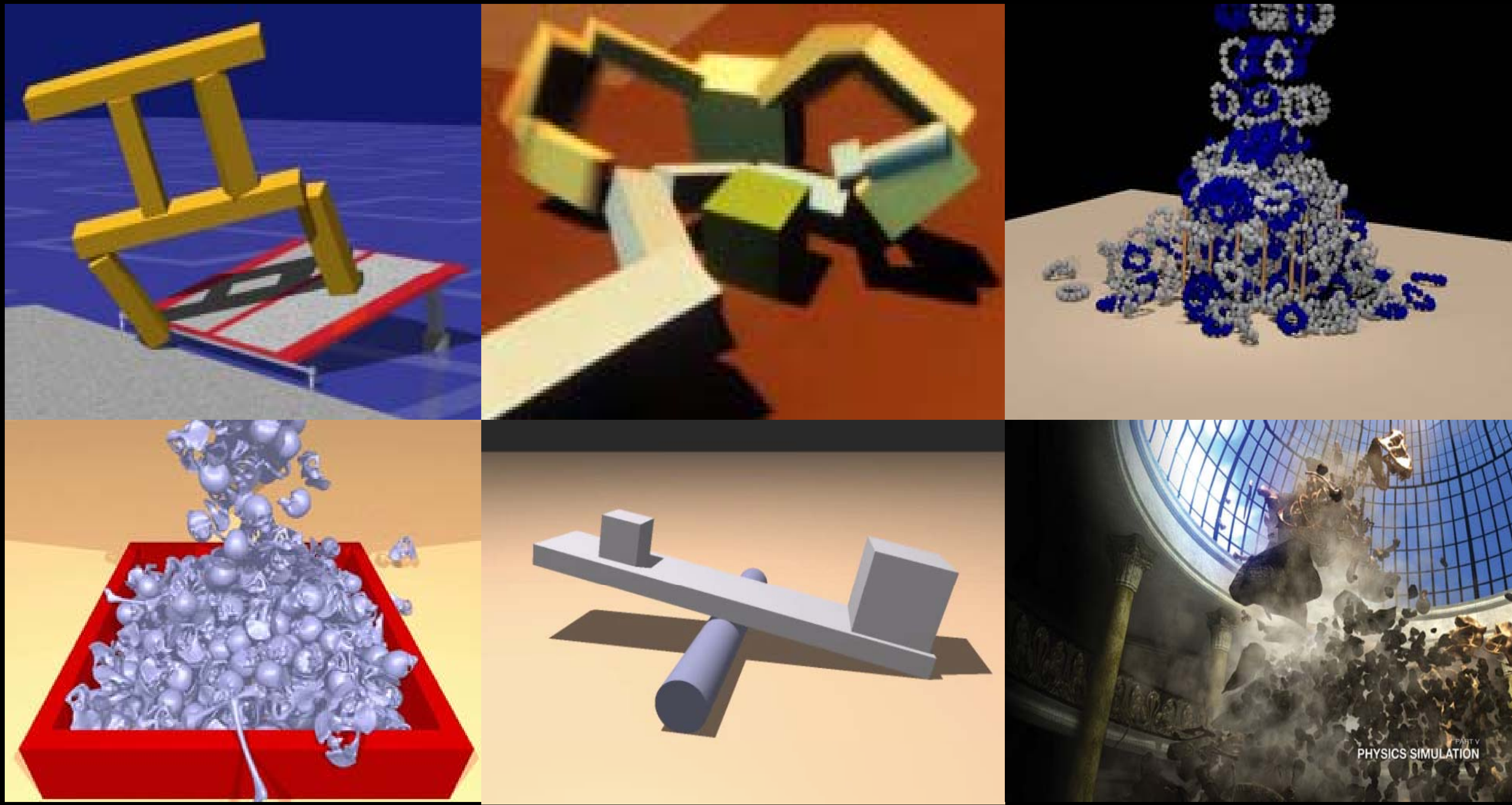


Rigid Body Dynamics

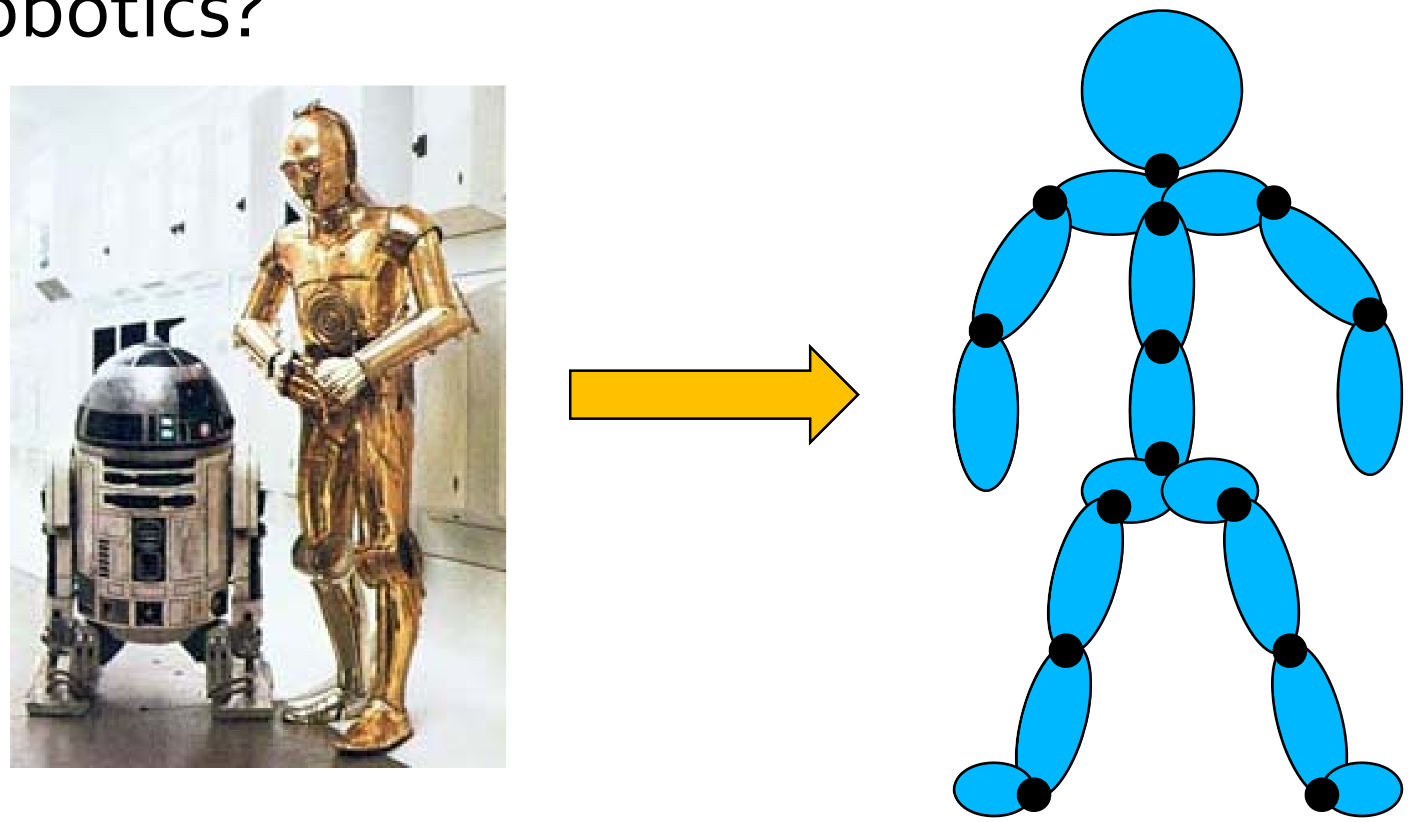
Tim Johnson & Michael Su

Agenda

- Introduction
- Notations
- Particle dynamics (basics)
- Rigid body dynamics
- Numerical integration
- Simulation loop
- Collision response

Why Study Rigid Bodies

- Why do we need to study the rigid body for robotics?

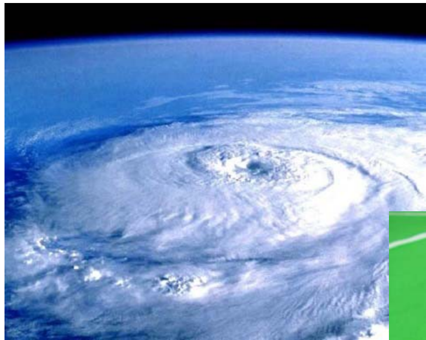


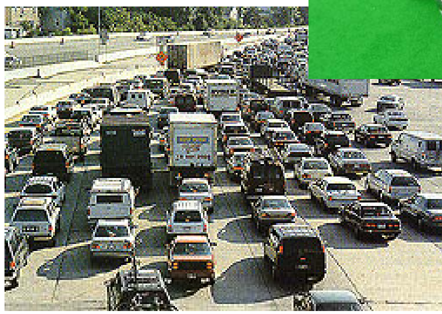
