Rodrigues Formula
0x03

DY Kim

Dept. of Artificial Intelligence, Ajou University

Contents

Previous

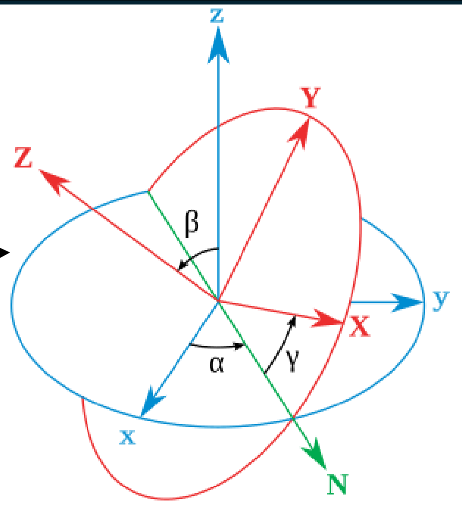
Orientation & Rotation

There are many ways to describe the rotation

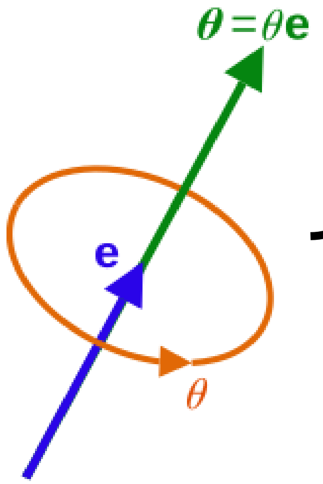
1. Rotation Matrix
2. Euler angles
3. Axis-angle
4. Rodrigues Formula
5. Rotation Vector
6. Unit Quaternion
   (Euler Parameters)

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

<1>



<2>



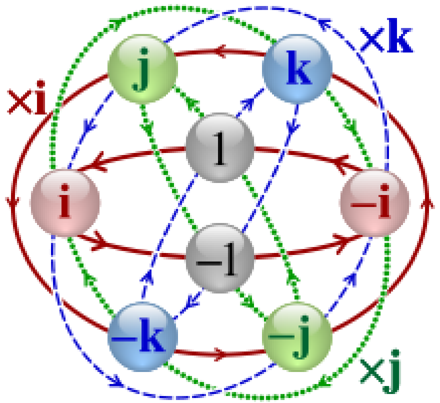
<3>

$$\mathbf{v}_{\mathrm{rot}} = \mathbf{v} + (\sin\theta)(\mathbf{e}\times\mathbf{v}) + (1-\cos\theta)(\mathbf{e}\times(\mathbf{e}\times\mathbf{v}))$$

