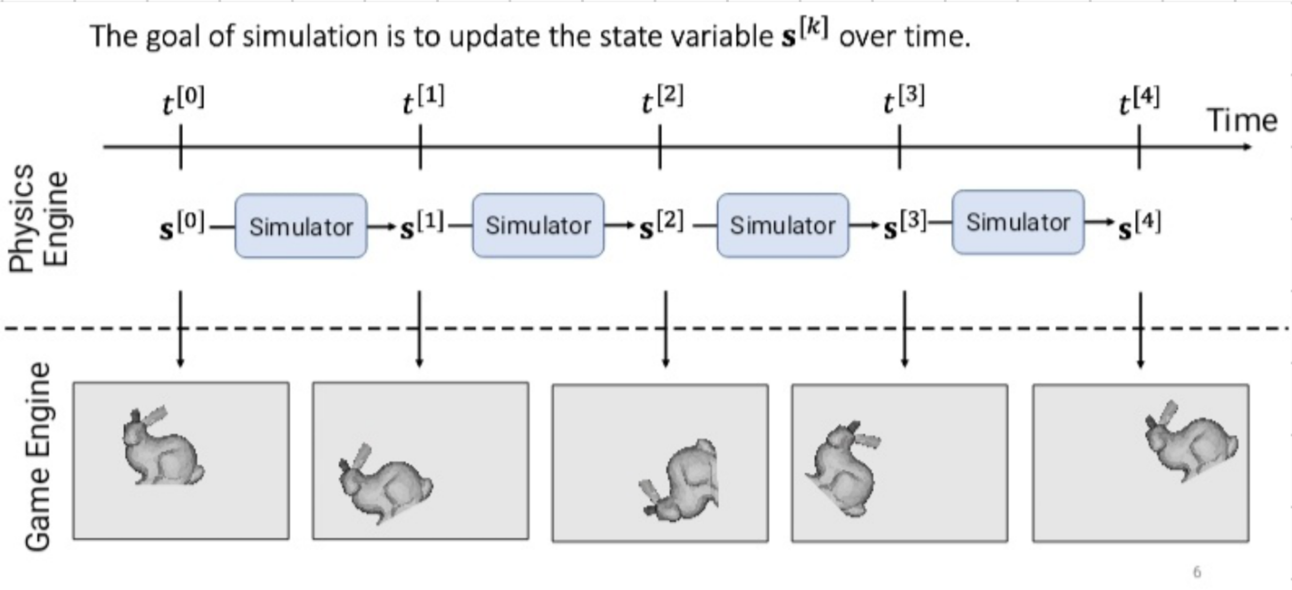
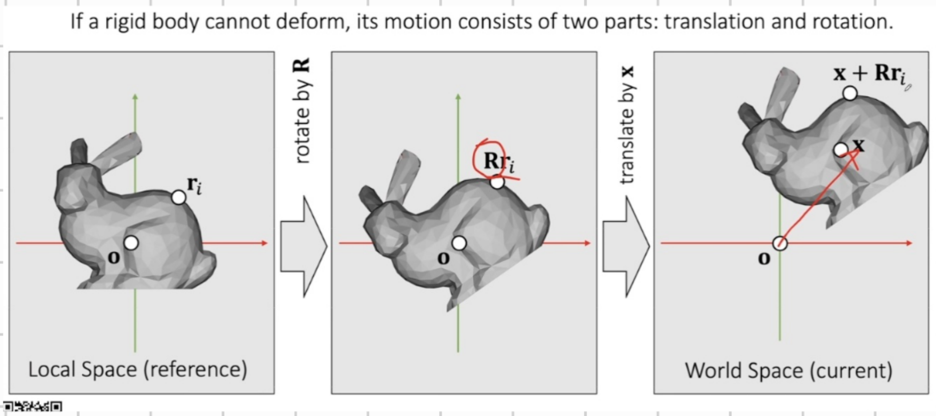
Rigid Body Simulation 刚体模拟