What is Dynamics?

- The study of bodies in motion
- Comprises kinematics and kinetics
  - Kinematics: describing the motion
  - Kinetics: explaining the motion
- Used to model many different objects
  - Particles
  - Rigid bodies
  - Fluids

Why Study Dynamics (1)

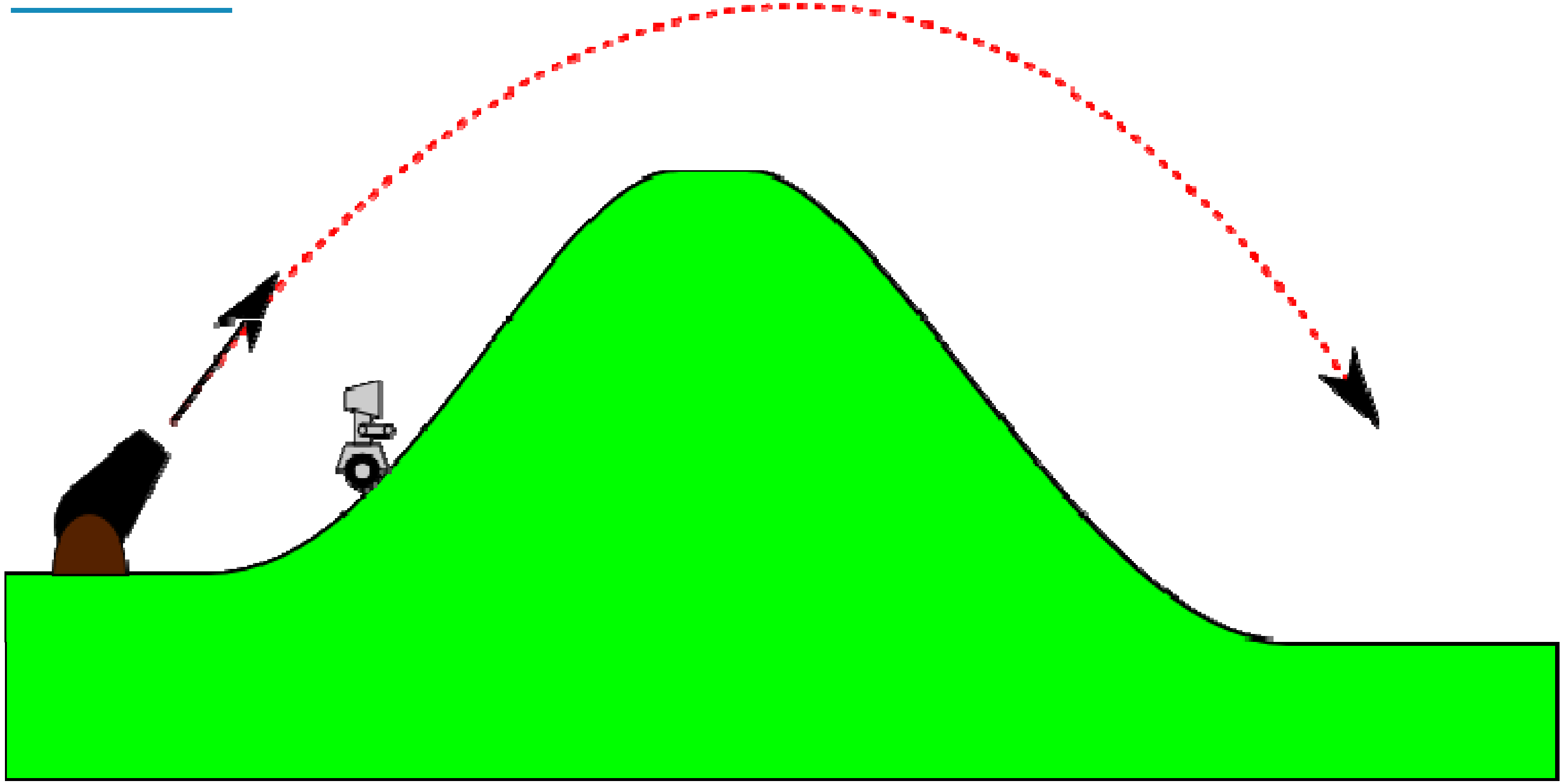
- To model and understand the world around us
  - Simulation
  - Animation
- To make predictions
  - Weather
  - Traffic
- To control robots