<4>

$$\begin{bmatrix} 0 \\ 0 \\ \frac{\pi}{2} \end{bmatrix}$$

<5>



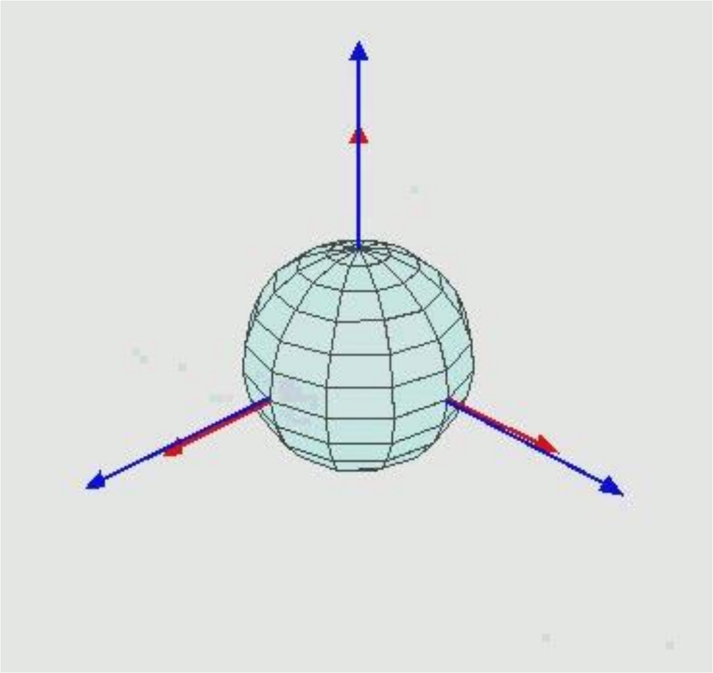
<6>

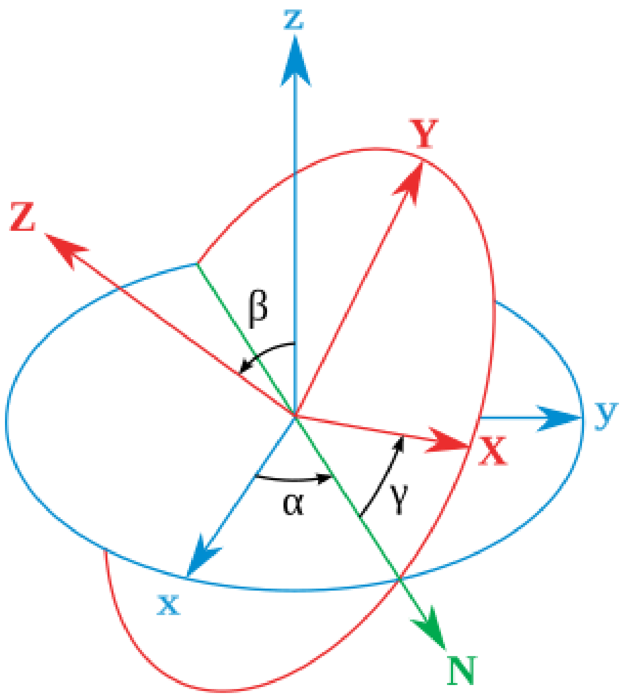
https://en.wikipedia.org/wiki/

Euler Angles

Euler Angles

The Euler Angles are three angles introduced by Leonhard Euler
to describe the **orientation** of a rigid body with respect to a fixed coordinate system.

