

Wednesdays 11am

Today

1. Questions about the last lecture
2. Introduction to Time Integration
3. Algorithms
    1. Forward Euler Time Integration
    2. Runge-Kutta Time Integration
    3. Backward (Implicit) Euler Time Integration
    4. Symplectic Euler Integration
4. Some C++ Review

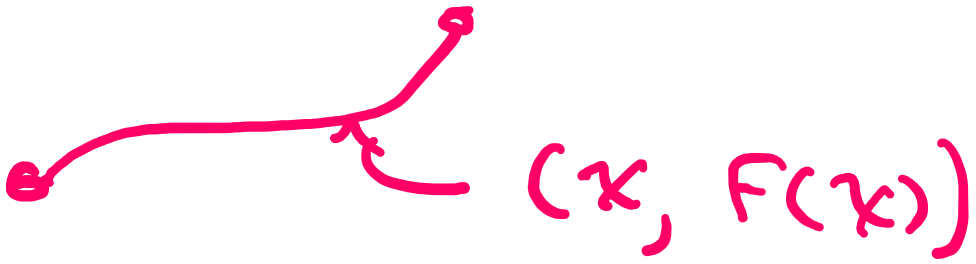
Any Questions about the Last Lecture ?

Fun Math Question

Use the calculus of variations to show that the shortest distance between two points is a straight line

$$d\ell = \sqrt{1+\left(\frac{\partial F}{\partial x}\right)^2}\,dx$$

arc length



Let our 2D curve be $(x, F(x))$ from x_0 to x_1

Curve length is $\ell = \int_{x_0}^{x_1} \sqrt{1 + F'(x)^2}\, dx$

where $F'(x) = \frac{dF}{dx}$

$$\delta\ell = \int_{x_0}^{x_1} \frac{d}{dF'(x)} \underbrace{\left(\sqrt{1 + F'(x)^2}\right)}_{L} \delta F' = \int_{x_0}^{x_1} \frac{F'(x)}{\sqrt{1+F'(x)^2}} \delta F' dx = 0$$

From Integration by Parts

$$\frac{d}{dx}\left(\frac{F'(x)}{\sqrt{1 + F'(x)^2}}\right) = 0$$

$$\frac{F'(x)}{\sqrt{1+F'(x)^2}} = a$$

Square both sides $\quad F'(x)^2 = \frac{a^2}{1-a^2}, \quad 0 \le a^2 \le 1$

Implies $\quad F'(x) = \underbrace{m}_{\text{some constant}}$

So $\quad F(x) = mx + b \quad$ which is a line.

Time Integration

Input: Ordinary Differential Equation: $\ddot{\mathbf{q}} = \mathbf{f}\left(\mathbf{q}, \dot{\mathbf{q}}\right)$

Output: Discrete Update Equation

$$\mathbf{q}^{t+1} = \mathbf{f}\left(\mathbf{q}^{t}, \mathbf{q}^{t+1}, \ldots, \dot{\mathbf{q}}^{t}, \dot{\mathbf{q}}^{t+1}, \ldots\right)$$