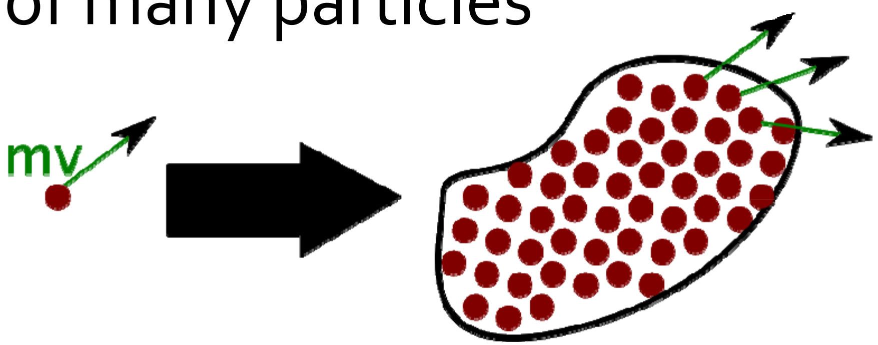
Why Study Dynamics (2)

- http://www.youtube.com/watch?v=W1czBcnX1Ww



Strategy

- Goal: develop equations for motion of rigid bodies
- Derive equations for particles first
  - Simpler physics than rigid bodies
  - Less degrees of freedom
- Generalize to rigid bodies
  - Rigid body = collection of many particles



Notations

- Vector representation: $\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$ $\vec{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$

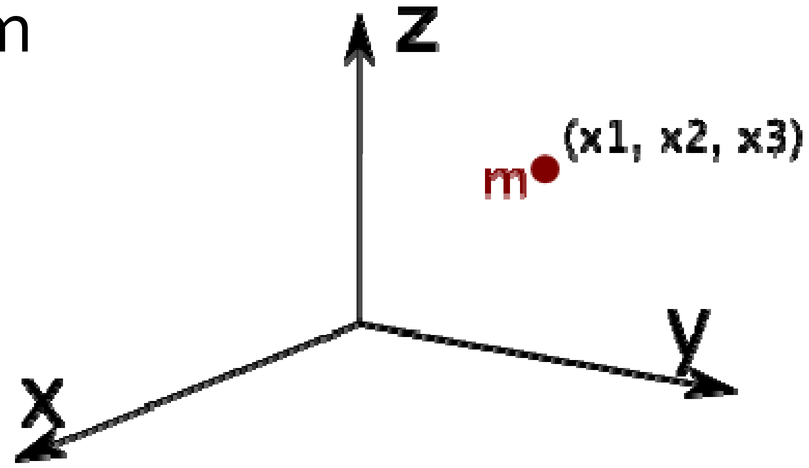
- Time derivative: $\dfrac{dx}{dt} => \dot{x}$ $\dfrac{d^2x}{d^2t} => \ddot{x}$

- Phase space (Spatial vector):

$$\begin{bmatrix} x \\ v \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} \qquad \begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v \\ \dfrac{f}{m} \end{bmatrix}$$

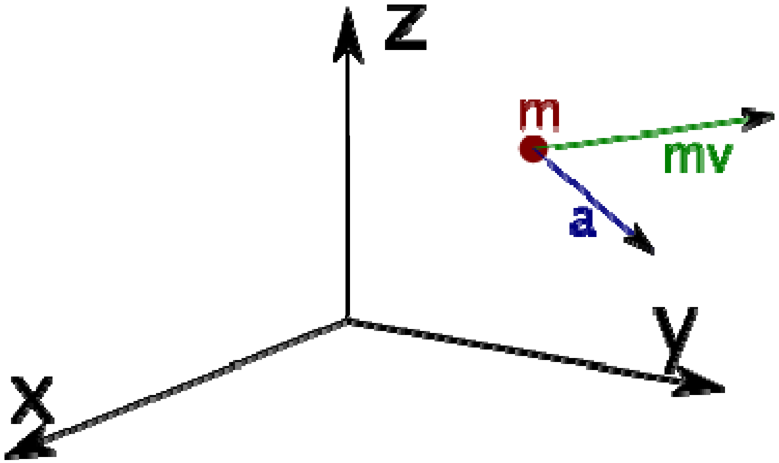
The Particle

- The particle is a point mass
  - Position vector (x1, x2, x3)
  - Mass m
  - 3 degrees of freedom