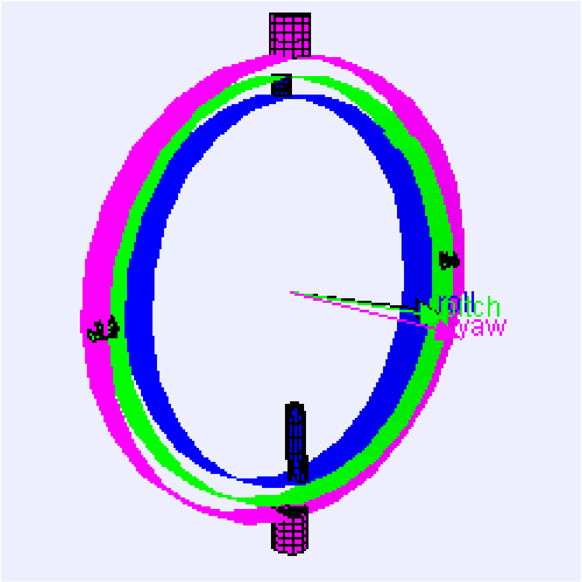


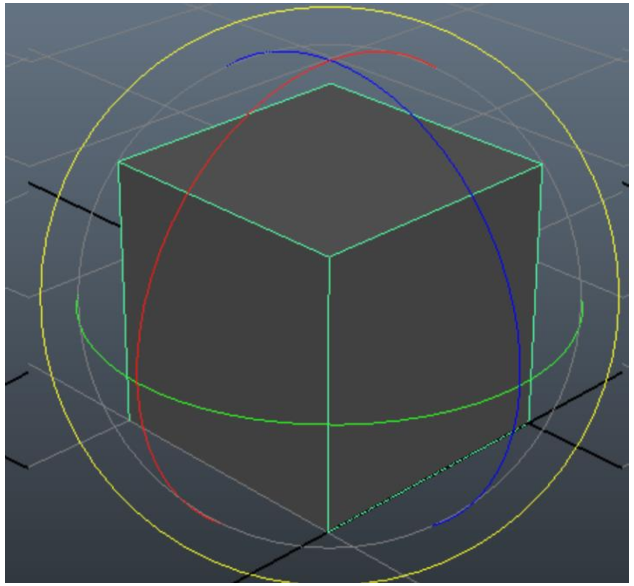


Euler Angles

Gimbal

- Hardware implementation of Euler angles
- Camera, airplane, maya, etc

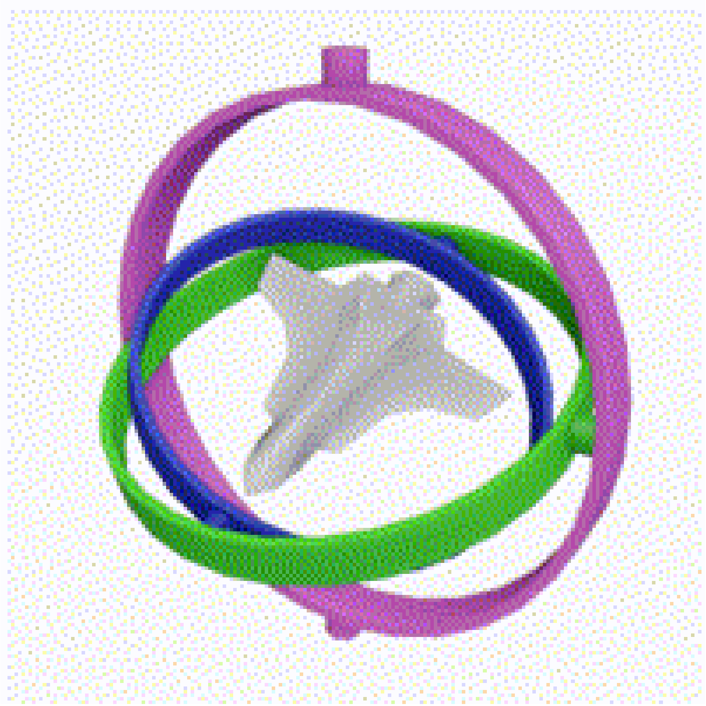
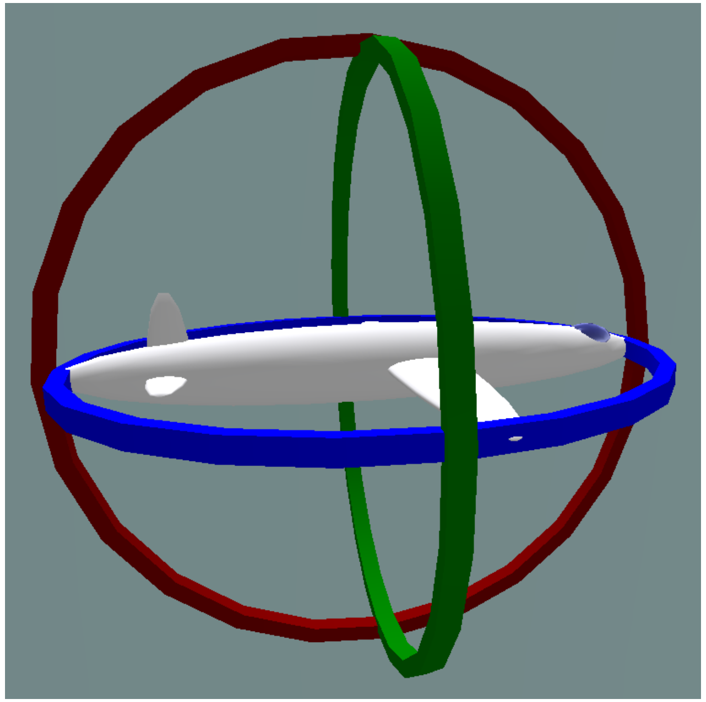
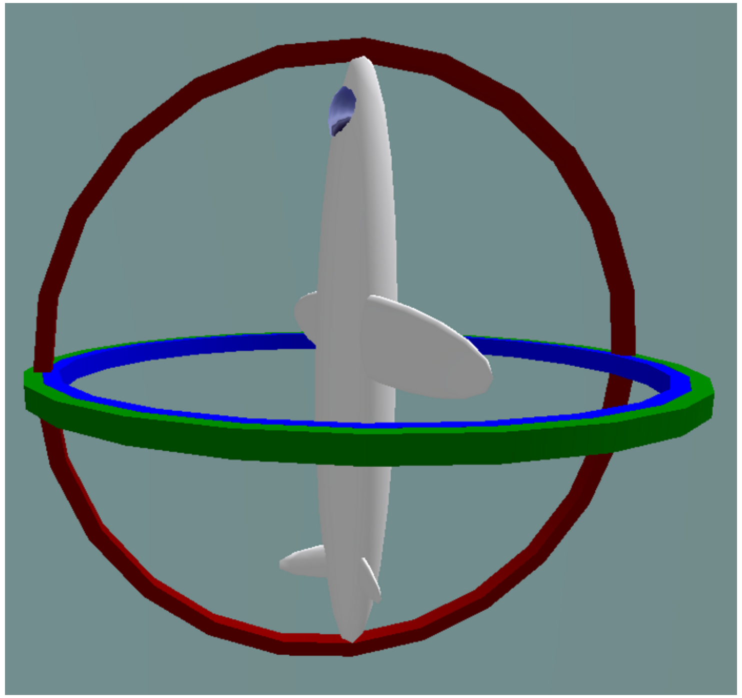




Euler Angles

Gimbal Lock

Gimbal lock is the loss of the one degree of freedom at certain alignments of the axes.

https://www.youtube.com/watch?v=zc8b2Jo7mno&t=193s

Axis-angles

Axis–angle representation

Article Talk Read E

From Wikipedia, the free encyclopedia

For broader coverage of this topic, see 3D rotation group.

