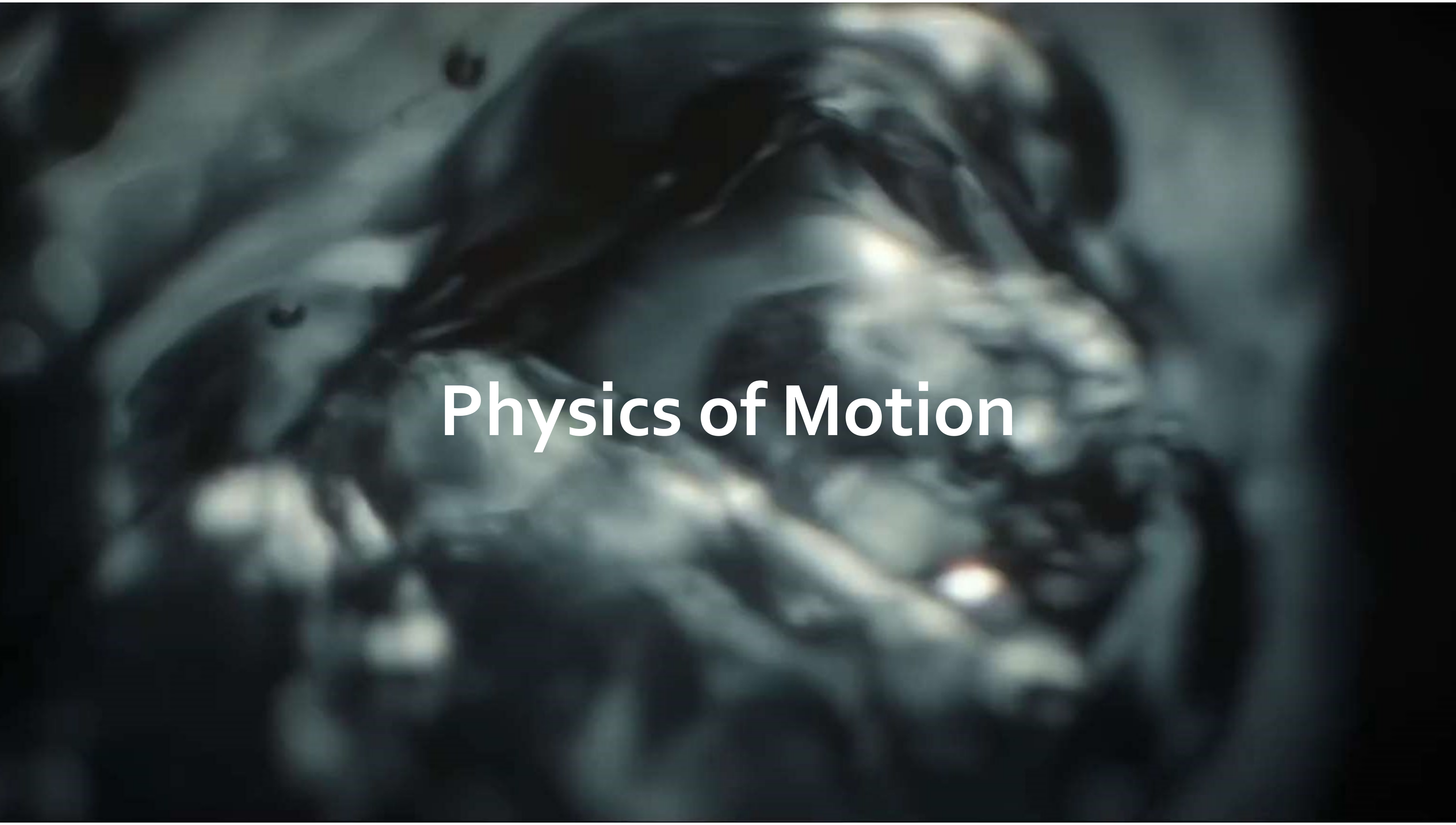
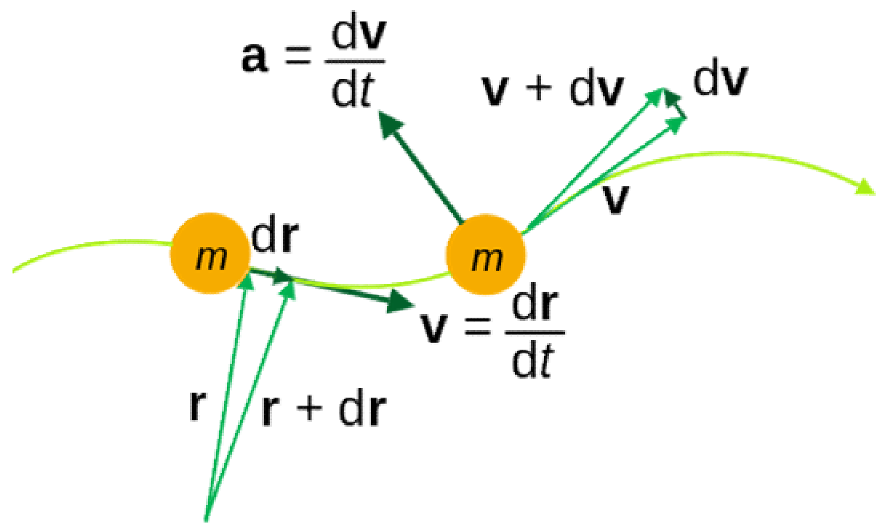