Physical Quantities of the Particle

- Mass
- Linear Velocity
- Linear Acceleration
- Linear momentum
- Does not have
  - Volume
  - Angular characteristics



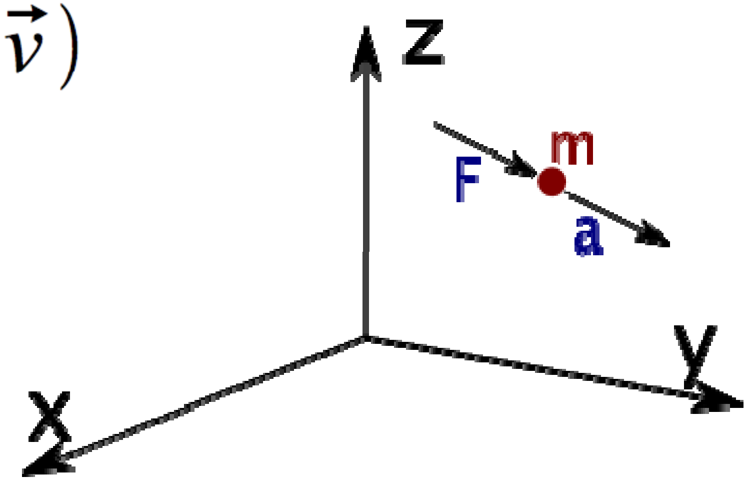
Newton's Law (Particle)

- Force applied to particle equals mass times acceleration

$$\vec{p} = m\vec{v}$$

$$\frac{d}{dt}(\vec{p}) = \frac{d}{dt}(m\vec{v}) = m\frac{d}{dt}(\vec{v})$$

$$\frac{d}{dt}(\vec{p}) = m\vec{a} = \vec{F}$$



Angular Momentum (Particle About an Origin)

$\vec{r} = position\ vector\ relative\ to\ origin$

$\vec{L} = angular\ momentum$

$\vec{\tau} = torque$

$$\vec{L} = \vec{r} \times (m\vec{v})$$

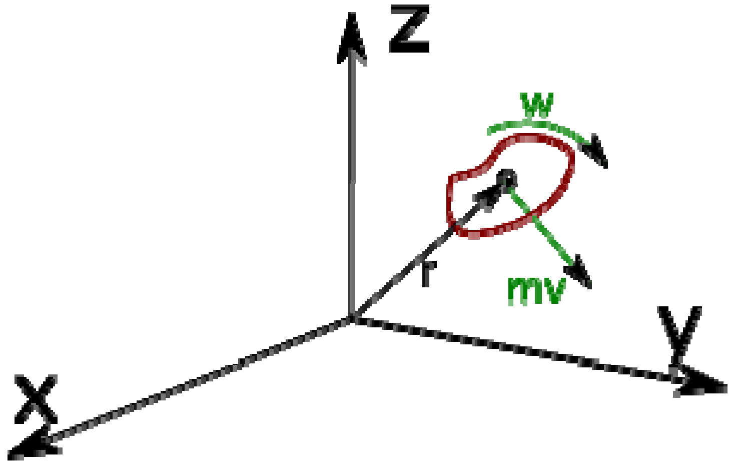
$$\vec{\tau} = \vec{r} \times \vec{F} = \frac{d}{dt}(\vec{r} \times m\vec{v})$$

$$\vec{\tau} = \frac{d}{dt}(\vec{r} \times m\vec{v}) = \vec{r} \times m\frac{d}{dt}(\vec{v}) + \boxed{\vec{v} \times m\vec{v}} = \vec{r} \times m\frac{d}{dt}(\vec{v})$$

= 0

Rigid Body

- A collection of many particles
- We use the center of mass for position and center of rotation
- 6 degrees of freedom
  - 3 position
  - 3 angular



Angular Momentum (Rigid Body) (1)

- Angular momentum is the sum over all particles that make up the rigid body

$$\vec{L} = \sum_i \vec{r}_i \times m_i \vec{v}_i$$

$$\vec{L} = \sum_i m_i \vec{r}_i \times (\omega \times \vec{r}_i)$$

$$m_i = \rho\, dv$$

$$\vec{L} = \int_V \vec{r} \times (\omega \times \vec{r}) \rho\, dV$$

$m_i = mass\ of\ i^{th}\ particle$

$\rho = density\ of\ rigid\ body$

$\vec{v}_i = \omega \times \vec{r}_i$

Angular Momentum (Rigid Body) (2)