In mathematics, the **axis-angle representation** parameterizes a rotation in a three-dimensional Euclidean space by two quantities: a unit vector $\mathbf{e}$ indicating the direction of an axis of rotation, and an angle of rotation θ describing the magnitude and sense (e.g., clockwise) of the rotation about the axis. Only two numbers, not three, are needed to define the direction of a unit vector $\mathbf{e}$ rooted at the origin because the magnitude of $\mathbf{e}$ is constrained. For example, the elevation and azimuth angles of $\mathbf{e}$ suffice to locate it in any particular Cartesian coordinate frame.

Rotation := Axis vector + Angle

$$(\text{axis}, \text{angle}) = \left(\begin{bmatrix} e_x \\ e_y \\ e_z \end{bmatrix}, \theta \right)$$

e.g.) X-axis, 90 degrees

$$\left(\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \pi/2 \right) \longrightarrow \begin{bmatrix} \pi/2 \\ 0 \\ 0 \end{bmatrix}$$

Rotation Vector

Axis-angles

e.g.) OpenGL

glRotate

glRotate — multiply the current matrix by a rotation matrix

C Specification

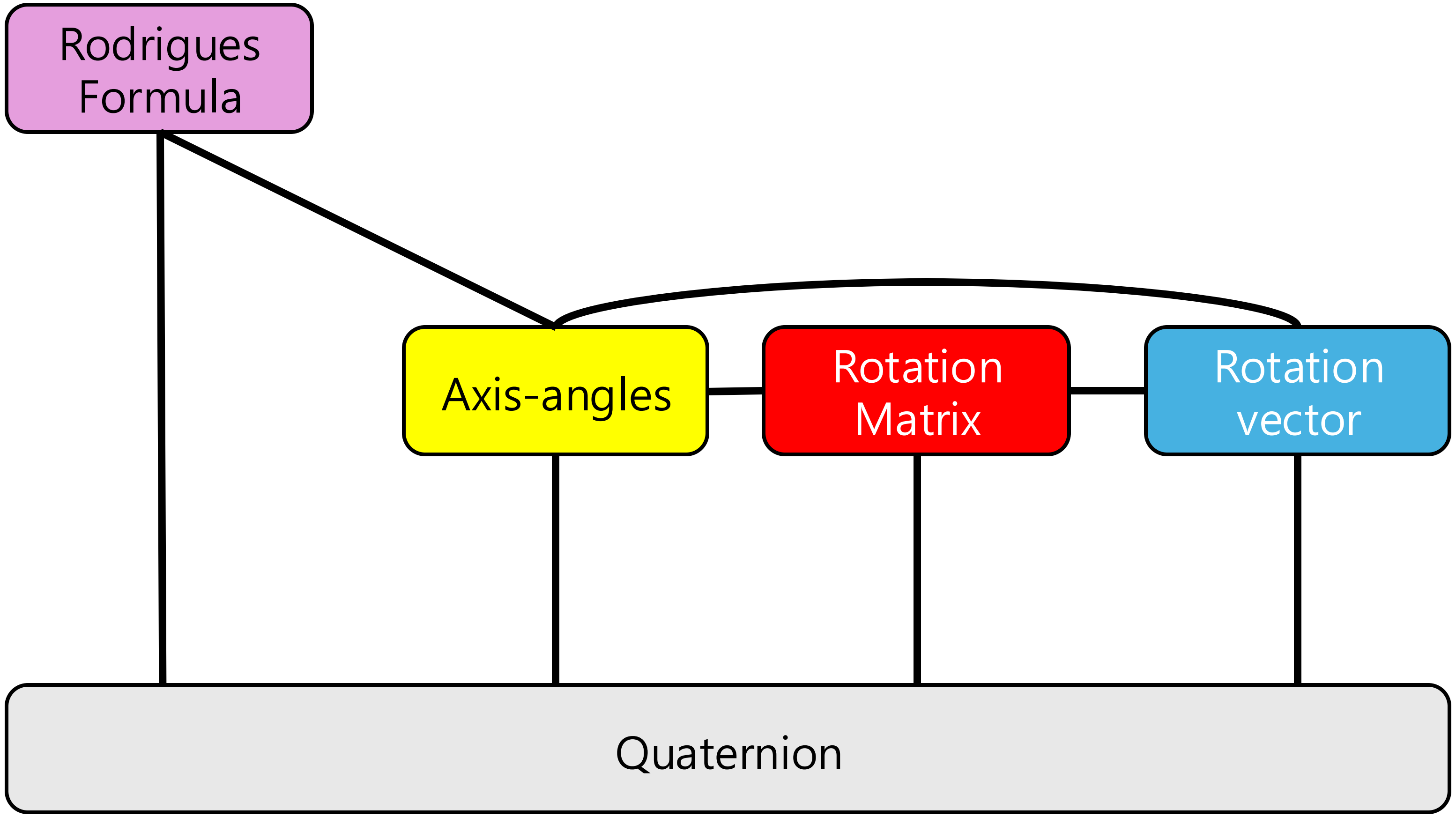
```
void glRotated(GLdoubleangle,
               GLdoublex,
               GLdoubley,
               GLdoublez);
```