Physics of Motion

Location vs. Velocity vs. Acceleration

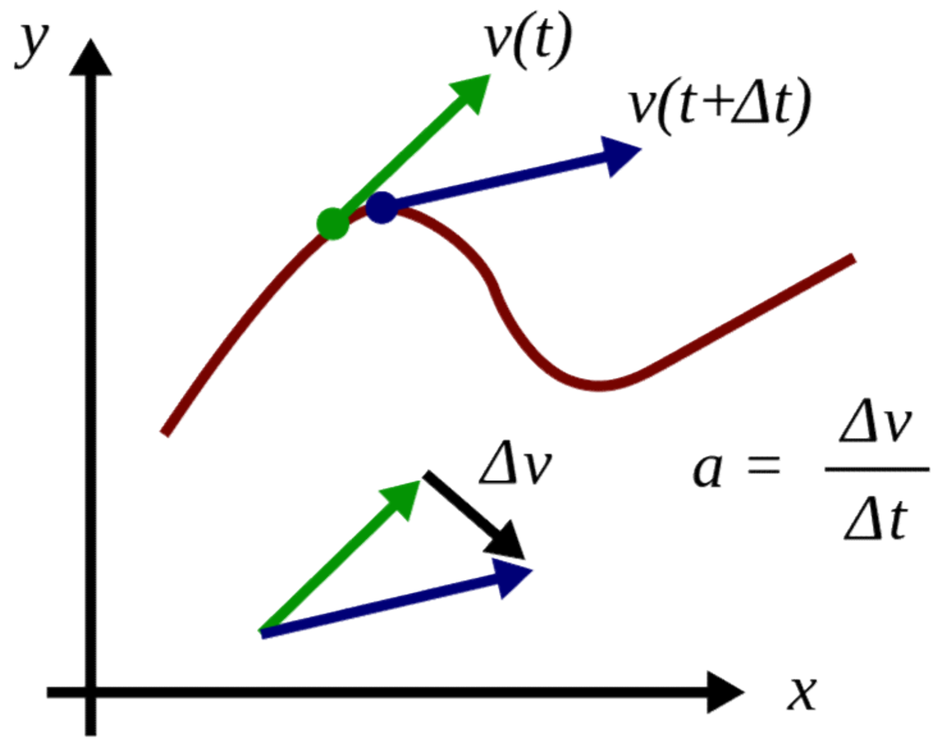
- Velocity = the rate of change of location $\left(\mathbf{v} = \frac{d\boldsymbol{r}}{dt}\right)$
- Acceleration = the rate of change of velocity $\left(\mathbf{a} = \frac{d\mathbf{v}}{dt}\right)$
- Acceleration affects velocity $\left(\mathbf{v} = \int \mathbf{a}dt\right)$
- Velocity affects location $\left(\boldsymbol{r} = \int \mathbf{v}dt\right)$
- Acceleration affects location $\left(\boldsymbol{r} = \iint \mathbf{a}dt\right)$
- if we find $\mathbf{a}$ we can calculate $\mathbf{v}$ and $\boldsymbol{r}$.



- $\longrightarrow$ only think about acceleration changing algorithms

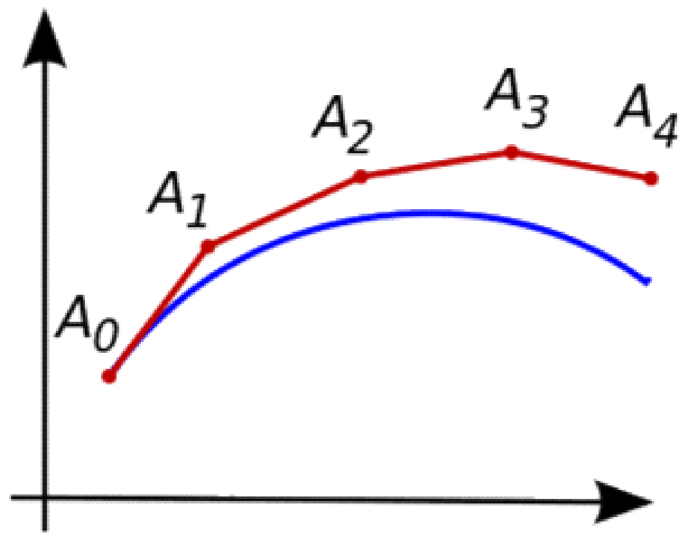
Location vs. Velocity vs. Acceleration (Discrete)