The Coupled First Order System

$$m\ddot{x} = -kx$$

$$\dot{x} = v \quad \text{velocity}$$

$$m\dot{v} = -kx$$

$$\dot{x} = v$$

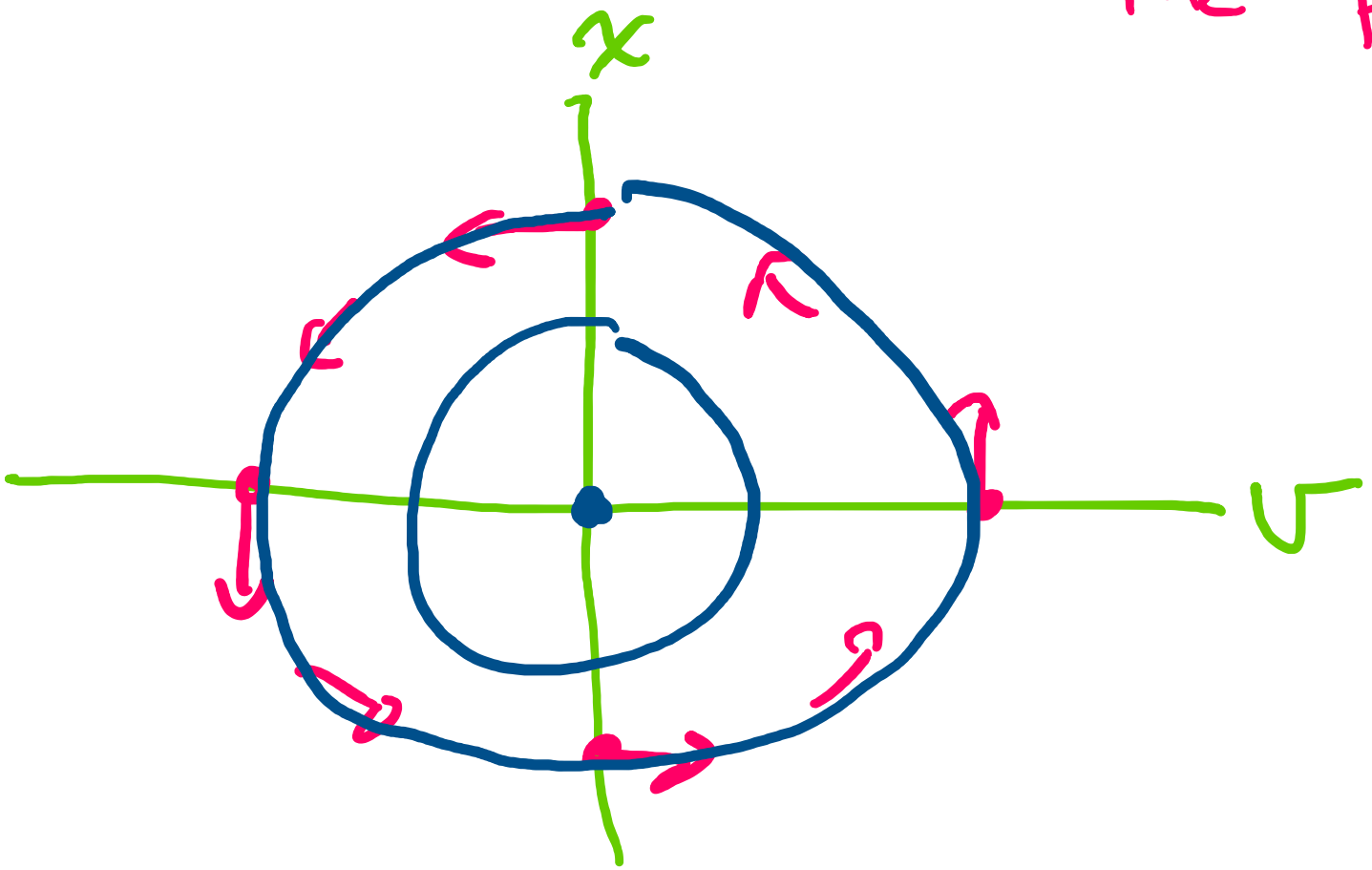
$$A\dot{y} = By$$

$$\dot{y} = A^{-1}By \quad C$$

$$\begin{bmatrix} m & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{v} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} 0 & -k \\ 1 & 0 \end{bmatrix} \begin{bmatrix} v \\ x \end{bmatrix}$$