$$\left(\begin{bmatrix} \hat{n}_x \\ \hat{n}_y \\ \hat{n}_z \end{bmatrix}, \theta \right)$$

$$R = \begin{bmatrix} (1-\cos(\theta))\hat{n}_x^2 + \cos(\theta) & (1-\cos(\theta))\hat{n}_x\hat{n}_y - \sin(\theta)\hat{n}_z & (1-\cos(\theta))\hat{n}_z\hat{n}_x + \sin(\theta)\hat{n}_y \\ (1-\cos(\theta))\hat{n}_x\hat{n}_y + \sin(\theta)\hat{n}_z & (1-\cos(\theta))\hat{n}_y^2 + \cos(\theta) & (1-\cos(\theta))\hat{n}_y\hat{n}_z - \sin(\theta)\hat{n}_x \\ (1-\cos(\theta))\hat{n}_x\hat{n}_z - \sin(\theta)\hat{n}_y & (1-\cos(\theta))\hat{n}_y\hat{n}_z + \sin(\theta)\hat{n}_x & (1-\cos(\theta))\hat{n}_z^2 + \cos(\theta) \end{bmatrix}$$

Axis-angles

The Relation: Rotation Vector, Rotation Matrix, Quaternion, Rodrigues Formula



Axis-angles

The Relation: Rotation Vector, Rotation Matrix, Quaternion, Rodrigues Formula

$$\left(\begin{bmatrix} \hat{n}_x \\ \hat{n}_y \\ \hat{n}_z \end{bmatrix}, \theta \right) \longrightarrow R = \begin{bmatrix} (1-\cos(\theta))\hat{n}_x^2 + \cos(\theta) & (1-\cos(\theta))\hat{n}_x\hat{n}_y - \sin(\theta)\hat{n}_z & (1-\cos(\theta))\hat{n}_z\hat{n}_x + \sin(\theta)\hat{n}_y \\ (1-\cos(\theta))\hat{n}_x\hat{n}_y + \sin(\theta)\hat{n}_z & (1-\cos(\theta))\hat{n}_y^2 + \cos(\theta) & (1-\cos(\theta))\hat{n}_y\hat{n}_z - \sin(\theta)\hat{n}_x \\ (1-\cos(\theta))\hat{n}_x\hat{n}_z - \sin(\theta)\hat{n}_y & (1-\cos(\theta))\hat{n}_y\hat{n}_z + \sin(\theta)\hat{n}_x & (1-\cos(\theta))\hat{n}_z^2 + \cos(\theta) \end{bmatrix}$$