$$\vec{L}=\int_V \vec{r}\times(\omega\times\vec{r})\rho\,dV$$

$$\vec{r}\times(\omega\times\vec{r})=\vec{r}\times(-\vec{r})\times\omega$$

$$\vec{L}=\omega\boxed{\int_V -r\times r\,\rho\,dV}$$

Integral is called the inertia tensor

$$\vec{L}=I\,\omega$$

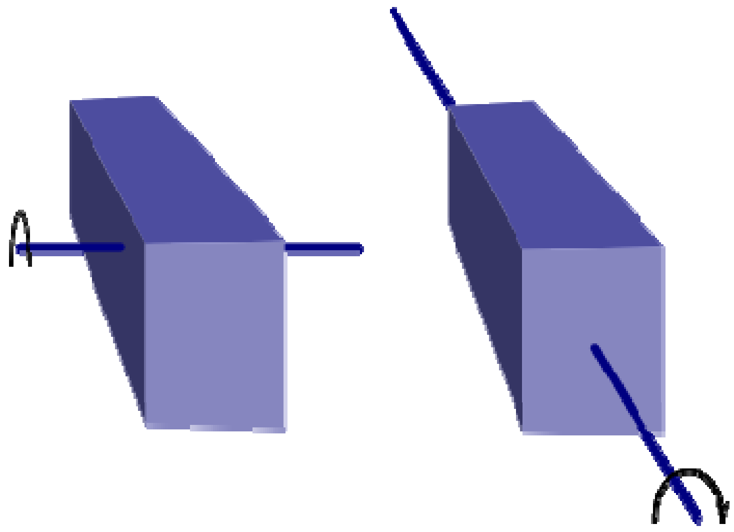
$$\vec{\tau}=\frac{d}{dt}(I\,\omega)=I\frac{d}{dt}(\omega)$$

Inertia Tensor (1)

- Inertia is rotation analog of mass
- In 2D, every rotation axis is parallel
- In 3D, objects can rotate about arbitrary axes
- The inertia tensor is used to find inertia for an arbitrary axis:

$$\text{Given axis of rotation } \vec{n} = \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix}$$

$$I = n^T I n$$



Inertia Tensor (2)

$$\boldsymbol{I} = \begin{pmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{pmatrix}$$

$$I_{xx} = \int\int\int (y^2 + z^2)\, \rho\, dx\, dy\, dz$$
$$I_{yy} = \int\int\int (x^2 + z^2)\, \rho\, dx\, dy\, dz$$
$$I_{zz} = \int\int\int (x^2 + y^2)\, \rho\, dx\, dy\, dz$$
$$I_{xy} = \int\int\int xy\, \rho\, dx\, dy\, dz$$
$$I_{xz} = \int\int\int xz\, \rho\, dx\, dy\, dz$$
$$I_{yz} = \int\int\int yz\, \rho\, dx\, dy\, dz$$

Summary (Rigid Body)