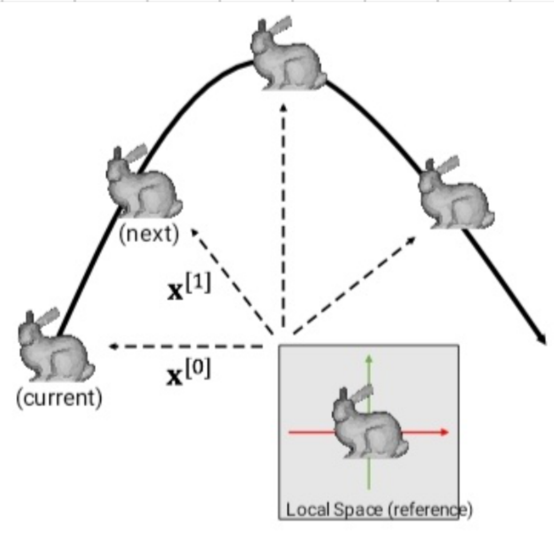


Rigid Body Motion 刚体运动

通过平移和旋转来改变刚体状态



Translational Motion 平移运动



对于平移运动，状态变量包含 位置x和速度v

① $v(t^{[1]}) = v(t^{[0]}) + M^{-1}\int_{t^{[0]}}^{t^{[1]}} f(x(t), v(t), t)\,dt$

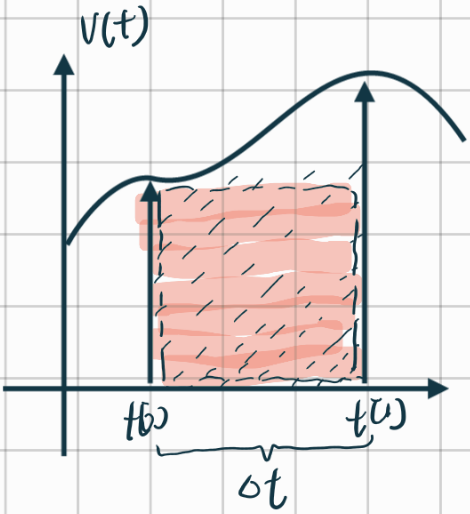
$$v = \frac{dx}{dt} = \int a\,dt = \frac{\int f\,dt}{M}$$

速度 = 力对时间的积分

② $x(t^{[1]}) = x(t^{[0]}) + \int_{t^{[0]}}^{t^{[1]}} v(t)\,dt$

Integration Methods Explained 积分方法

显式积分 1阶正确



求 $x(t)=\int v(t)\,dt$ 就是求面积，可以近似地将其看做一个box

$$\int_{t^{[0]}}^{t^{[1]}} v(t)\,dt \approx \Delta t\, v(t^{[0]})$$

$$\int_{t^{[0]}}^{t^{[1]}} v(t)\,dt = \Delta t\, v(t^{[0]}) + \frac{\Delta t^2}{2} v'(t^{[0]}) + \dots$$

$$= \Delta t\, v(t^{[0]}) + O(\Delta t^2)$$

error

1阶准确

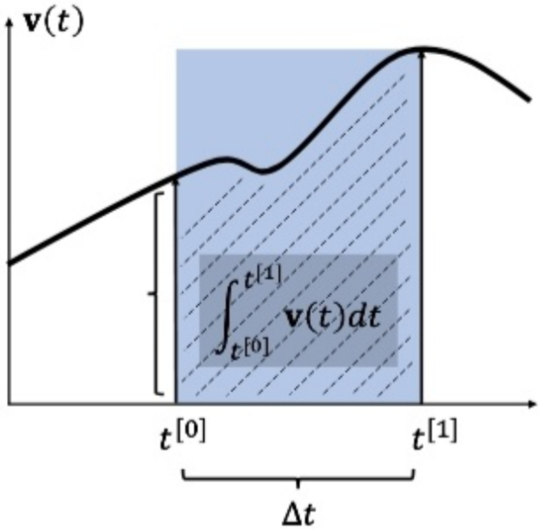
By definition, the integral $\mathbf{x}(t) = \int \mathbf{v}(t)dt$ is the area. Many methods estimate the area as a box.