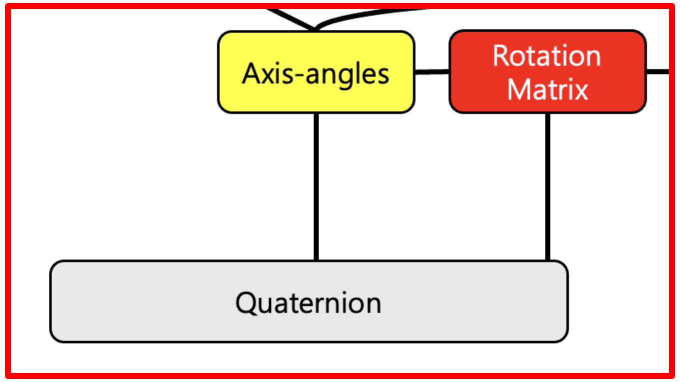
$$Rotation = (\hat{n}, \theta)$$

$$\mathrm{q} = \left(\cos\frac{\theta}{2}, \hat{n}\sin\frac{\theta}{2}\right)$$

$$q = (q_0, q_1, q_2, q_3)$$

$$q_0 = \cos\frac{\theta}{2}$$

$$\mathrm{q}_1, q_2, q_3 = \hat{n}\sin\frac{\theta}{2}$$



Axis-angles

Quaternion to Matrix

$$q = a + bi + cj + dk \longrightarrow \begin{bmatrix} 2a^2 - 1 + 2b^2 & 2bc + 2ad & 2bd - 2ac \\ 2bc - 2ad & 2a^2 - 1 + 2c^2 & 2cd + 2ab \\ 2bd + 2ac & 2cd - 2ab & 2a^2 - 1 + 2d^2 \end{bmatrix}$$

Rodrigues Formula

Rodrigues Formula

The Rodrigues rotation formula is formula that rotates a vector in three-dimensional space using an arbitrary axis and angle of rotation.

- It allows vectors to be rotated without directly computing rotation matrices.
- high computational efficiency
- computer graphics, robotics, SLAM(Simultaneous Localization And Mapping)

$$\mathbf{v}_{\mathrm{rot}} = \mathbf{v}\cos\theta + (\mathbf{k} \times \mathbf{v})\sin\theta + \mathbf{k}\,(\mathbf{k} \cdot \mathbf{v})(1 - \cos\theta)$$

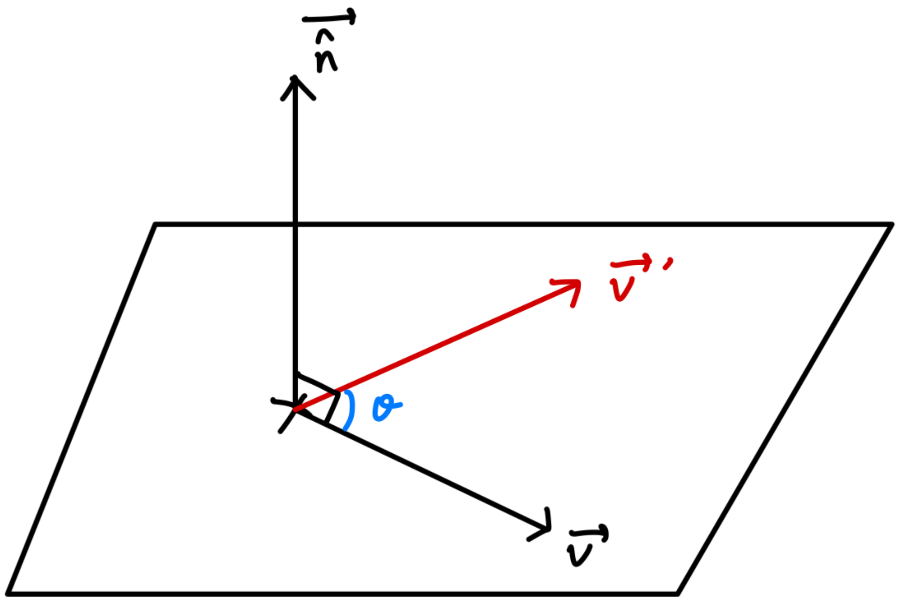
v: A Vector
k: Rotation-Axis
$\theta : angle$

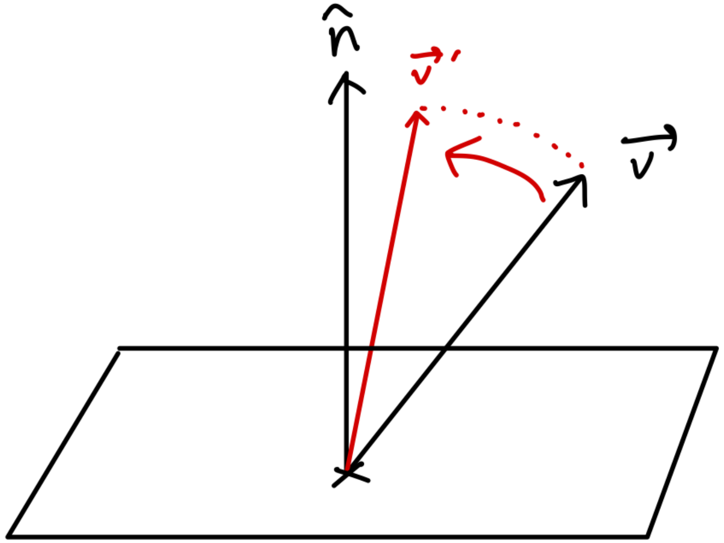
$\longrightarrow \mathbf{v}_{\mathrm{rot}}$