An Illustrative Example

The phase space



Types of Time Integration

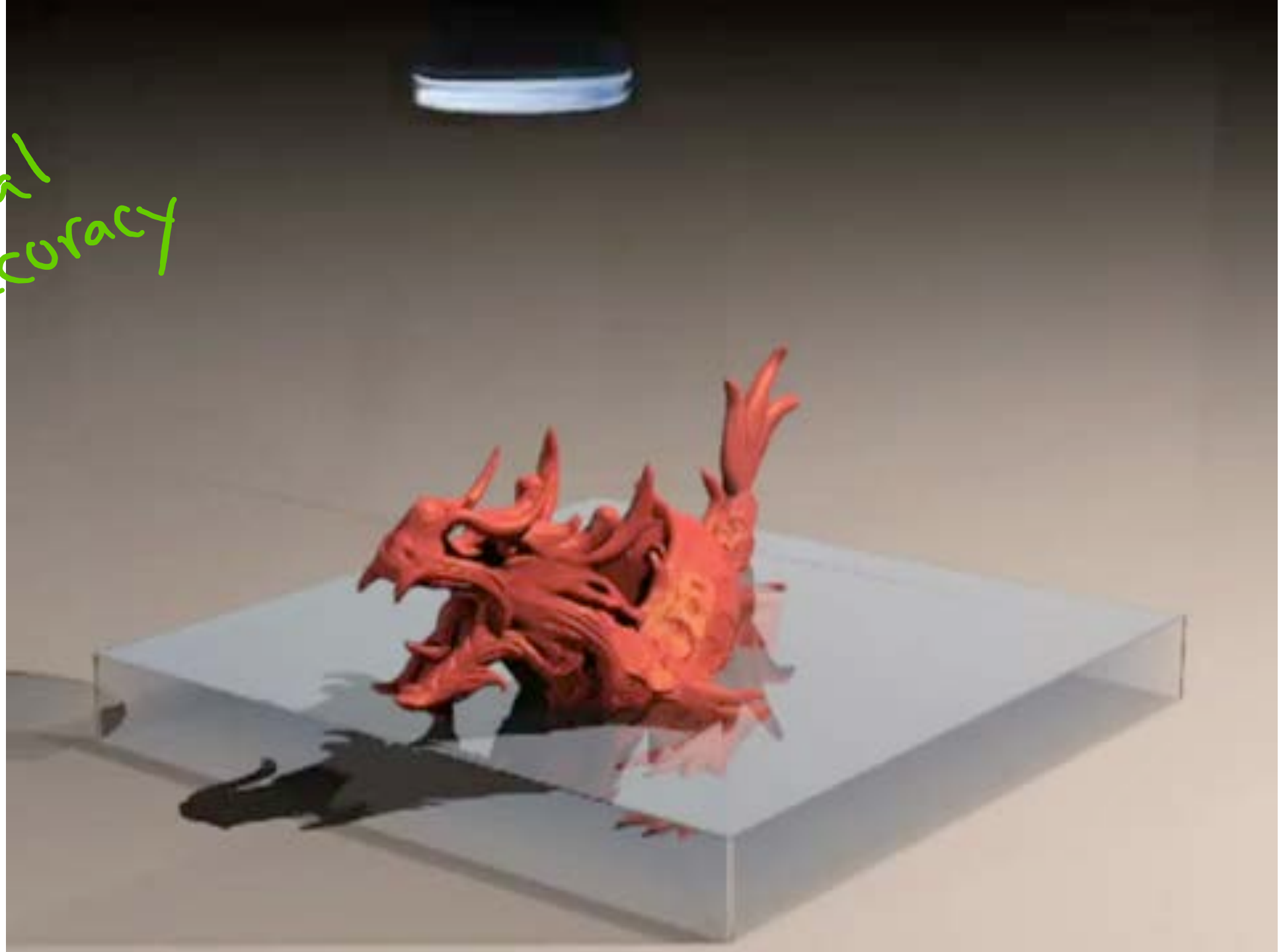
Explicit: Next time step can be computed entirely using values from the current time step

Implicit: Next time step is computed using values from the future!

