CGP

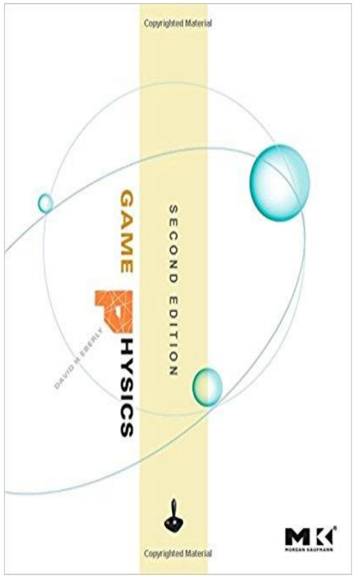
Dr Wojciech Palubicki

Next

- Physics: translational and rotational Newtonian physics, equations of motion for a system of bodies (**rigid body dynamics**)
- Numerical methods for integration
- Collision detection and Spatial Structures
- Advanced numerical methods for collision detection

References

- Game Physics, D. Eberly
- Real-Time Collision Detection, Christer Ericson



Rigid body

- A rigid body is characterized by the region that its mass is at.
- The simplest rigid body is a single **particle** of mass m that occupies a single location x.
- A **particle system** is a collection of a finite number of particles p_i (discrete body).

Physical model for a rigid body

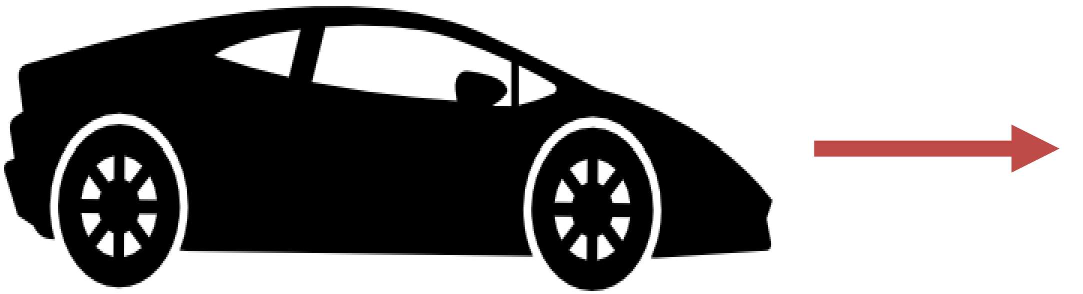
- Describe motion
- Position, Velocity and Acceleration
- Rotation, Angular velocity and Torque
- Cartesian coordinates in 2D and 3D