- $\mathbf{v}_{\Delta t} = \frac{\Delta \boldsymbol{r}}{\Delta t}$ $\mathbf{a}_{\Delta t} = \frac{\Delta \mathbf{v}}{\Delta t}$ will drop Δt subscript for visual simplicity
- How to integrate?



Forward Euler Method

- Numerical integration (solves first order differential equations)
- Idea: incrementally take small steps along tangent
- $f(t_0) = A_0$
- $f(t_{n+\Delta t}) = f(t_n) + \Delta t\, f'(t_n)$
- First order method – error proportional Δt
- Better integrators exist
  - More computationally expensive
  - Need to store/evaluate more data points



Location vs. Velocity vs. Acceleration (Discrete)

- $\mathbf{v} = \frac{\Delta \boldsymbol{r}}{\Delta t} \quad \mathbf{a} = \frac{\Delta \mathbf{v}}{\Delta t}$
- Euler integration in two steps
- Acceleration affects velocity ($\mathbf{v} = \mathbf{v_0} + \mathbf{a}\,\Delta t$)
- Velocity affects location ($\boldsymbol{r} = \boldsymbol{r_0} + \boldsymbol{v}\,\Delta t$)

Example Program

$\mathbf{a}cceleration = initialAcceleration;$
$\mathbf{v}elocity = initialVelocity;$
$\mathbf{p}osition = initialPosition;$

while (not finished) *// each frame once*

{

$\mathbf{a}cceleration = magicFunction(\mathbf{a}cceleration, \Delta t);$

$\mathbf{v}elocity \mathrel{+}= \mathbf{a}cceleration * \Delta t;$

$\mathbf{p}osition \mathrel{+}= \mathbf{v}elocity * \Delta t;$ *// move everything a little bit*

render(); *// draw everything*

}

Example Program with $\Delta t = 1$

$\mathbf{a}cceleration = initialAcceleration;$
$\mathbf{v}elocity = initialVelocity;$
$\mathbf{p}osition = initialPosition;$

while (not finished) *// each frame once*

{

$\mathbf{a}cceleration = magicFunction(\mathbf{a}cceleration);$

$\mathbf{v}elocity \mathrel{+}= \mathbf{a}cceleration;$

$\mathbf{p}osition \mathrel{+}= \mathbf{v}elocity;$ *// move everything a little bit*

render(); *// draw everything*

}

Newtonian Physics

1. A body will remain at rest or continue to move in a straight line at a constant velocity unless acted upon by another force.
   - (So, Atari Breakout had realistic physics! ☺)
2. The acceleration of a body is proportional to the resultant force acting on the body and is in the same direction as the resultant force.
3. For every action, there is an equal and opposite reaction.

- More recent physics show laws break down when trying to describe universe (Einstein), but still good enough for computer games

Newtons 2nd Law of motion

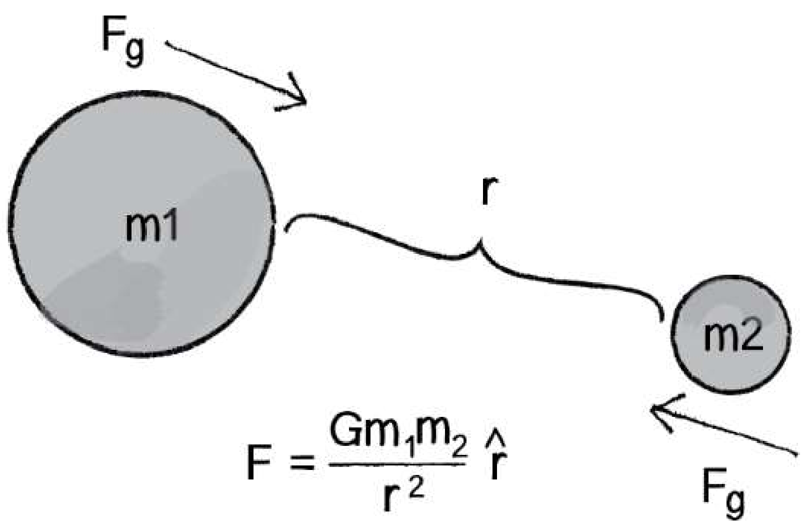
- $\boldsymbol{F} = m\boldsymbol{a}$
- Force = mass * acceleration … for a given body
- Force and accelerations are vectors
- We want acceleration
  - $\boldsymbol{a} = \frac{\boldsymbol{F}}{m} \ldots$ acceleration is indirect proportional to force applied to body
- More than one force?
  - $\boldsymbol{a} \mathrel{+}= \frac{\boldsymbol{F_i}}{m}$ … force accumulation