Concerns

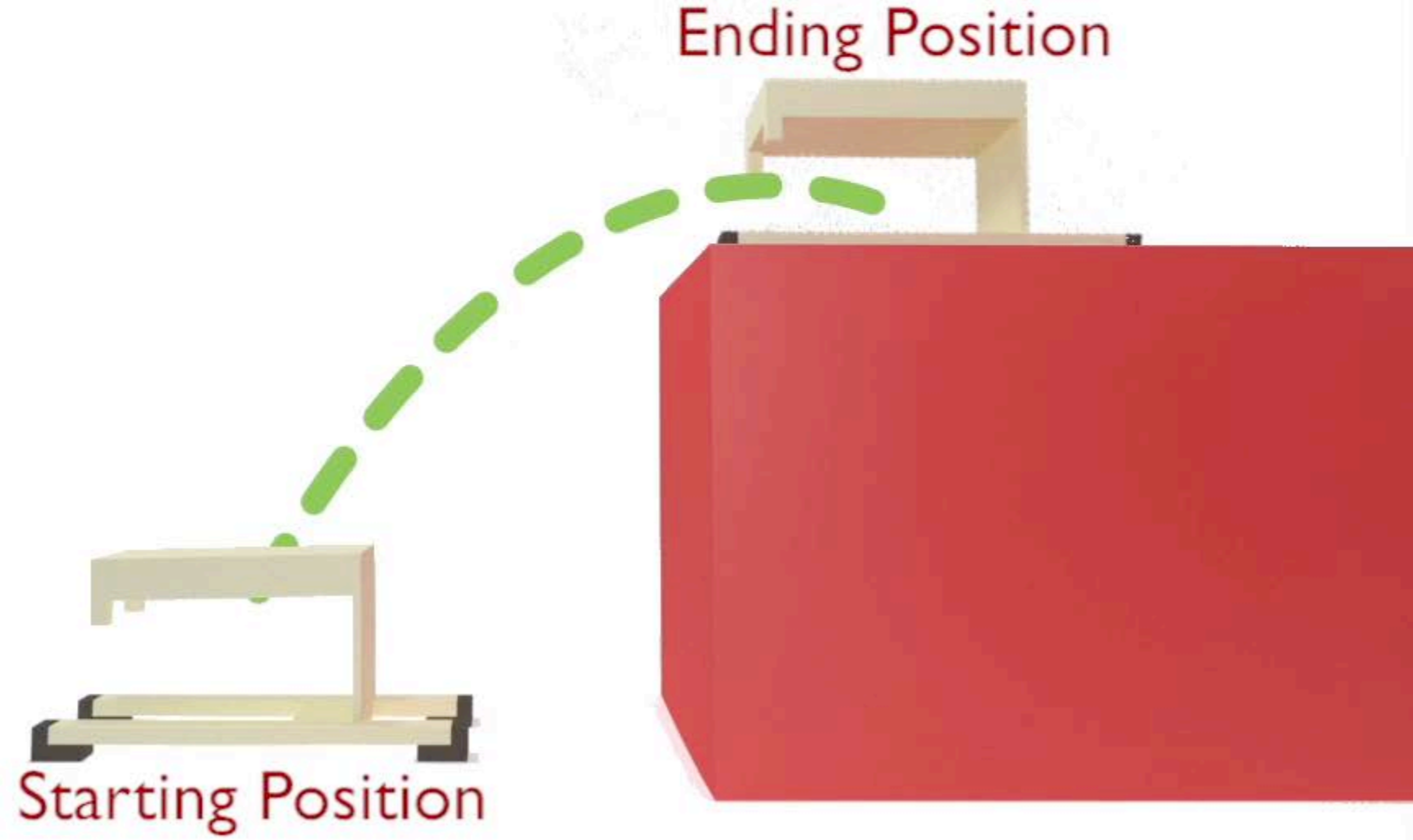
1. Stability - Bounded energy behaviour
2. Accuracy - Visual accuracy or experimental accuracy
3. Performance - fast to evaluate