Newton's laws of motion

- Linear momentum
- Angular momentum

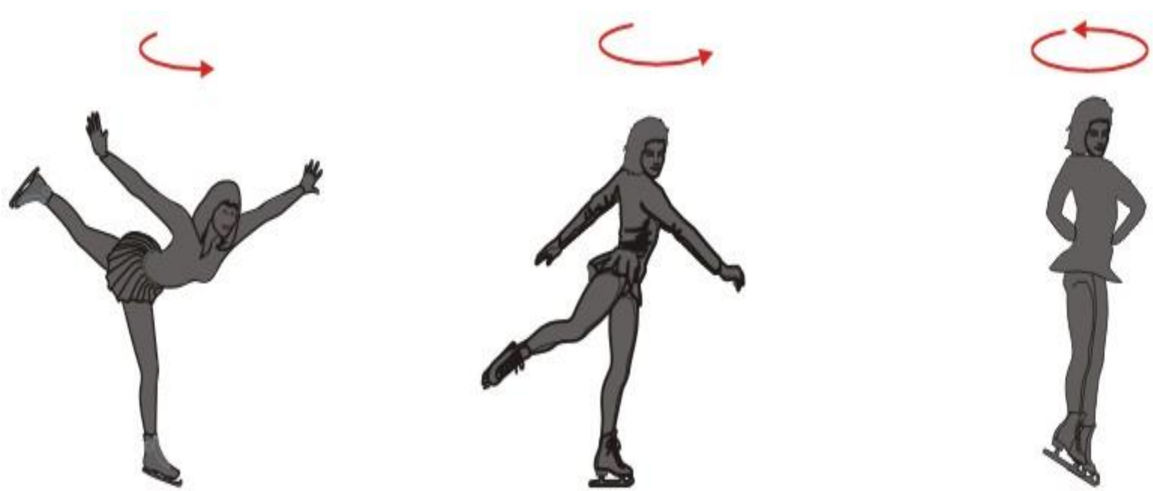
Newton’s laws of motion

- <u>Linear momentum</u>
- Angular momentum



Newton's laws of motion

- Linear momentum
- <u>Angular momentum</u>



Angular momentum

- Center of mass
- Inertia
- Torque

Particle kinematics 2D

- We start with a particle moving across the xy plane
- Position at time t is

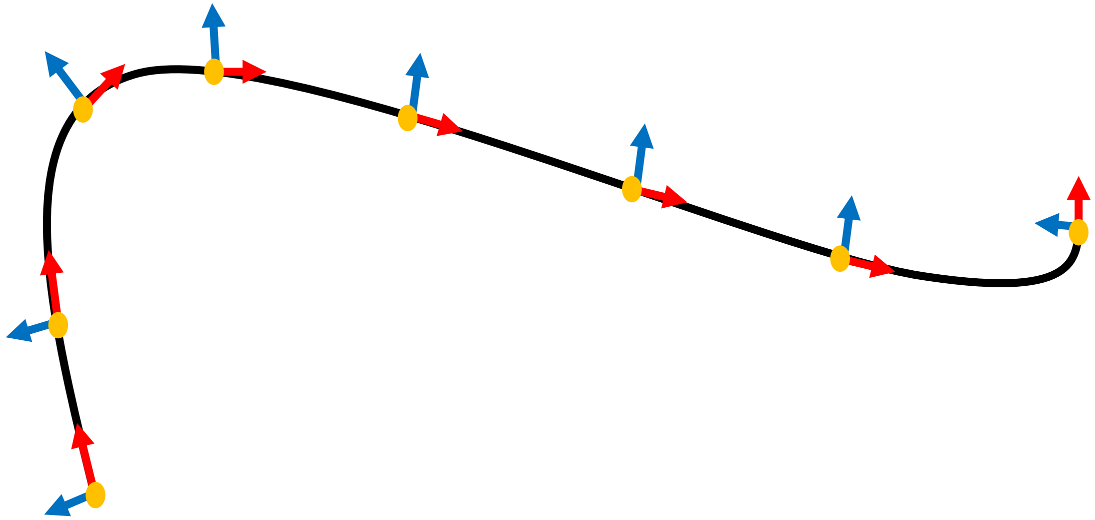
$$r(t) = \ x(t) + \ y(t)$$

or $(x(t), y(t))$

Particle kinematics 2D

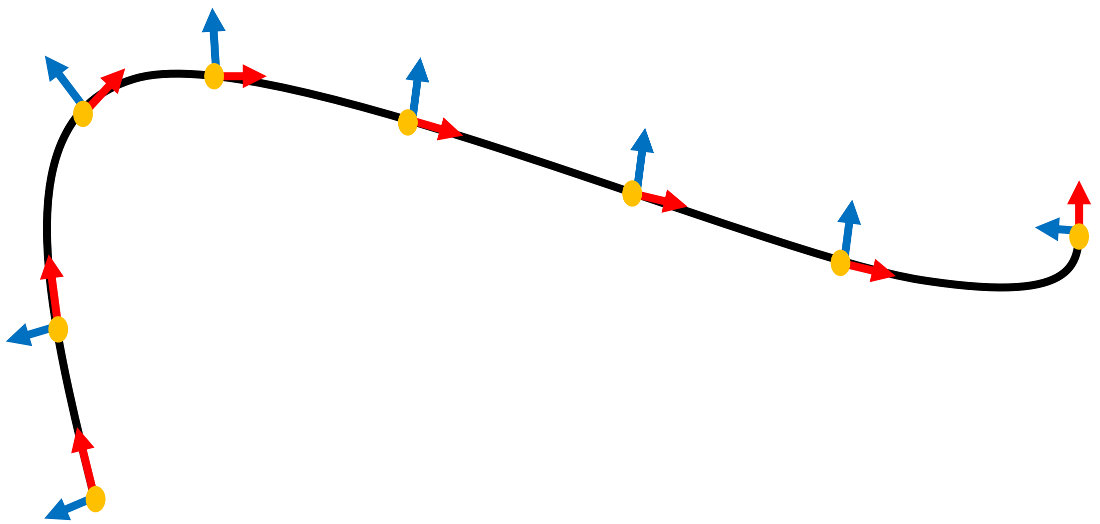
- Velocity at time t is $v(t) = \dot{r} = (\dot{x}, \dot{y})$
- Speed is $\dot{s} = |v|$
- Acceleration is $a(t) = \dot{v} = \ddot{r} = (\ddot{x}, \ddot{y})$