$$\vec{p} = m\vec{v}$$

$$\vec{F} = m\vec{a} = \frac{d}{dt}(\vec{p})$$

$$\vec{L} = I\omega$$

$$\vec{\tau} = \vec{r} \times \vec{F} = \vec{r} \times \frac{d}{dt}\vec{L}$$

$m = mass$

$\vec{v} = linear\ velocity$

$\vec{a} = linear\ acceleration$

$\vec{p} = linear\ momentum$

$\vec{F} = net\ force$

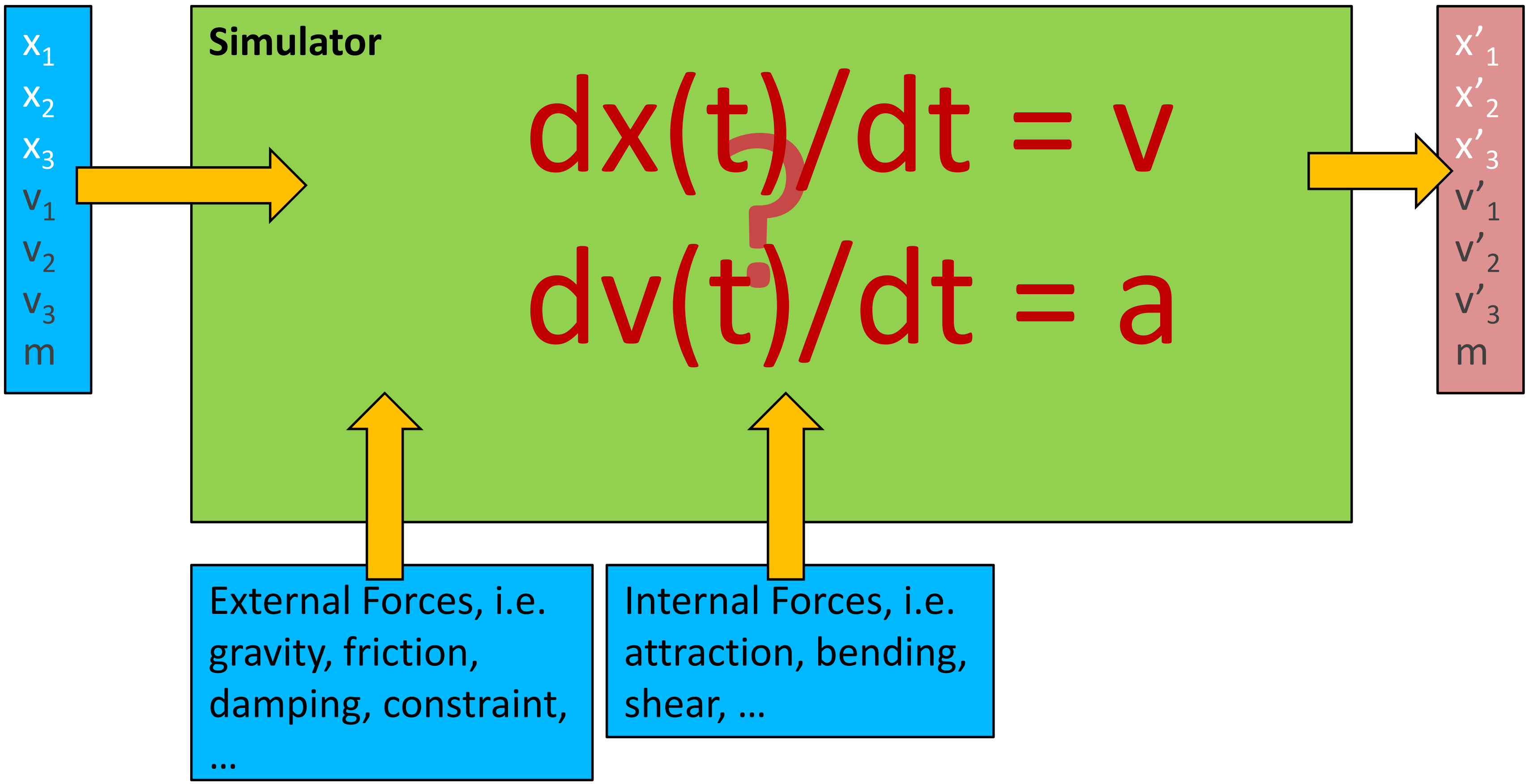
$\vec{L} = angular\ momentum$

$\omega = angular\ velocity$

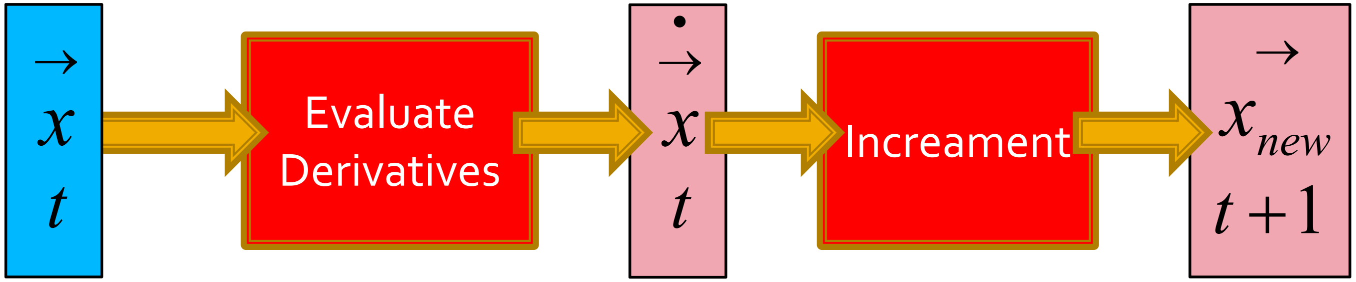
$\vec{\tau} = torque$

$\vec{r} = center\ of\ mass$

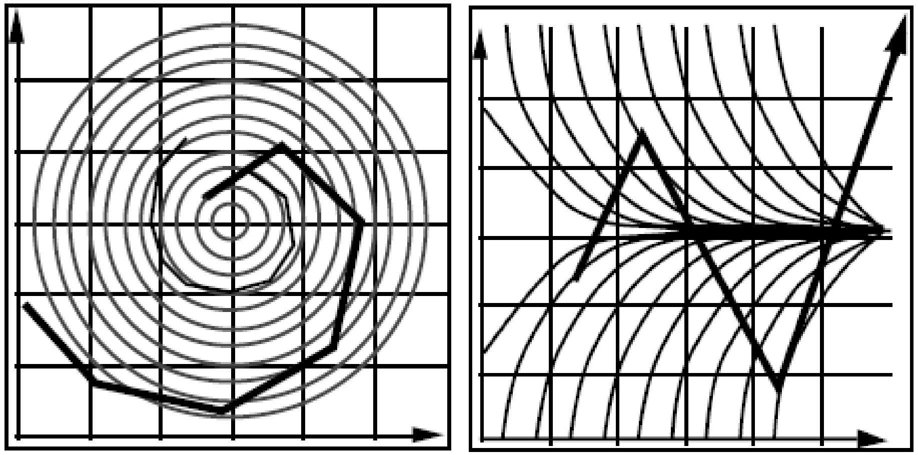
Simulation Loop



Numerical Integration (1)

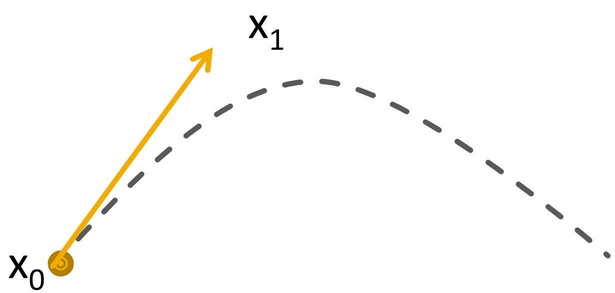
- Idea: 

- Evaluation:
  - Stability
  - Error
- Case study:
  - Euler's method
  - Mid point method
  - Adaptive method



Time Integration (2)

- Euler's method (1st order):