Visual Accuracy

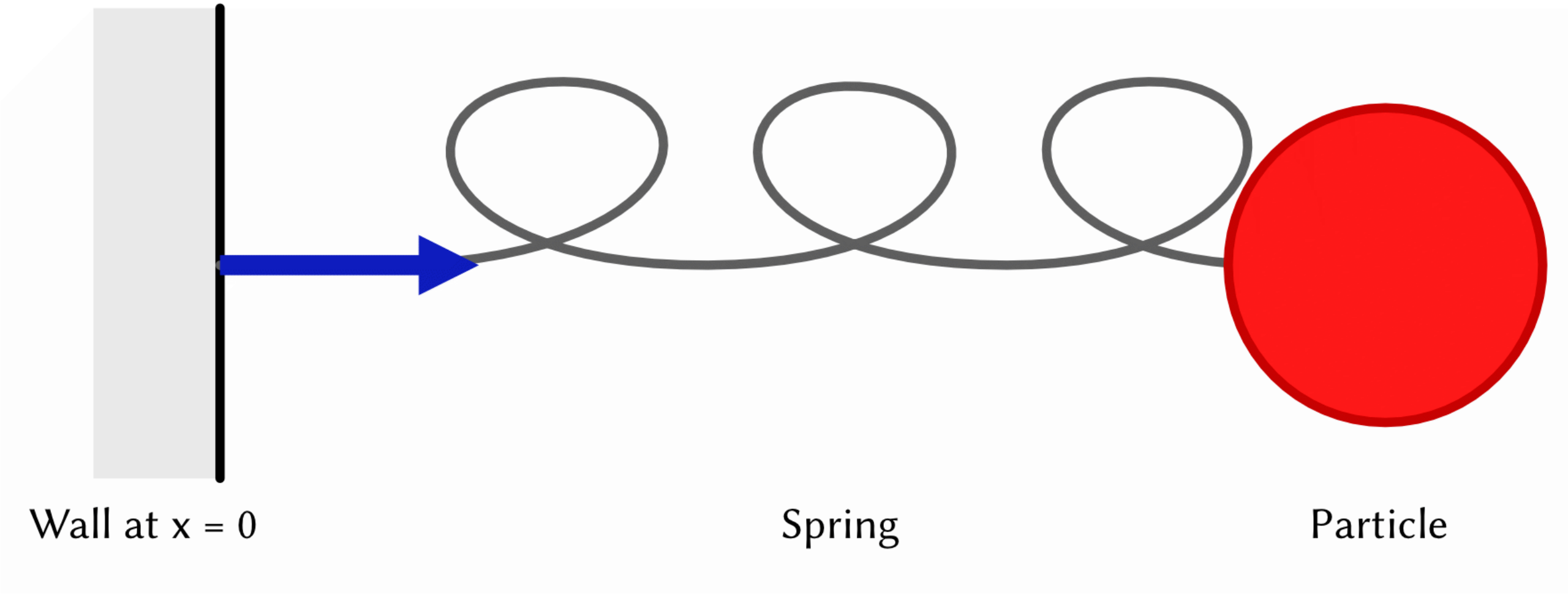


Results

Experimental Accuracy



For Assignment 1



Forward-Euler Time Discretization

$A\dot{y} = By$ ← evaluate RHS at time t

$$\dot{y} = \frac{1}{\Delta t}\left(y^{t+1} - y^t\right)$$

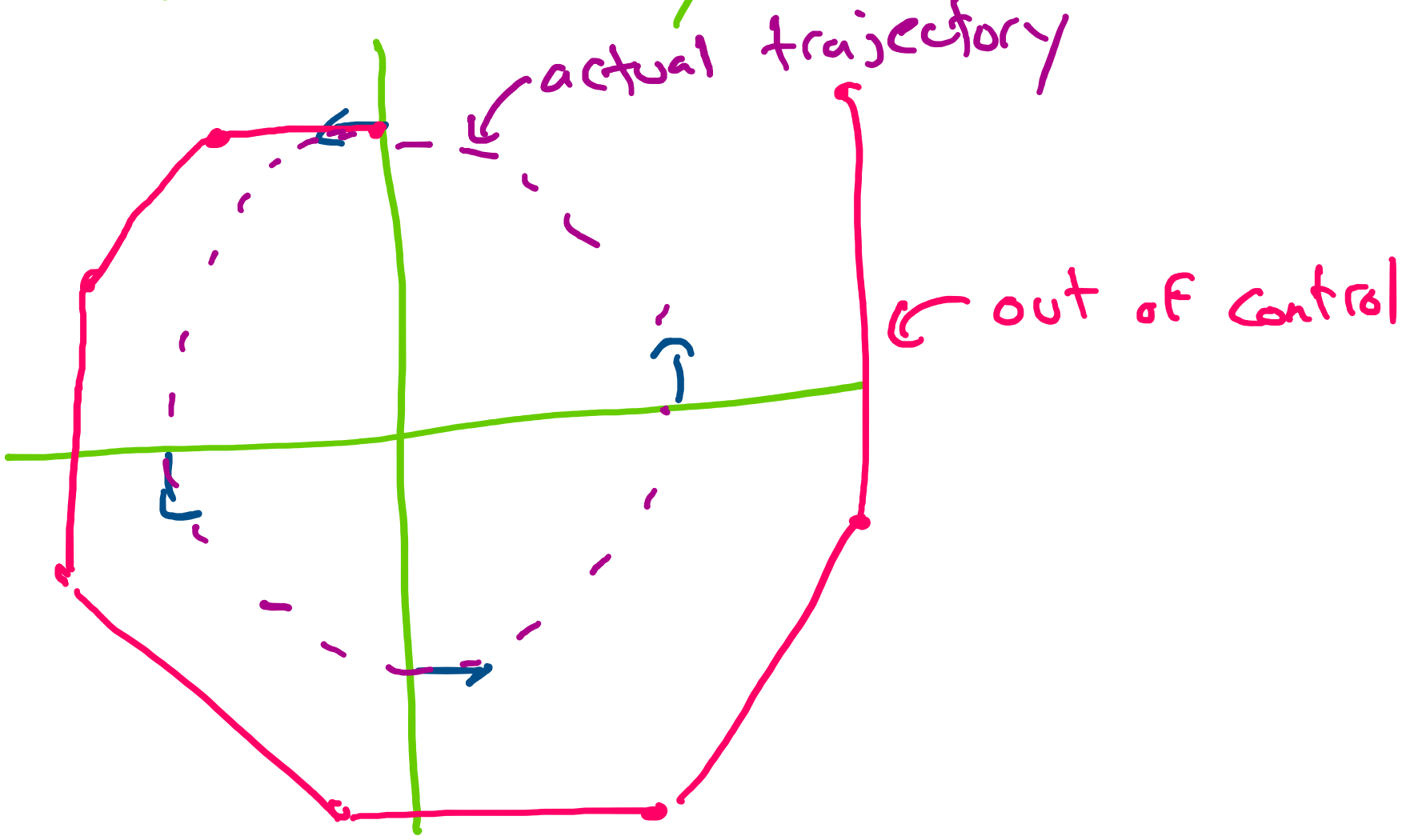
$$A y^{t+1} = A y^t + \Delta t\, B y^t$$

For our mass-spring system

$$v^{t+1} = v^t - \Delta t \frac{k}{m} x^t$$

$$x^{t+1} = x^t + \Delta t\, v^t$$

Phase Space Trajectory



Runge-Kutta

Explicit, multi-step method

Use multiple function evaluations to get a more accurate time step

Forward Euler

$$y^{t+1} = y^t + \frac{\Delta t}{2}\left(C y^t + C\left(y^t + \Delta t\, C y^t\right)\right)$$