Implicit Euler (1st-order accurate) sets the height at $t^{[1]}$.

$$\int_{t^{[0]}}^{t^{[1]}} \mathbf{v}(t)dt \approx \underbrace{\Delta t}_{\text{width}} \, \underbrace{\mathbf{v}\left(t^{[1]}\right)}_{\text{height}}$$

$$\int_{t^{[0]}}^{t^{[1]}} \mathbf{v}(t)dt = \Delta t \, \mathbf{v}\left(t^{[1]}\right) - \frac{\Delta t^2}{2}\mathbf{v}'\left(t^{[1]}\right) + \cdots$$

$$= \Delta t \, \mathbf{v}\left(t^{[1]}\right) + \underbrace{\boxed{O(\Delta t^2)}}_{\text{error}}$$



二阶准确

By definition, the integral $\mathbf{x}(t) = \int \mathbf{v}(t)dt$ is the area. Many methods estimate the area as a box.

Mid-point (2nd-order accurate) sets the height at $t^{[0.5]}$.

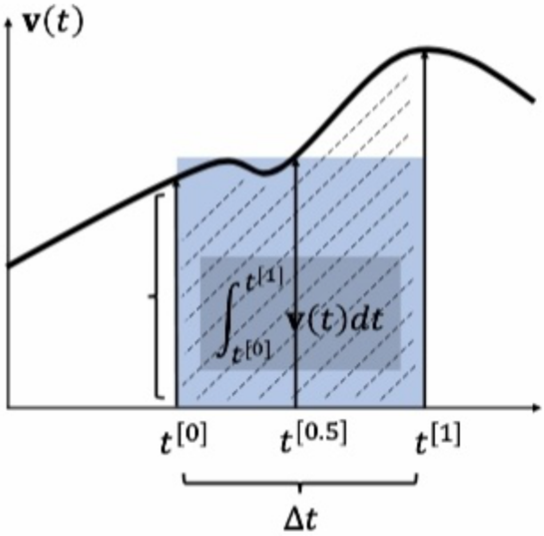
$$\int_{t^{[0]}}^{t^{[1]}} \mathbf{v}(t)dt \approx \underbrace{\Delta t}_{\text{width}} \, \underbrace{\mathbf{v}\left(t^{[0.5]}\right)}_{\text{height}}$$

$$\int_{t^{[0]}}^{t^{[1]}} \mathbf{v}(t)dt = \int_{t^{[0]}}^{t^{[0.5]}} \mathbf{v}(t)dt + \int_{t^{[0.5]}}^{t^{[1]}} \mathbf{v}(t)dt$$

$$= \frac{1}{2}\Delta t \, \mathbf{v}\left(t^{[0.5]}\right) - \frac{\Delta t^2}{2}\mathbf{v}'\left(t^{[0.5]}\right) + O(\Delta t^3) +$$

$$\frac{1}{2}\Delta t \, \mathbf{v}\left(t^{[0.5]}\right) + \frac{\Delta t^2}{2}\mathbf{v}'\left(t^{[0.5]}\right) + O(\Delta t^3)$$

$$= \Delta t \, \mathbf{v}\left(t^{[0.5]}\right) + \underbrace{\boxed{O(\Delta t^3)}}_{\text{error}}$$



By definition, the integral $\mathbf{x}(t) = \int \mathbf{v}(t)dt$ is the area. Many methods estimate the area as a box.

Explicit Euler (1st-order accurate) sets the height at $t^{[0]}$.

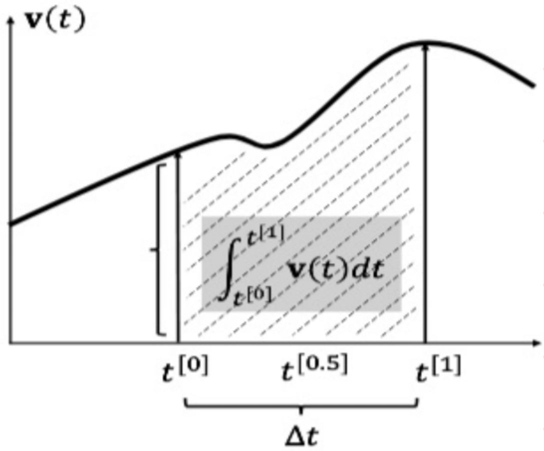
$$\int_{t^{[0]}}^{t^{[1]}} \mathbf{v}(t)dt \approx \Delta t \, \mathbf{v}\left(t^{[0]}\right)$$

Implicit Euler (1st-order accurate) sets the height at $t^{[1]}$.

$$\int_{t^{[0]}}^{t^{[1]}} \mathbf{v}(t)dt \approx \Delta t \, \mathbf{v}\left(t^{[1]}\right)$$