Particle kinematics 2D



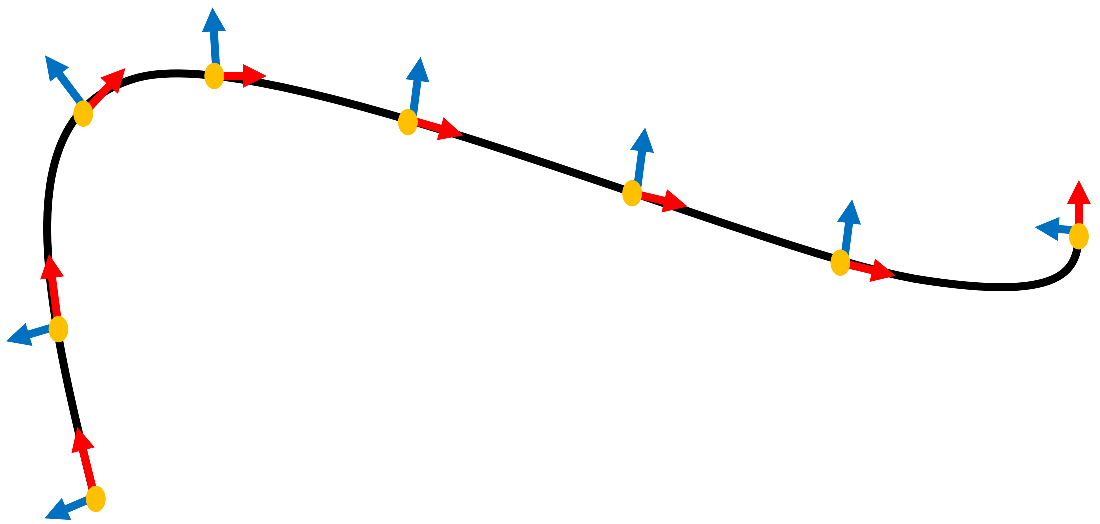
Particle kinematics 2D

- Tangent is
  $T(t) = \frac{v}{|v|}$



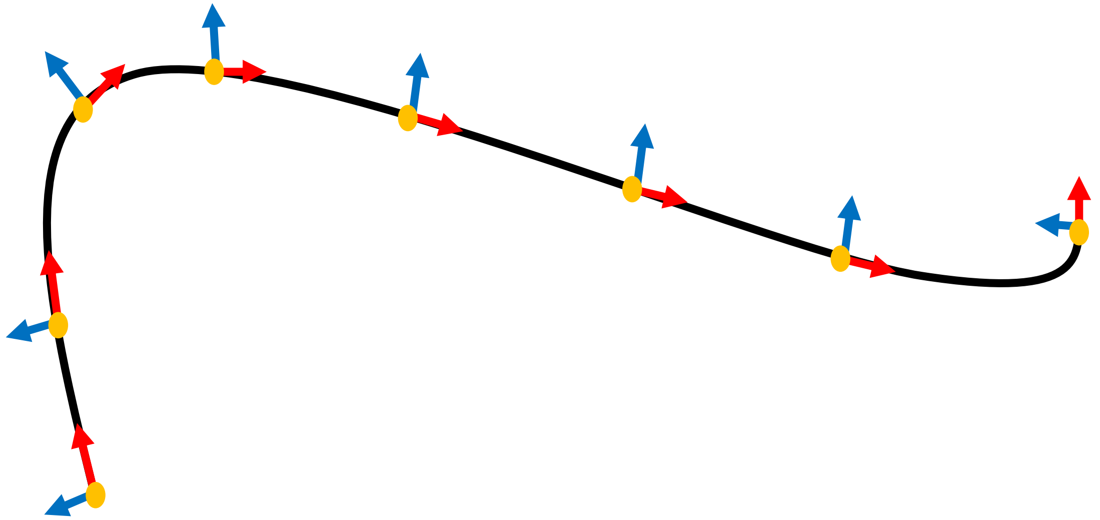
Particle kinematics 2D

- Tangent is
  $$T(t) = \frac{v}{|v|} = (\cos(\Phi(t)), \sin(\Phi(t)))$$



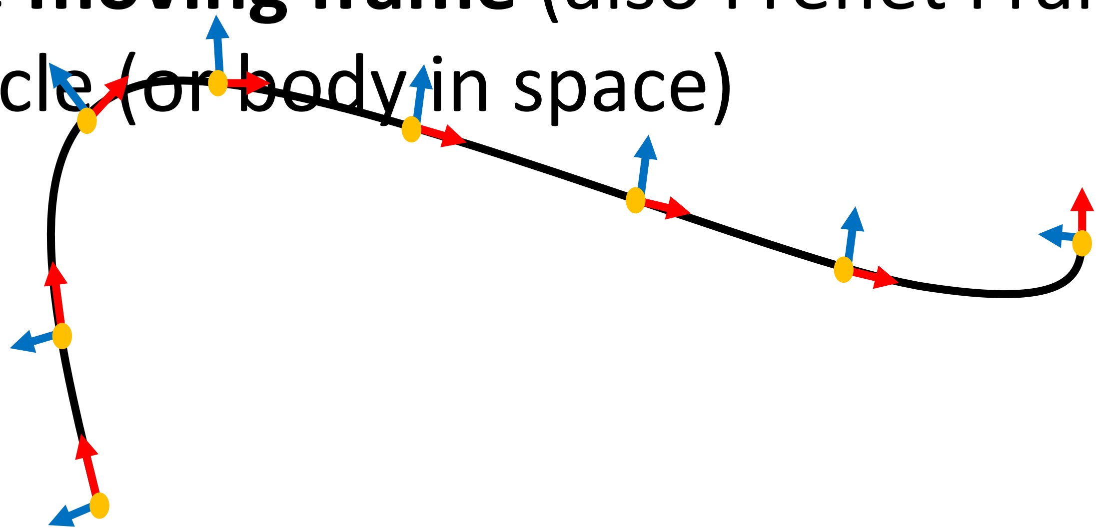
Particle kinematics 2D

- Tangent is
$$T(t) = \frac{v}{|v|} = (\cos\big(\Phi(t)\big), \sin(\Phi(t)))$$
- Normal is $N(t) = (-sin\big(\Phi(t)\big), \cos(\Phi(t)))$



Particle kinematics 2D

- Tangent is $T(t) = \frac{v}{|v|} = (\cos(\Phi(t)), \sin(\Phi(t)))$
- Normal is $N(t) = (-sin(\Phi(t)), \cos(\Phi(t)))$
- r, T, N is the **moving frame** (also Frenet Frame) of the particle (or body in space)



Particle kinematics 3D

- Position at time t is

$$r(t) = x(t) + y(t) + z(t)$$

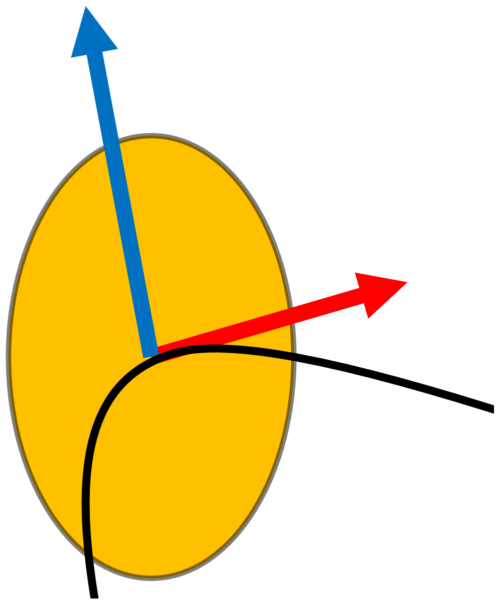
or $(x(t), y(t), z(t))$

Particle kinematics 3D

- Velocity at time t is $v(t) = \dot{r} = (\dot{x}, \dot{y}, \dot{z})$
- Speed is $\dot{s} = |v|$
- Acceleration is $a(t) = \dot{v} = \ddot{r} = (\ddot{x}, \ddot{y}, \ddot{z})$

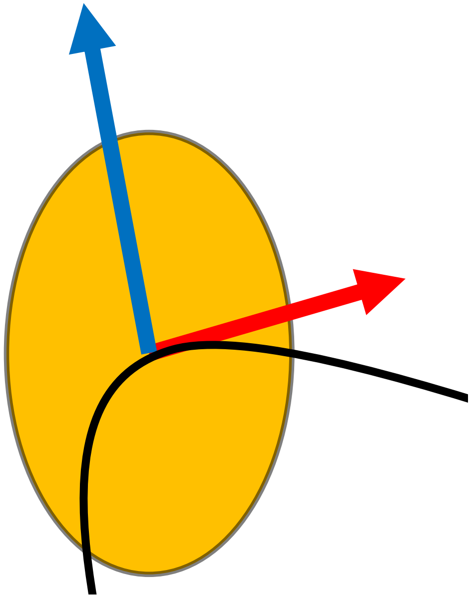
Particle kinematics 3D

- Tangent is $T(t) = \frac{v}{|v|}$



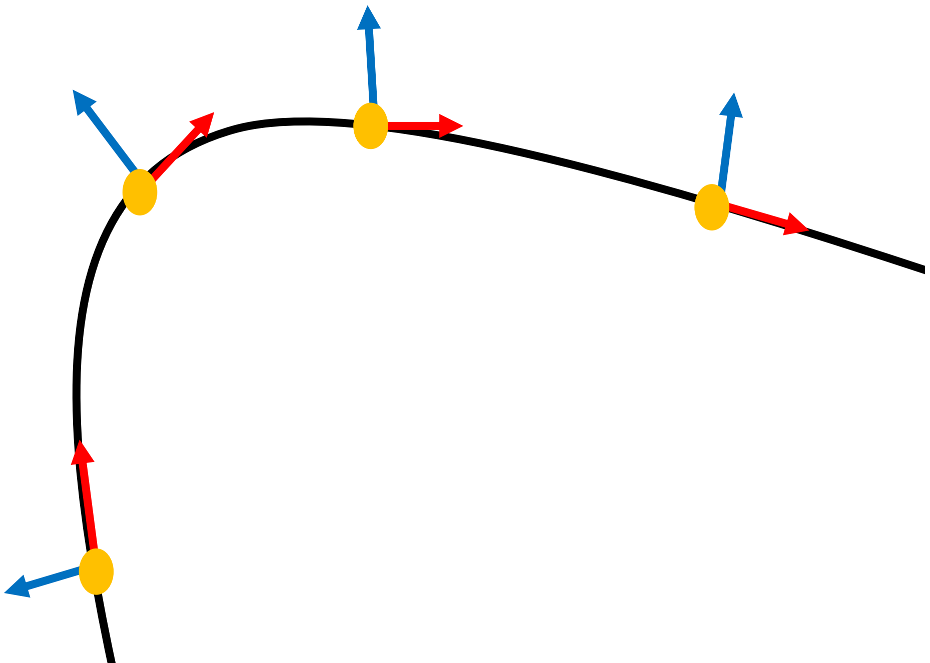
Particle kinematics 3D

- Tangent is $T(t) = \frac{v}{|v|}$
- We have an infinite set of possible vector normals to T