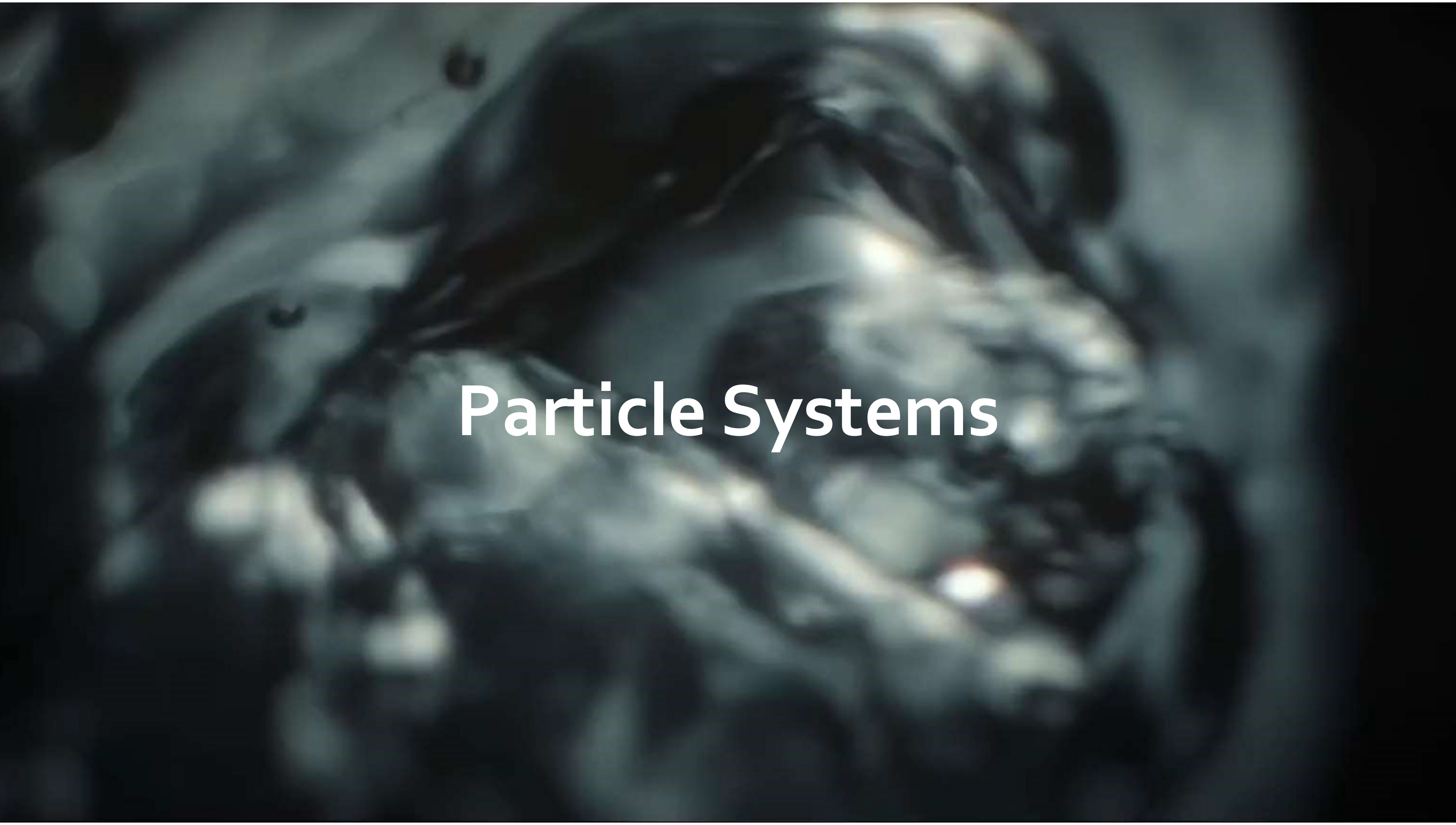
What is $\boldsymbol{F}$?

- Usual forces that hang around (with direction and magnitude)
  - Gravity
  - Friction
  - Resistance
  - Wind
  - Drag
  - Magnetism
  - …
  - Invent your force
    - Jedi
    - …

Force Example – Gravitational Attraction

- G …*universal gravitational constant* (6.67428 x 10^{-11})
- Direction?
  - Towards each other




Particle Systems