Mid-point (2nd-order accurate) sets the height at $t^{[0.5]}$.

$$\int_{t^{[0]}}^{t^{[1]}} \mathbf{v}(t)dt \approx \Delta t \, \mathbf{v}\left(t^{[0.5]}\right)$$



回到之前我们需要对时间进行积分

For translational motion, the state variable contains the position $\mathbf{x}$ and the velocity $\mathbf{v}$.

$$\begin{cases} \mathbf{v}(t^{[1]}) = \mathbf{v}(t^{[0]}) + M^{-1}\int_{t^{[0]}}^{t^{[1]}} \mathbf{f}(\mathbf{x}(t), \mathbf{v}(t), t)dt \\ \mathbf{x}(t^{[1]}) = \mathbf{x}(t^{[0]}) + \int_{t^{[0]}}^{t^{[1]}} \mathbf{v}(t)dt \end{cases}$$

$\Downarrow$ 积分

$v^{[1]} = v^{[0]} + \Delta t M^{-1} f^{[0]}$ Explicit 显式

$x^{[1]} = x^{[0]} + \Delta t v^{[1]}$ Implicit 隐式

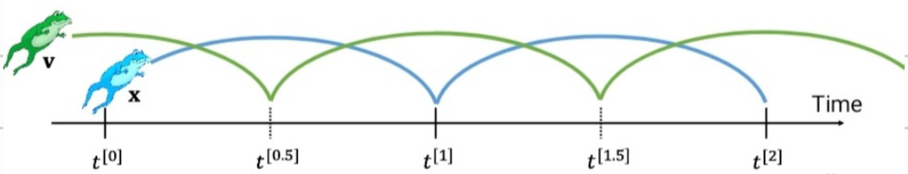
Leapfrog Integration 跳蛙积分

In some literature, such a approach is called *semi-implicit*.

$$\begin{cases} \mathbf{v}^{[1]} = \mathbf{v}^{[0]} + \Delta t M^{-1}\mathbf{f}^{[0]} & \leftarrow \text{Explicit} \\ \mathbf{x}^{[1]} = \mathbf{x}^{[0]} + \Delta t \mathbf{v}^{[1]} & \leftarrow \text{Implicit} \end{cases}$$

It has a funnier name: the *leapfrog method*.

$$\begin{cases} \mathbf{v}^{[0.5]} = \mathbf{v}^{[-0.5]} + \Delta t M^{-1}\mathbf{f}^{[0]} & \leftarrow \text{Mid-point} \\ \mathbf{x}^{[1]} = \mathbf{x}^{[0]} + \Delta t \mathbf{v}^{[0.5]} & \leftarrow \text{Mid-point} \end{cases}$$