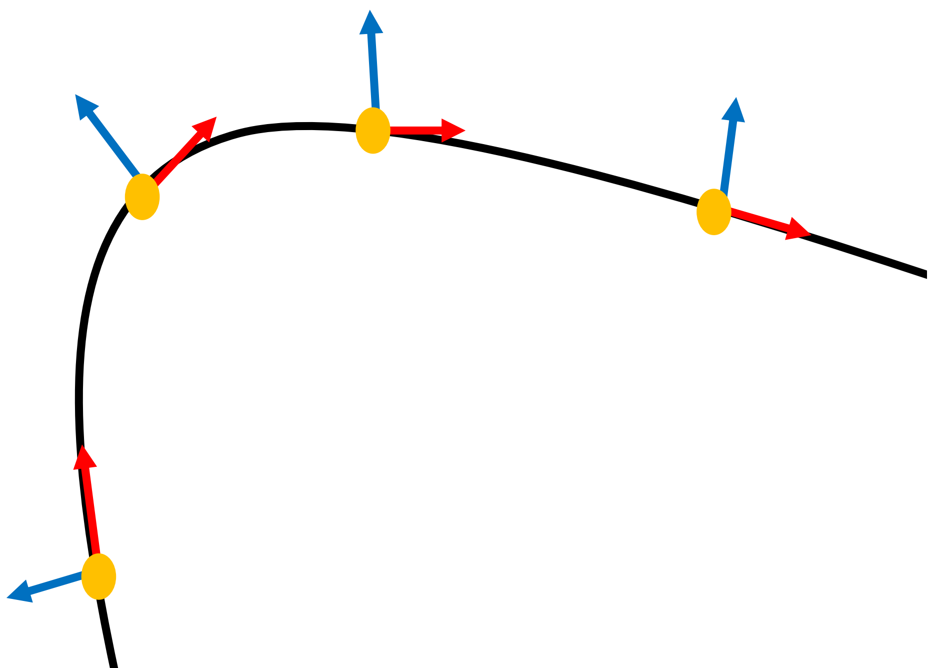
Particle kinematics 3D

- Define normal N as change of T, that is
  $\frac{dT}{ds} = kN$



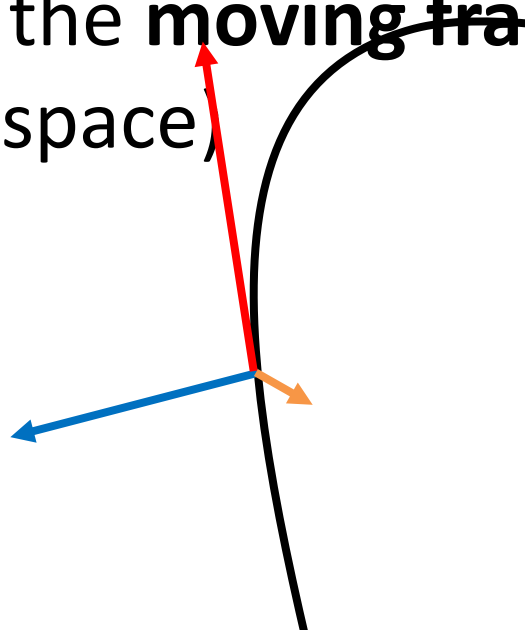
Particle kinematics 3D

- Define normal N as change of T, that is $\frac{dT}{ds} = kN$
- Binormal is $B = T \times N$



Particle kinematics 3D

- Define normal N as change of T, that is $\frac{dT}{ds} = kN$
- Binormal is $B = T \times N$
- r, T, N, B is the **moving fra**
  (or body in space)



Rigid Body Rotation

- $R = [T\ N\ B]$ put in matrix form is the **rotation matrix** of the body

Rigid Body Rotation

- $R = [T\ N\ B]$ put in matrix form is the **rotation matrix** of the body

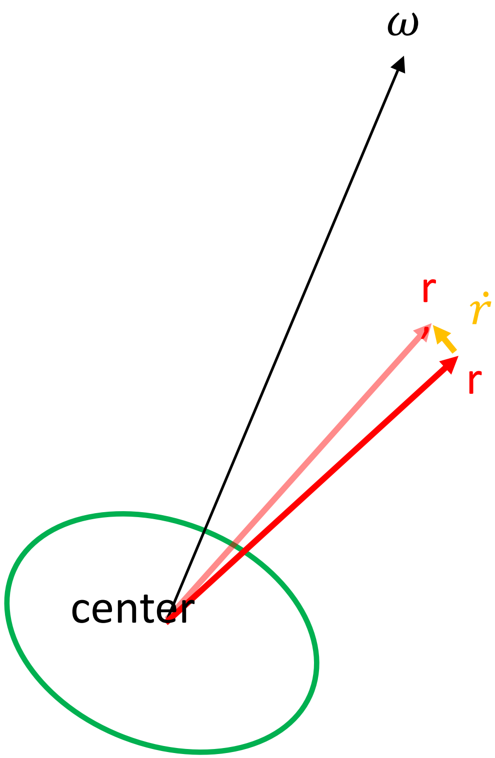
- $r(t) = R(t)r_0 + x(t)$ where $x(t)$ is the position of the **center** of the body

Rigid Body Rotation

- $R = [T\ N\ B]$ put in matrix form is the **rotation matrix** of the body
- $r(t) = R(t)r_0 + x(t)$ where $x(t)$ is the position of the **center** of the body
- $\omega(t)$, a vector, is the **angular velocity** of the body
  - Its direction is the rotation axis
  - Its magnitude is in rad/s