Rodrigues Formula

3D rotation

Special Case -> General Case(Rodrigues Formula)



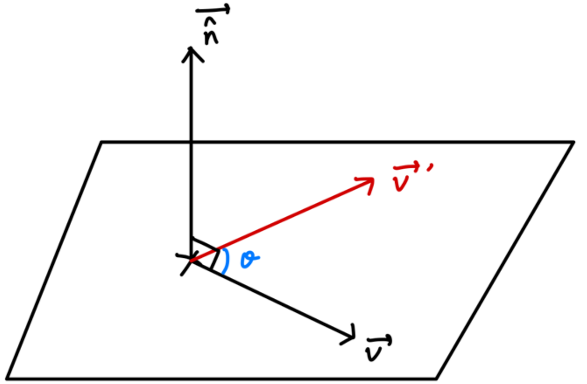


$$\vec{v'} = \cos\theta \cdot \vec{v} + \sin\theta \cdot (\hat{n} \times \vec{v})$$

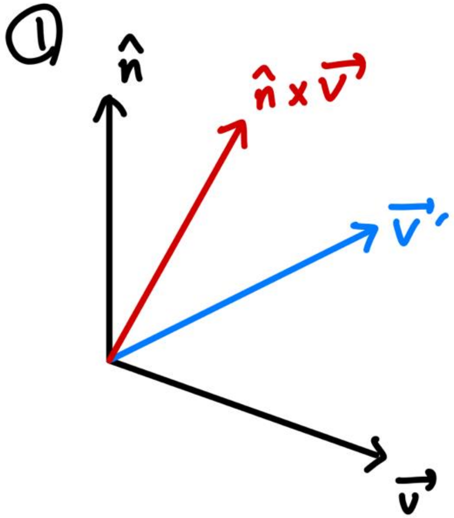
$$\vec{v'} = (1 - \cos\theta)(\vec{v} \cdot \hat{n})\hat{n} + \cos\theta \cdot \vec{v} + \sin\theta \cdot (\hat{n} \times \vec{v})$$