16

Types of Forces 力的类型

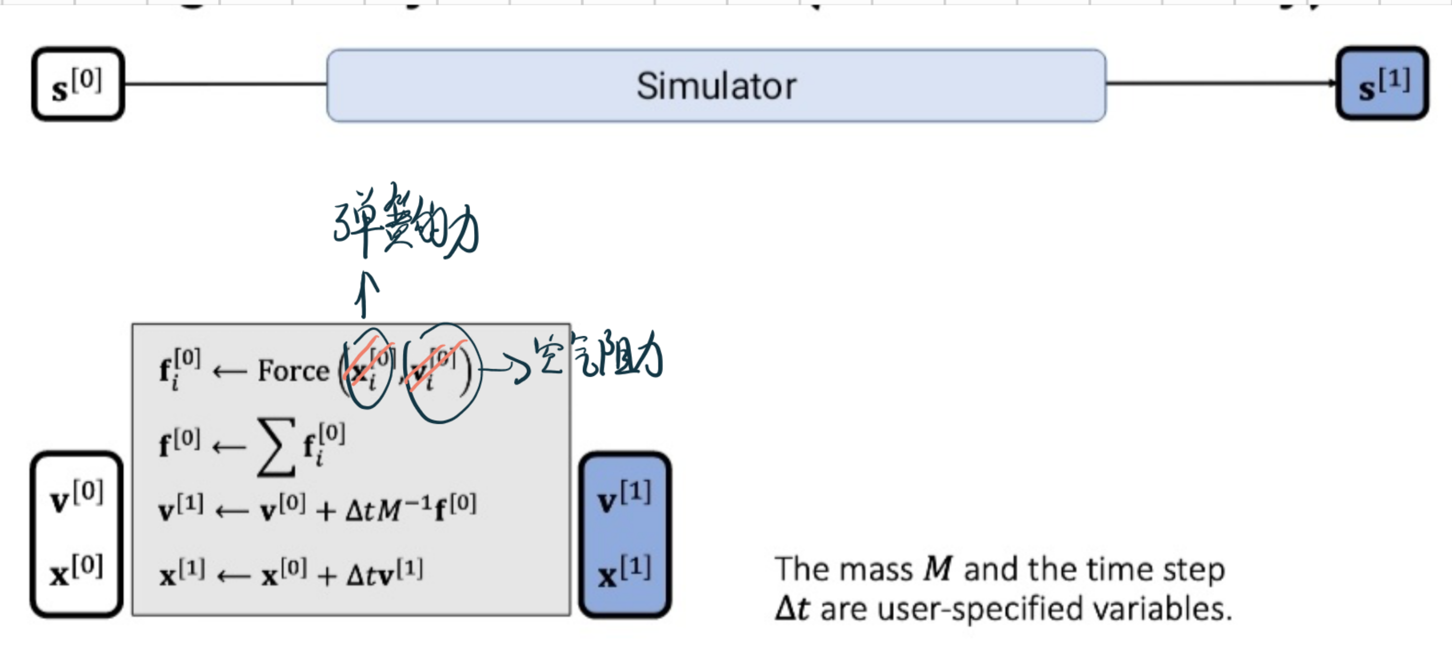
重力 $f^{[0]}_{gravity} = \underset{\text{mass}}{M}\,\overset{\text{gravity}}{g}$

空气阻力 $f^{[0]}_{drag} = -\underset{\text{drag coefficient}}{\sigma}\, v^{[0]}$ → velocity

更通用的方式计算空气阻力

$v^{[1]} = \alpha v^{[0]}$

Rigid Body Simulation 刚体模拟



Rotation Represented by Matrix 旋转的矩阵表示

$$R = \begin{bmatrix} r_{00} & r_{01} & r_{02} \\ r_{10} & r_{11} & r_{12} \\ r_{20} & r_{21} & r_{22} \end{bmatrix}$$

9个值但 旋转只有3个自由度

不直观

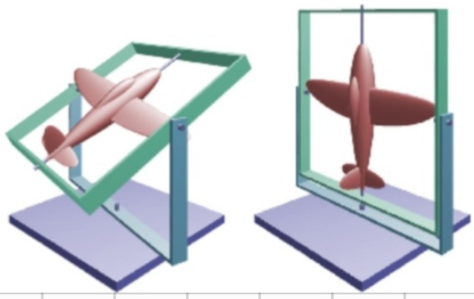
Rotation Represented by Euler Angles 旋转的欧拉角表示

- The Euler Angles representation is also popular, often in design and control.
- It is intuitive. It uses three axial rotations to represent one general rotation. Each axial rotation uses an angle.
- In Unity, the order is rotation-by-Z, rotation-by-X, then rotation-by-Y.

- But it is not suitable for dynamics either:
  - It can lose DoFs in certain statuses: *gimbal lock*.
  - Defining its time derivative (*rotational velocity*) is difficu

万向锁

The alignment of two or more axes results in a loss of rotational DoFs.



22

Rotation Represented by Quaternion 用四元数表示旋转

Complex multiplications

	1	i
1	1	i
i	i	−1

Quaternion multiplications

	1	i	j	k
1	1	i	j	k
i	i	−1	k	−j
j	j	−k	−1	i
k	k	j	−i	−1

用四个点表示三维向量

Quaternion Arithematic

$$q = [\underset{\uparrow\, \mathbb{R}}{s} \quad \underset{\uparrow\, \text{向量}}{\vec{v}}]$$

$$aq = [as \quad a\vec{v}]$$

$$q_1 \pm q_2 = [s_1 \pm s_2 \quad \vec{v}_1 \pm \vec{v}_2]$$

$$q_1 \times q_2 = [s_1 s_2 - \vec{v}_1 \cdot \vec{v}_2 \quad s_1\vec{v}_2 + s_2\vec{v}_1 + \vec{v}_1 \times \vec{v}_2]$$

$$\|q\| = \sqrt{s^2 + v \cdot v}$$

$$\begin{cases} q = [\cos\frac{\theta}{2} \quad \vec{v}] \\ \|q\| = 1 \end{cases} \Rightarrow \quad q = [\cos\frac{\theta}{2} \quad \vec{v}], \quad \|v\|^2 = \sin^2\frac{\theta}{2}$$