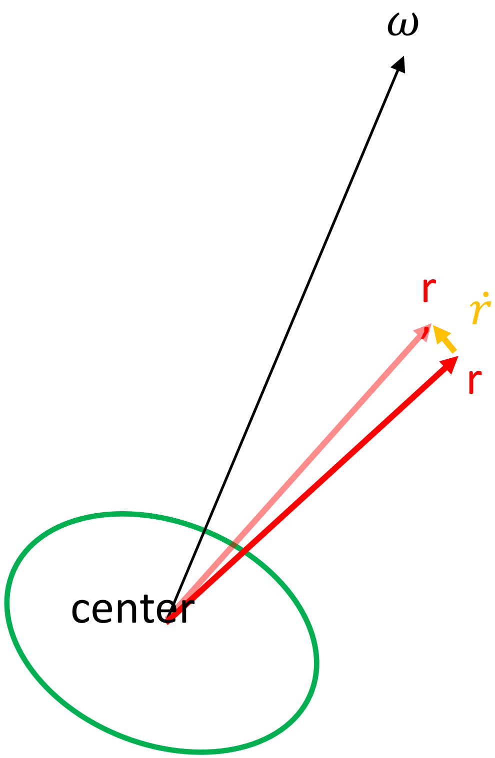
Rigid Body Rotation

- To determine $\dot{R}$ we calculate $\dot{r}(t)$



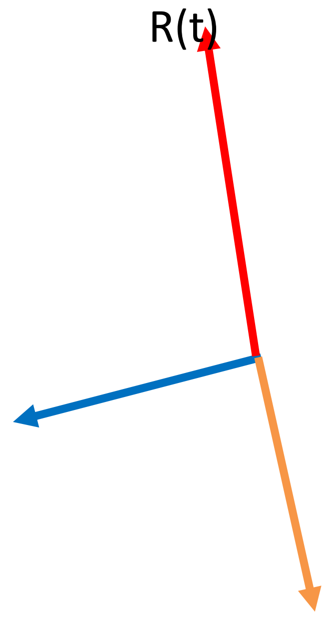
Rigid Body Rotation

- To determine $\dot{R}$ we calculate $\dot{r}(t)$

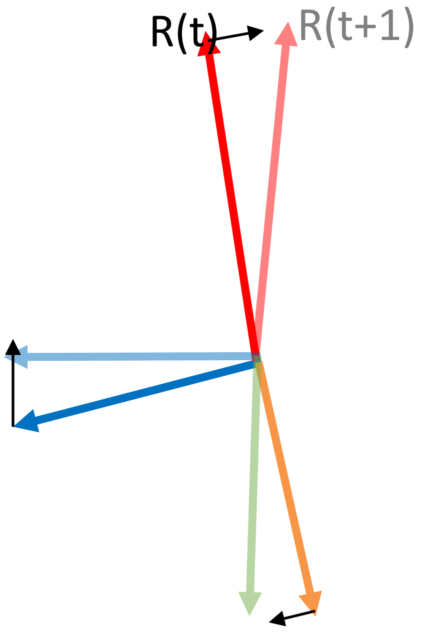


$$\dot{r} = \omega(t) \times r(t)$$

Geometric Interpretation of $\dot{R}$



Geometric Interpretation of $\dot{R}$



Rigid Body Rotation

- We now consider the columns of the rotation matrix

- We compute $\dot{T} = \omega(t) \times T, \dot{N} = \omega(t) \times N$ and $\dot{B} = \omega(t) \times B$

Rigid Body Rotation

- We now consider the columns of the rotation matrix

- We compute $\dot{T} = \ \omega(t) \ \times \ T, \dot{N} = \omega(t) \ \times \ N$ and $\dot{B} = \omega(t) \ \times \ B$

- $\dot{T}, \dot{N}$ and $\dot{B}$ are the velocities of the axes of the moving frame (the columns of $\dot{R}$)

Newton’s Laws

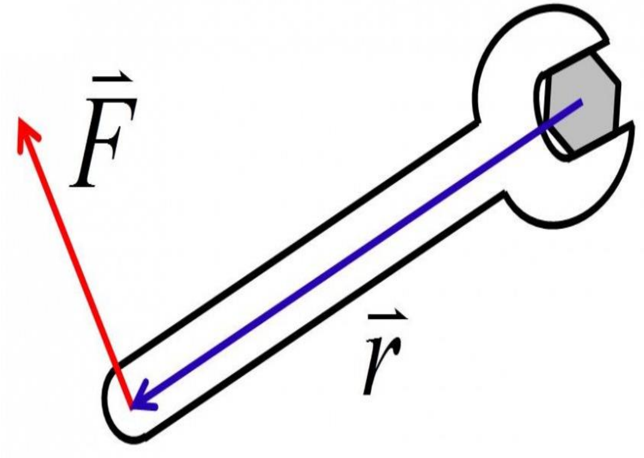
- **Inertia**, the tendency of an object to resist change of motion
- **Force**, the mechanism by which inertia is changed

Newton’s Laws

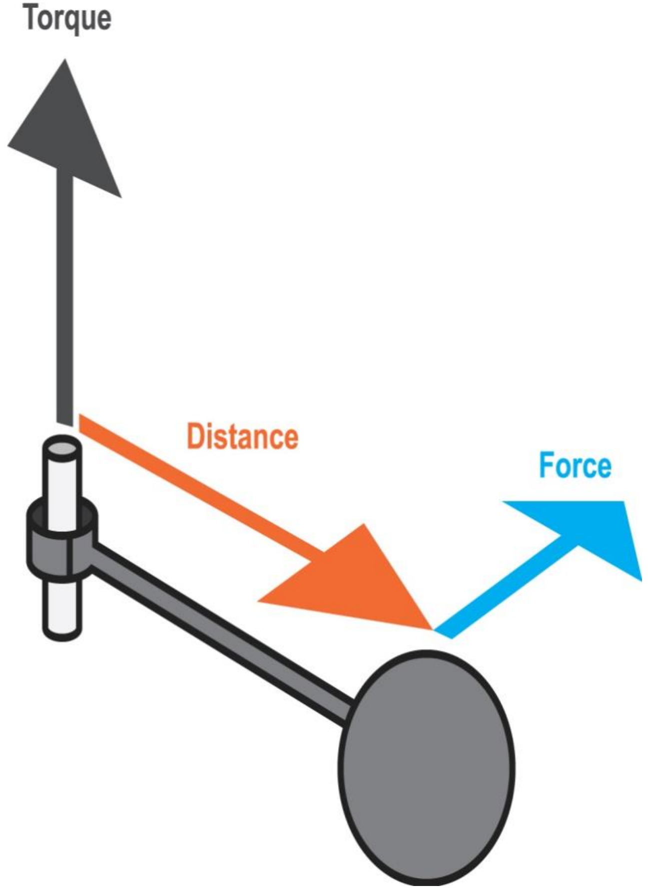
- The **second law** is most useful to game physics engines
- It states that the application of an external force on an object causes a change in the object's momentum over time.
- Mass is assumed to be always constant, so
- $F = \frac{d}{dt}(mv) = \frac{d}{dt}(mv) = ma$
- Each of the vector quantities of position, velocity, and acceleration is measured with