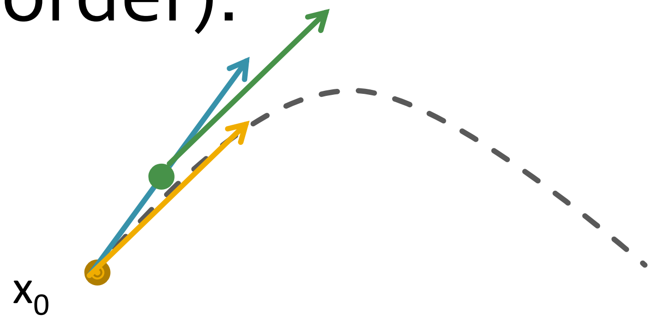


$x_1 = x_0 + f(x_{0\,,}\ t_0)*h,$

h: step size

f(x, t): first derivative at x

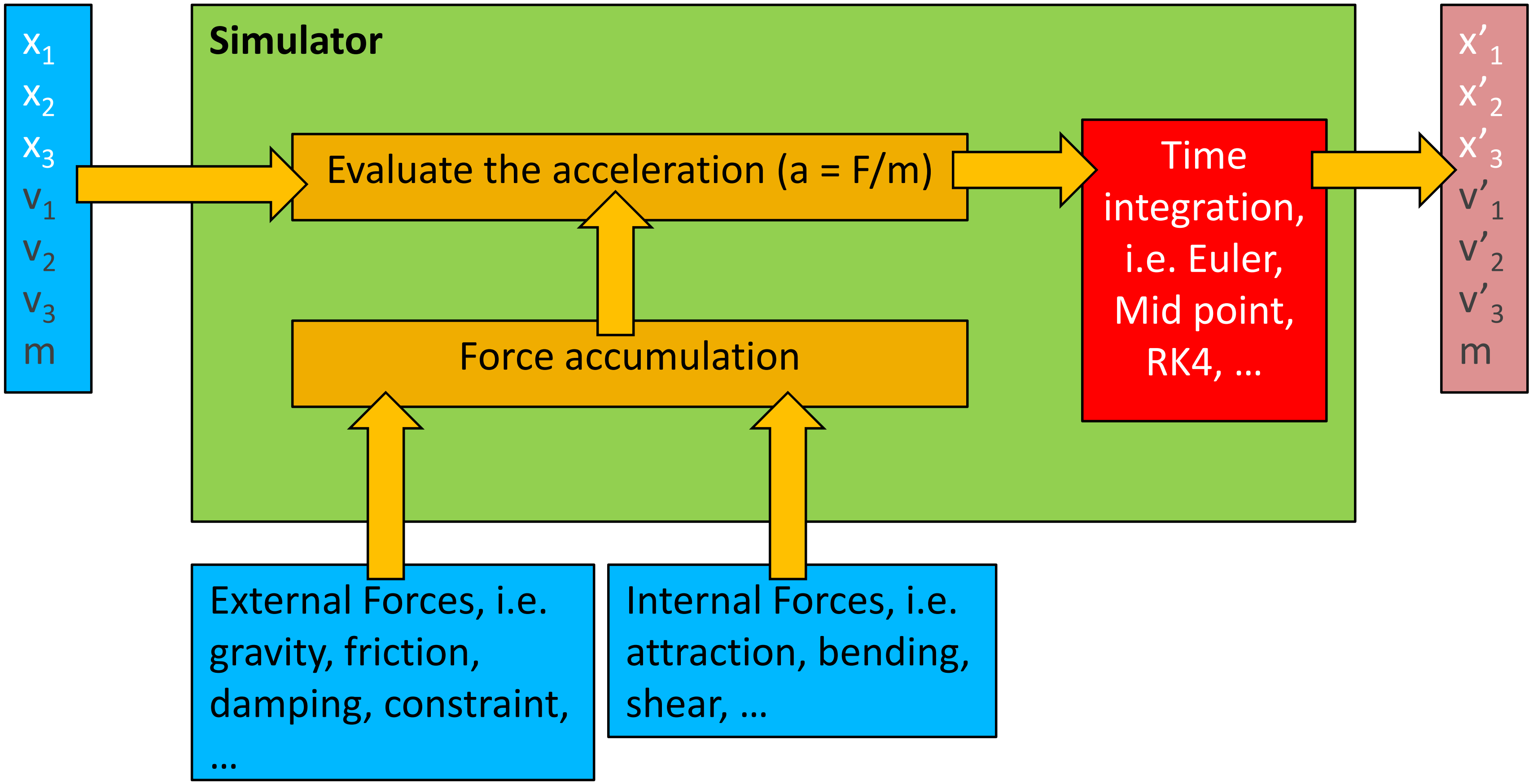
- Mid point method (Runge-Kutta 2, 2nd order):



$\Delta x = f(x_{0\,,}\ t_0)*h$

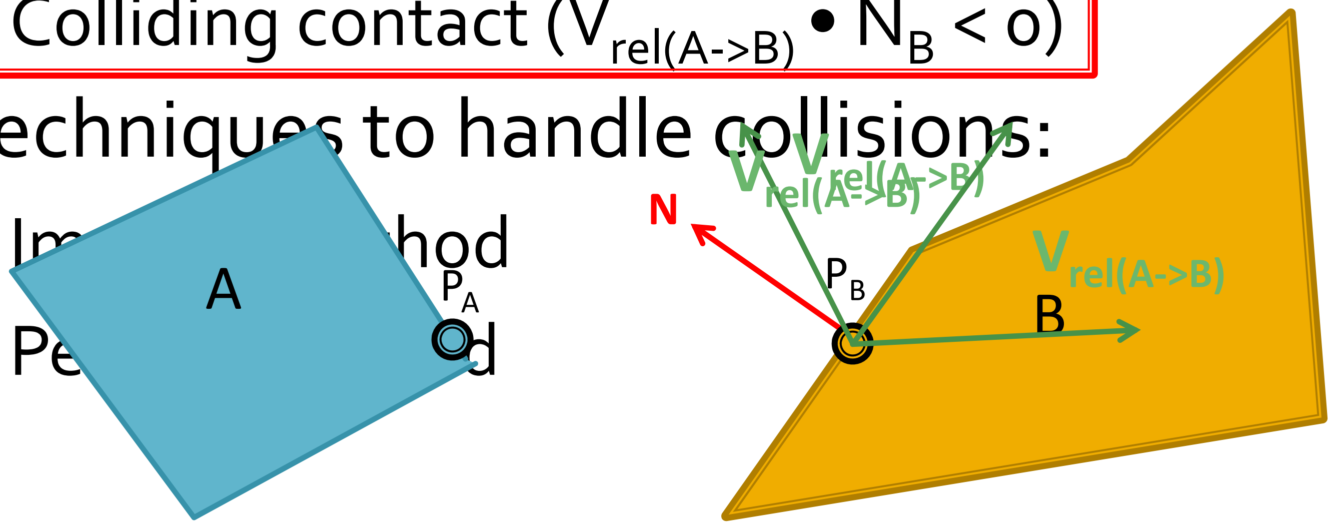
$x_1 = x_0 + f(x_0+\Delta x/2_{\,,}\ t_0 + h/2)*h$