Forward Euler

This is "2nd order" accurate

For A1 implement Runge-Kutta 4

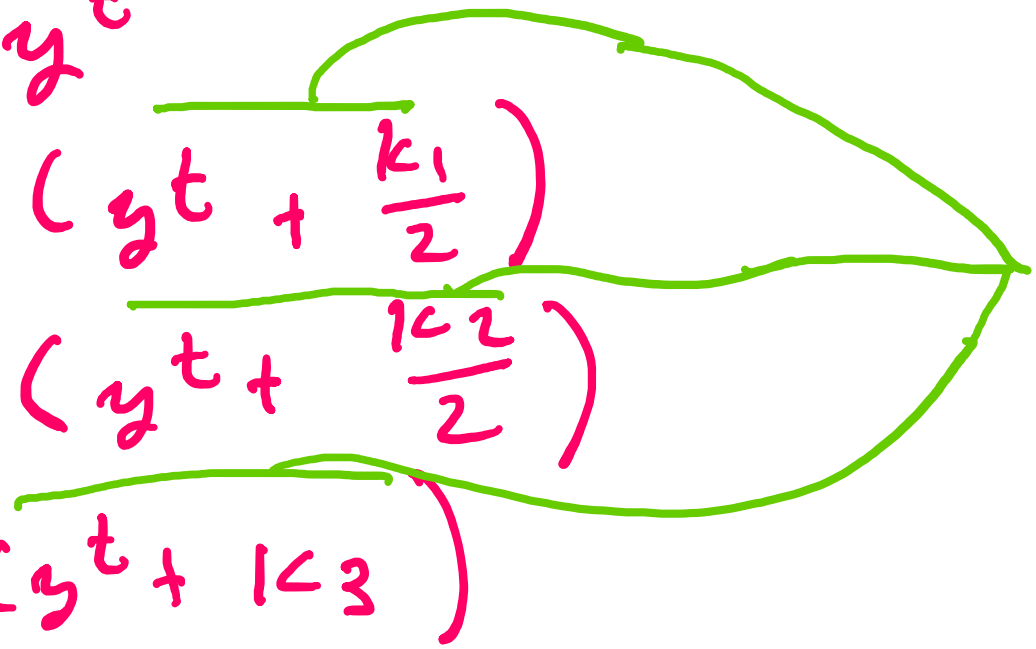
$$y^{t+1} = y^t + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$k_1 = \Delta t \; C y^t$$