From Force to Torque

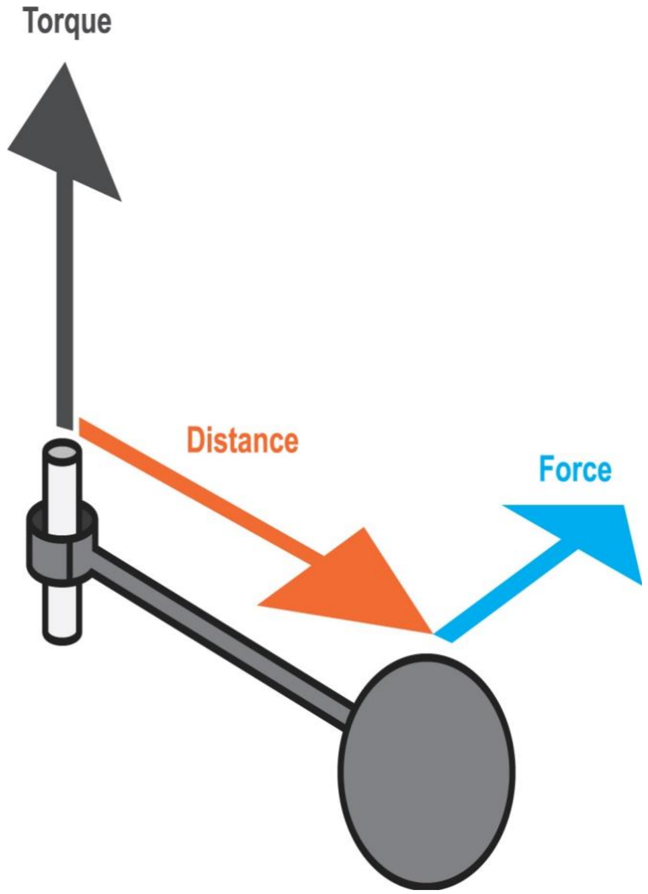
- Removing log nuts with a wrench
- Exert a force on the end of the wrench, the nut turns
- The longer the wrench, the easier (but slower) the nut tur



Torque τ



Torque τ



$$\tau = r \times F$$

Torque

- The ease of turning is proportional to the length of the wrench and the force applied
- This product is referred to as torque or moment of force
- Torque is defined as $\tau \ = \ r \ \times \ F$
  – Direction of torque is axis of rotation
  – Length of torque is in rad/s

Multiple Torques

- Multiple torques (just like forces) are simply added together
- $\tau = \sum_i r_i \times F_i$ (discrete body)

Momenta

- Quantification of Newton's Second Law
- How much motion does the body have?
  - **A lot** means that a lot of force is needed to change it
  - **Little** means that little force is needed to change it

Linear momentum

- How much linear motion does a body have?
- $p = mv = \sum_i m_i v_i$

Linear momentum

- How much linear motion does a body have?
- $p = mv = \sum_i m_i v_i$
- Linear momentum is conserved in a system (all bodies $dp/dt = 0$)

Linear momentum

- How much linear motion does a body have?
- $p = mv = \sum_i m_i v_i$
- Linear momentum is conserved in a system (all bodies $dp/dt = 0$)

$$\frac{d(mv)}{dt} = m\,\frac{dv}{dt} = ma = F$$

Linear momentum

- How much linear motion does a body have?
- $p = mv = \sum_i m_i v_i$
- Linear momentum is conserved in a system (all bodies $dp/dt = 0$)
- Force integrates linear momentum directly

$$\frac{dp}{dt} = \frac{d(mv)}{dt} = m \frac{dv}{dt} = ma = F$$

Angular momentum

- How much rotational motion does a body have?
- $L =$

Angular momentum

- How much rotational motion does a body have?
- $L = r \times p =$

Angular momentum

- How much rotational motion does a body have?
- $L = r \times p = mr \times v$

Angular momentum

- How much rotational motion does a body have?
- $L = r \times p = mr \times v = \sum_i m_i r_i \times v_i$
- Right-hand rule of cross-product:
  - Angular momentum refers to the tendency of the body to rotate around a given axis, L
  - The longer the axis, the harder it is to stop the rotation

Angular momentum

- The derivative of the angular momentum is torque (when the body does not change shape, $dr/dt = 0$)

$$\frac{dL}{dt} = \tau$$

Center of mass

- In mechanical systems, each object can behave as if its mass is concentrated at a single point. The location of this point is called the **center of mass** of the object.
- We compute the center of mass by a weighted average of the body particles relative positions x_i and their respective masses m_i (M is the mass of the whole body)

$$\bar{x} = \sum \frac{m_i x_i}{}$$

Force projection

- When an **external force** F_{ext} is applied to a body from some position r_f
- We use the center of mass to split the force between linear force and torque

$$F = F_{ext}(F_{ext} \cdot (r_f - \bar{x})),$$
$$\tau = F_{ext} \times (r_f - \bar{x})$$

The implementation of the model so far

- Linear motion:

$$x$$

The implementation of the model so far

- Linear motion:

```
x = x + dt*v
v
```

The implementation of the model so far

- Linear motion:

```
x = x + dt*v
v = v + dt*a
    a
```

The implementation of the model so far

- Linear motion:

```
x = x + dt*v
v = v + dt*a
   a = f /m
```

The implementation of the model so far

- Linear motion:

```
x = x + dt*v
v = v + dt*a
   a = f /m
```

- Angular motion:

The implementation of the model so far

- Linear motion:

```
x = x + dt*v
v = v + dt*a
   a = f /m
```

- Angular motion:

```
R =
```

The implementation of the model so far

- Linear motion:

```
x = x + dt*v
v = v + dt*a
a = f /m
```

- Angular motion:

```
R = R + dt*dR
```

dR

The implementation of the model so far

- Linear motion:

```
x = x + dt*v
v = v + dt*a
a = f /m
```

- Angular motion:

```
R = R + dt*dR
dR = [omega*T, omega*N, omega*B]
```

The implementation of the model so far

- Linear motion:

```
x = x + dt*v
v = v + dt*a
a = f /m
```

- Angular motion:

```
R = R + dt*dR
dR = [omega*T, omega*N, omega*B]
omega?
```

The implementation of the model so far

- Linear motion:

```
x = x + dt*v
v = v + dt*a
a = f /m
```

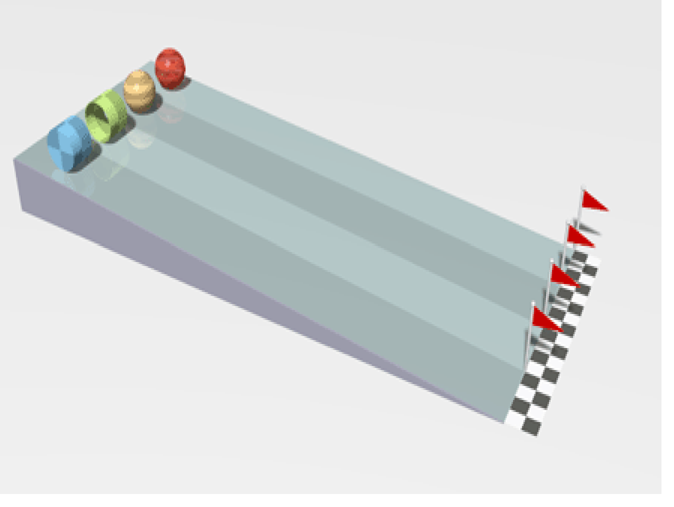
- Angular motion:

```
R = R + dt*dR
dR = [omega*T, omega*N, omega*B]
omega?
```

$$L = I \times \omega$$

Moments of Inertia

- How difficult is it to set an object into rotation around an axis?
- Rotational equivalent to mass for linear movement



Moment of Inertia

- Empirical studies for a **single** particle show that the moment of inertia is $\boldsymbol{mr^2}$, where m is the mass of the particle and r is its distance to the axis.