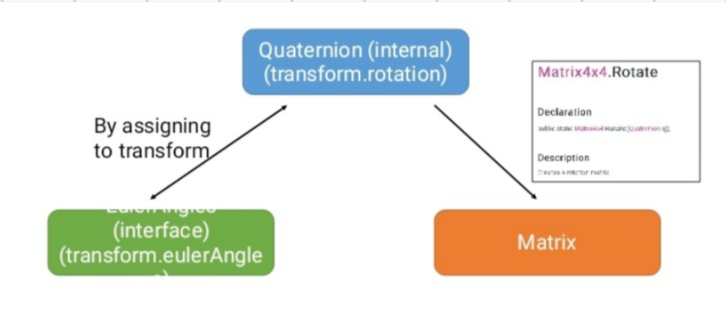
绕一个轴，通过θ角的旋转

罗德里格斯旋转公式

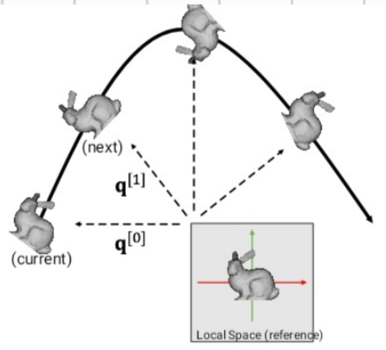
- Convertible to the matrix:

$$\mathbf{R} = \begin{bmatrix} s^2 + x^2 - y^2 - z^2 & 2(xy - sz) & 2(xz + sy) \\ 2(xy + sz) & s^2 - x^2 + y^2 - z^2 & 2(yz - sx) \\ 2(xz - sy) & 2(yz + sx) & s^2 - x^2 - y^2 + z^2 \end{bmatrix}$$

Rotation Representations in Unity



Rotation Motion 旋转运动



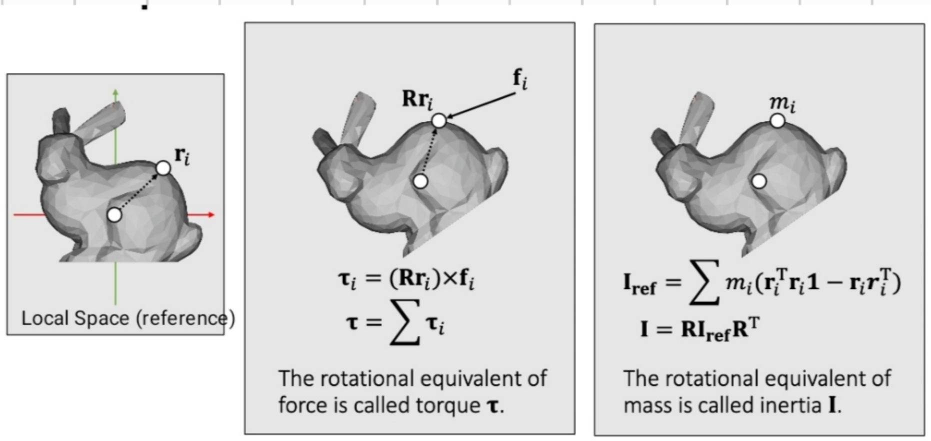
角速度

ω

$\|\omega\|$

ω 方向

$\|\omega\|$ 速度



Translation and Rotational Motion

Translation (liner)

$$v^{[1]} = v^{[0]} + \Delta t M^{-1} f^{[0]}$$
$$x^{[1]} = x^{[0]} + \Delta t v^{[1]}$$

Rotational (Angual)

$$\begin{cases} w^{[1]} = w^{[0]} + \Delta t (I^{[0]})^{-1} \tau^{[0]} \\ q^{[1]} = q^{[0]} + \left[0 \ \ \frac{\Delta t}{2} w^{[1]}\right] \times q^{[0]} \end{cases}$$