New Simulation Loop



Collision Handling (1)

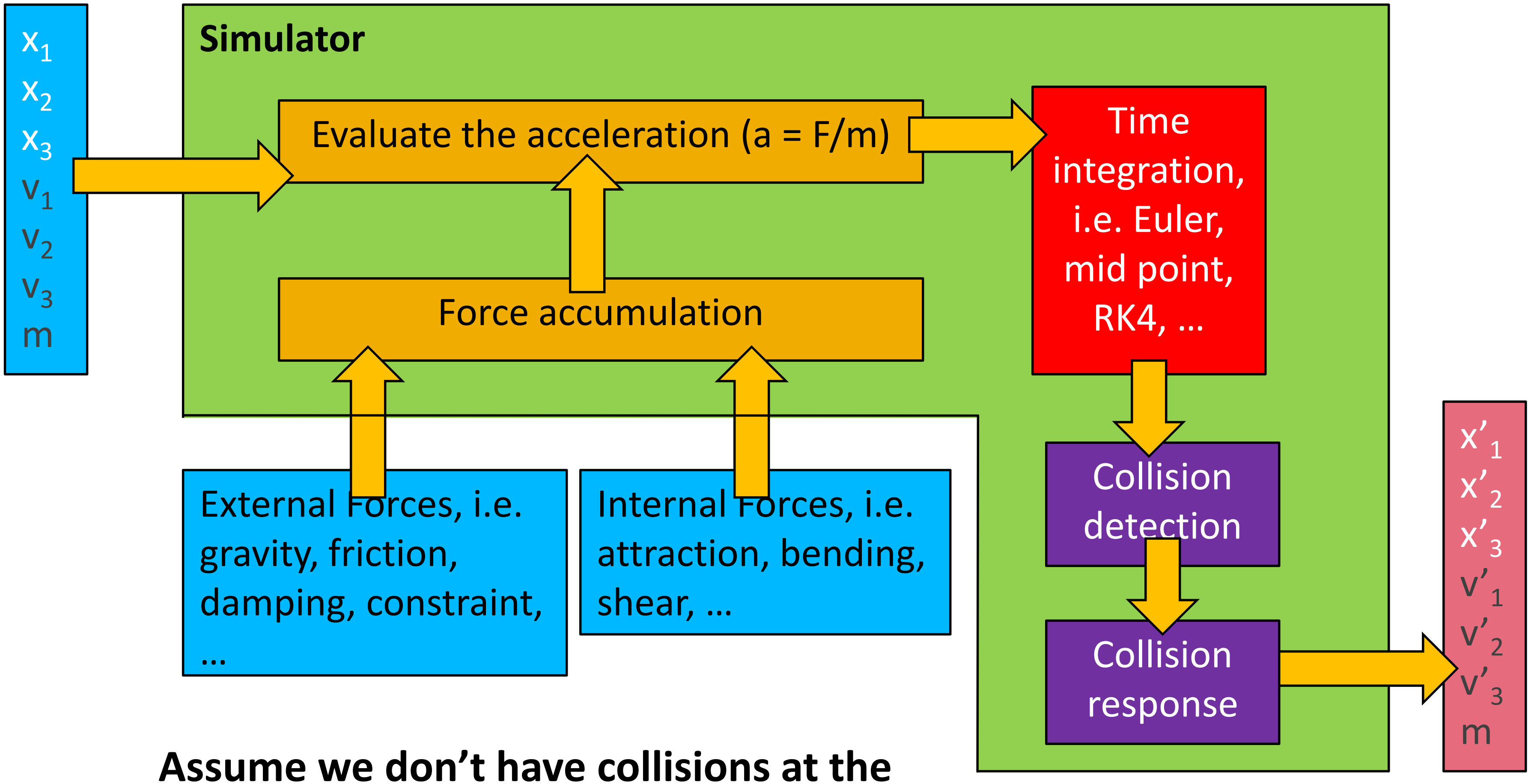
- How to detect collisions
- Types of colliding contact:
  - Separation ($V_{rel(A\text{->}B)} \bullet N_B > 0$)
  - Resting contact ($V_{rel(A\text{->}B)} \bullet N_B = 0$)
  - Colliding contact ($V_{rel(A\text{->}B)} \bullet N_B < 0$)
- Techniques to handle collisions:
  - Im … hod
  - Pe … d



Collision Handling (2)

- Impulse method (for colliding contact)
  - Instantaneously change the velocity at the contacting point.
- Penalty method
  - Use spring forces to pull penetrated objects out.
- How to deal with resting contact?
  - Compute and apply repulsion forces for all the contacting point to maintain the equilibrium.
  - Need to solve a quadratic programming problem.

Simulation Loop with Collision Handling



Assume we don't have collisions at the beginning…

Conclusion

- Why understanding rigid body is important for robotics
- From particles to rigid bodies
- The properties of the rigid body
- The motion of the rigid body
- Numerical integration
- How to handle collisions
- Rigid body simulation framework

References

- Baraff & Witkin, *"An Introduction to Physically Based Modeling"* course notes (Rigid Body Dynamics I and II, Differential Equation Basics, and Particle Dynamics), SIGGRAPH2001
- Stanford Univeristy, "Lecture 11 Introduction to Robotics", http://www.youtube.com/watch?v=03Xx3vi6qzo
- Professor Karen Liu's slides from the Computer Animation class at Georgia Tech http://web.mac.com/ckarenliu/CS4496/Calendar.html