Moment of Inertia in 2D

- The more a particle weighs or the further from the center the harder to set the object into rotation
- In 2D the moment of inertia is a single number because we can only rotate in one plane

$$I = \sum_i m_i \,|(x_i , y_i) - (\bar{x}, \bar{y})|^2$$

Moment of Inertia in 3D

- Harder to express, because suddenly we can rotate along an infinite number of axes
- Consider a particle
  - Located at relative vector r
  - Moving with linear velocity $v = \omega \times r$

Mass matrix in 3D

- $L_i = r_i \times m_i v_i = m_i r_i \times (\omega \times r_i) = J_i \omega$

Mass matrix in 3D

- $L_i = r_i \times m_i v_i = m_i r_i \times (\omega \times r_i) = J_i \omega$

- $J_i = m_i \begin{bmatrix} {y_i}^2 + {z_i}^2 & -x_i y_i & -x_i z_i \\ -x_i y_i & {x_i}^2 + {z_i}^2 & -y_i z_i \\ -x_i z_i & -y_i z_i & {y_i}^2 + {z_i}^2 \end{bmatrix}$

Mass matrix in 3D

- $L_i = r_i \times m_i v_i = m_i r_i \times (\omega \times r_i) = J_i \omega$

- $J_i = m_i \begin{bmatrix} {y_i}^2 + {z_i}^2 & -x_i y_i & -x_i z_i \\ -x_i y_i & {x_i}^2 + {z_i}^2 & -y_i z_i \\ -x_i z_i & -y_i z_i & {y_i}^2 + {z_i}^2 \end{bmatrix}$
- $L_i = J_i \omega$, just like $p = mv$

Mass matrix in 3D

- $L_i = r_i \times m_i v_i = m_i r_i \times (\omega \times r_i) = J_i \omega$

- $J_i = m_i \begin{bmatrix} {y_i}^2 + {z_i}^2 & -x_i y_i & -x_i z_i \\ -x_i y_i & {x_i}^2 + {z_i}^2 & -y_i z_i \\ -x_i z_i & -y_i z_i & {y_i}^2 + {z_i}^2 \end{bmatrix}$
- $L_i = J_i \omega$, just like $p = mv$
- For the whole body we sum all the J_i matrices of the particles
- $J = \sum_i J_i, L = J\omega$

Mass matrix in 3D

- $L_i = r_i \times m_i v_i = m_i r_i \times (\omega \times r_i) = J_i \omega$

- $J_i = m_i \begin{bmatrix} {y_i}^2 + {z_i}^2 & -x_i y_i & -x_i z_i \\ -x_i y_i & {x_i}^2 + {z_i}^2 & -y_i z_i \\ -x_i z_i & -y_i z_i & {y_i}^2 + {z_i}^2 \end{bmatrix}$
- $L_i = J_i \omega$, just like $p = mv$
- For the whole body we sum all the J_i matrices of the particles

- $J = \sum_i J_i \, , L = J\omega$

Whole Model - Position

- **Position**, integrated from velocity $\dot{x} = v$
- **Velocity**, derived from linear momentum $\dot{v} = p/m$
- **Linear momentum**, integrated from force

$$\dot{p} = F_{ext}(F_{ext} \cdot (r_f - \bar{x}))$$

Whole Model - Rotation

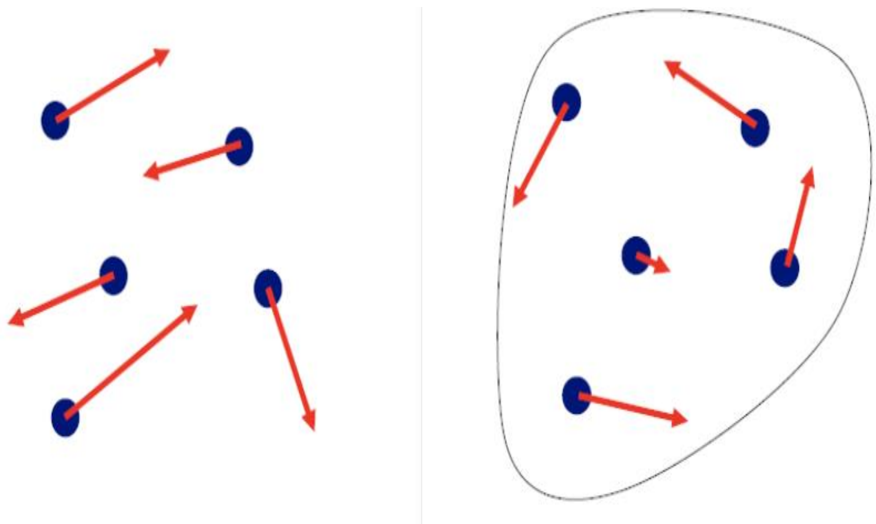
- **Rotation**, integrated from angular veolcity

$$\dot{R} = [\omega \times T \;\; \omega \times N \;\; \omega \times B]$$

- **Angular velocity**, derived from angular momentum $\omega = J^{-1}L$
- **Angular momentum**, integrated from torque

$$\dot{L} = \tau = F_{ext} \times (r_f - \bar{x})$$

From Particles to Rigid Bodies



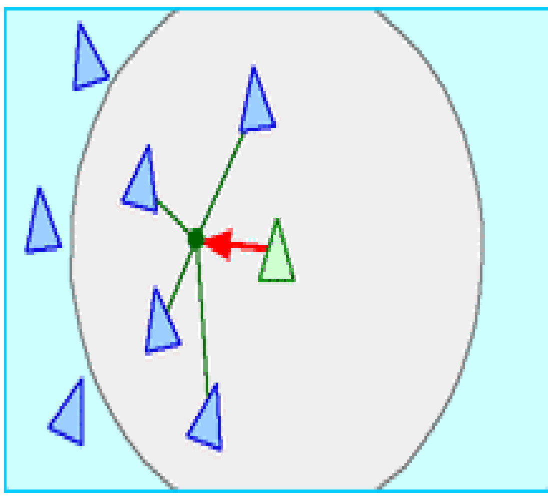
- Particles
  - No rotations
  - Linear velocity only
  - 3 DoF
- Rigid bodies
  - 6 DoF (translation + rotation)
  - Linear velocity
  - Angular velocity

Flocking (C. Reynolds)