$$k_2 = \Delta t \; C\left(y^t + \frac{k_1}{2}\right)$$

$$k_3 = \Delta t \; C\left(y^t + \frac{k_2}{2}\right)$$

$$k_4 = \Delta t \; C\left(y^t + k_3\right)$$



Forward Euler

Backward (Implicit) Euler

$$A\dot{y} = By$$

$$A\left(\frac{y^{t+1} - y^t}{\Delta t}\right) = By^{t+1}$$

Important

$$m(v^{t+1} - v^t) = -\Delta t K x^{t+1}$$

$$x^{t+1} - x^t = \Delta t v^{t+1}$$

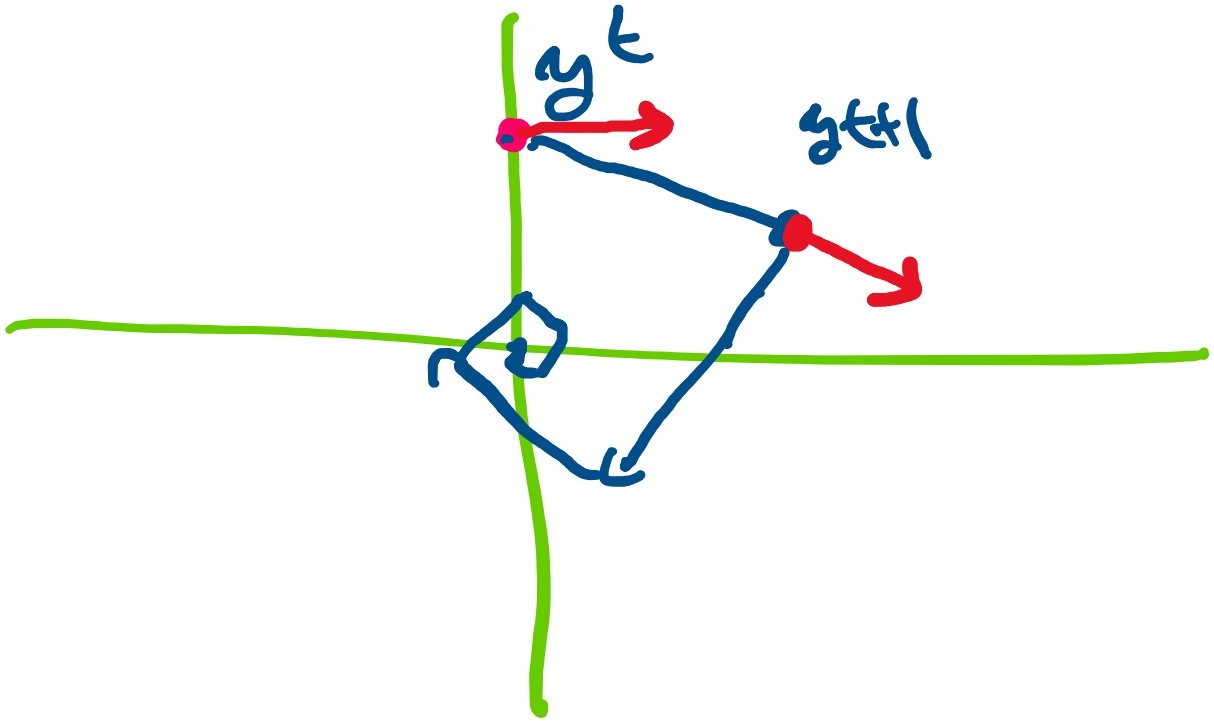
$$v^{t+1} = v^t - \Delta t \frac{k}{m} x^{t+1}$$

$$x^{t+1} = x^t + \Delta t\, v^{t+1}$$

$$\underline{\underline{v}}^{t+1} = \underline{\underline{v}}^t - \Delta t \frac{k}{m} \left(\underline{\underline{x}}^t + \Delta t\, \underline{\underline{v}}^{t+1} \right)$$

$$\left(1 + \Delta t^2 \frac{k}{m}\right) v^{t+1} = v^t - \Delta t \frac{k}{m} x^t$$

Stability Analysis



Symplectic Euler

Symplectic Euler comes from a discrete variational principle

Last class we wrote down the continuous version of the Principle of Least Action, derived the E-L Equations and now we are discretizing these equations

What if we discretize the variational principle ?

Discrete Variational Principle

Symplectic Euler

$$v^{t+1} = v^t - \Delta t \frac{k}{m} x^t$$ ← explicit

⊕

$$x^{t+1} = x^t + \Delta t\, v^{t+1}$$ ← implicit

bounded energy

Can get arbitrarily "bad" but won't explode.

```
coordiantes for the mass-spring system
ed velocity for the mass spring system
ep in seconds
```

wrong, actually Force(Fext, q, qdot)

```
 a function that computes the force acting on the mass as a function. This takes q and qdot as parameters.
t) - a function that computes the stiffness (negative second derivative of the potential energy). This takes q and qd
```

same as Force

```
 updated generalized coordinate using Backward Euler time integration
to the updated generalized velocity using Backward Euler time integration

RCE, typename STIFFNESS>
_euler(Eigen::VectorXd &q, Eigen::VectorXd &qdot, double dt, double mass,  FORCE &force, STIFFNESS &stiffness) {
```

return by reference

Next Week:

3D Mass-Spring Systems

Final Project Info