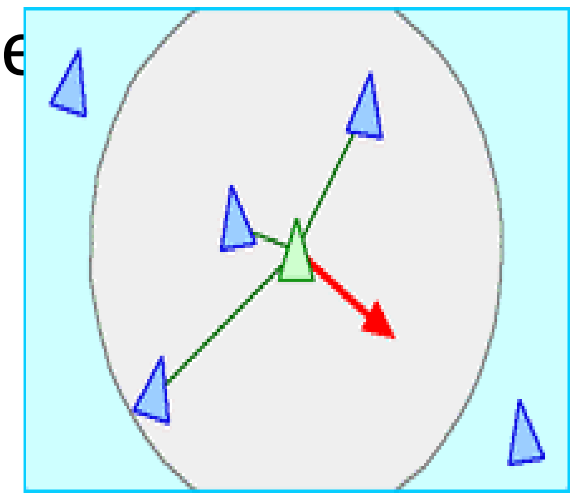
Reynolds, Craig (1987). "Flocks, herds and schools: A distributed behavioral model." SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques (Association for Computing Machinery): 25–34

Flocking

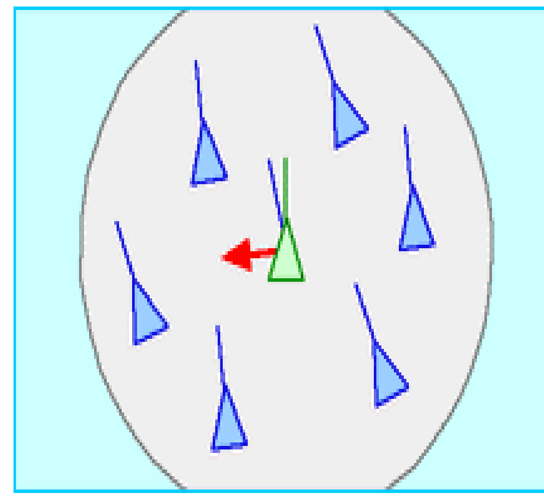
- We will model three virtual forces describing the local interaction between particles (in fact we will calculate dire



Cohesion



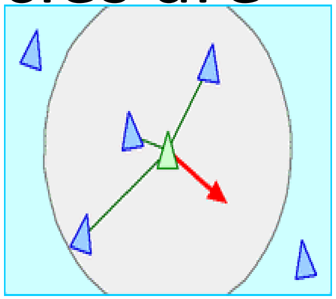
Separation



Alignment

Local neighborhood

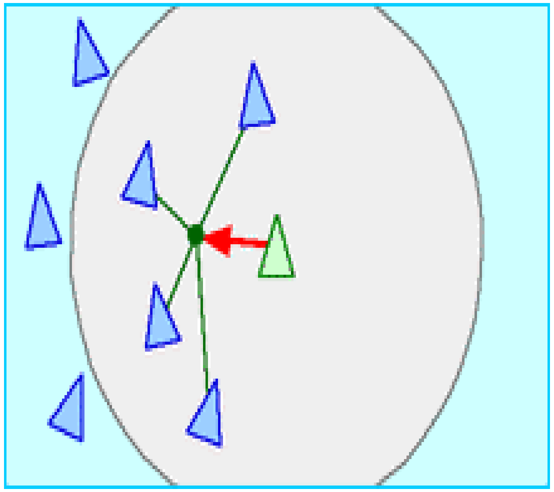
- For position $\vec{b}$ of a given particle we test whether positions $\vec{x}_i$ of other particles are smaller than d:
- $\left|\vec{b} - \vec{x}_i\right| < d$



Cohesion

Compute the vector pointing to the center of mass of neighboring particle positions

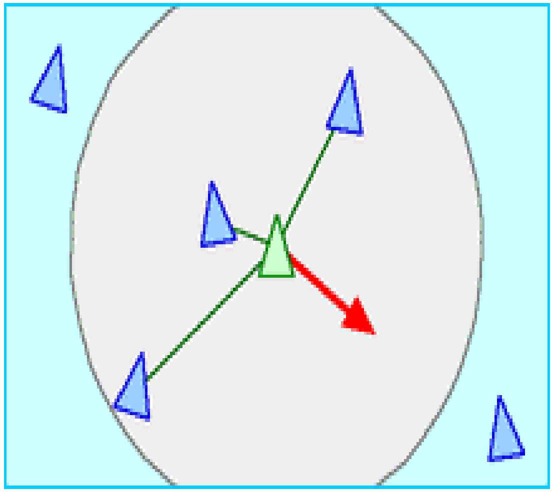
- $\vec{v}_1 = \frac{\sum_1^n \vec{x}_i}{n} - \vec{b}$



**Spójnoś
ć**

Separation

- Pseudocode:
- $\vec{v}_2$ *zero vector*
- *for all neighboring particles*
  - If $\left|\vec{b} - \vec{x}_i\right| < \boldsymbol{min}$
    - *then* $\vec{v}_2 = \vec{v}_2 - (\vec{b} - \vec{x}_i)$
- *return* $\vec{v}_2$

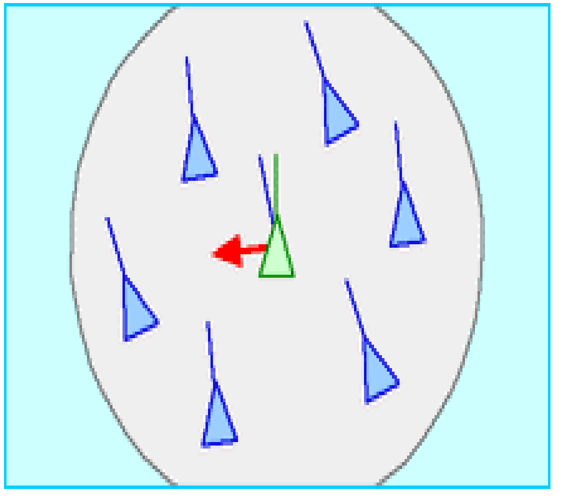


Separation

Alignment

- Calculate the average velocity of neighboring particles $\vec{x}_{v_i}$ and determine new velocity $\vec{v}_3$

- $\vec{v}_3 = \frac{\sum_1^n \vec{x}_{v_i}}{n} - \vec{b}_v$



Wyrówn
anie

Weighted average of virtual forces

- Velocity:
- $\vec{b}_v = \vec{b}_v + w_1\vec{v}_1 + w_2\vec{v}_2 + w_3\vec{v}_3$

Weighted average of virtual forces

- Velocity:
- $\vec{b}_v = \vec{b}_v + w_1\vec{v}_1 + w_2\vec{v}_2 + w_3\vec{v}_3$

- Position:
- $\vec{b} = \vec{b} + \vec{b}_v$

Exerc

- Implement boids algorithm and represent particles geomtrically as spheres

Exer

- Substitute spheres with the fish model. How can we extend our particle system to have a better flocking simulation?