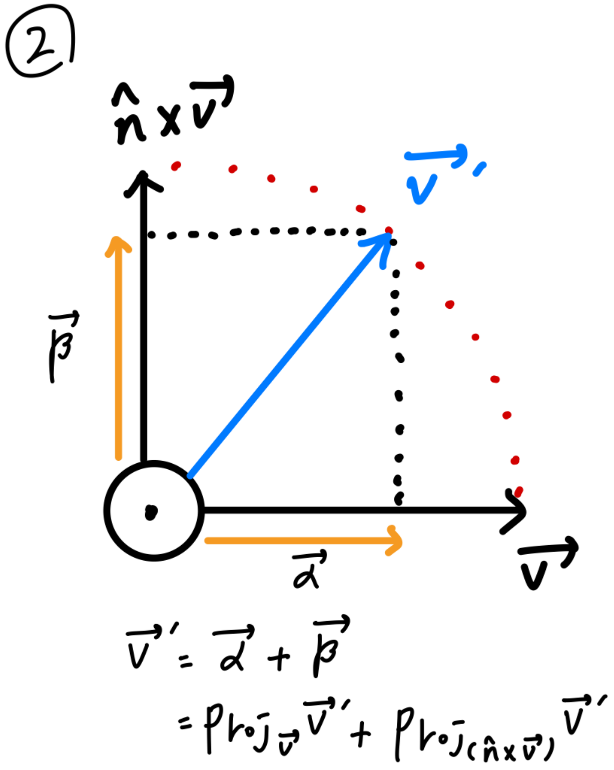
Rodrigues Formula: Special Case

3D rotation



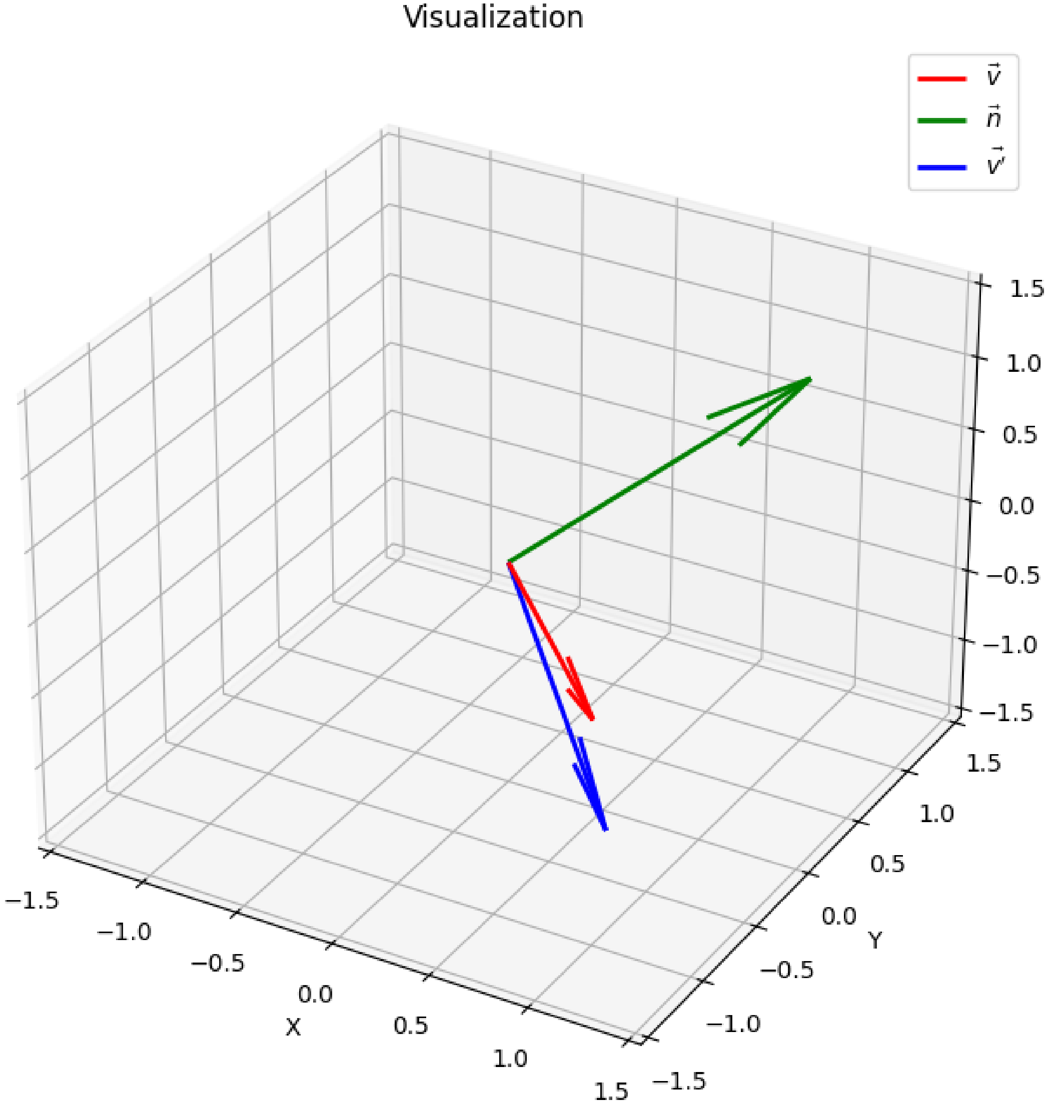
$$\vec{v}' = \cos\theta \cdot \vec{v} + \sin\theta \cdot (\hat{n} \times \vec{v})$$





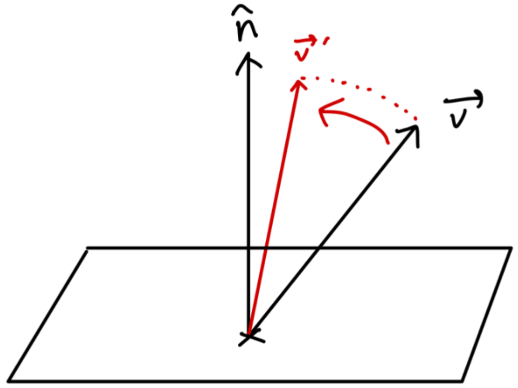
Rodrigues Formula: Special Case

Roatate $\vec{v} = (1, -1, 0)$ by $\frac{\pi}{6}$ radians about the axis $\vec{n} = (1, 1, 1)$

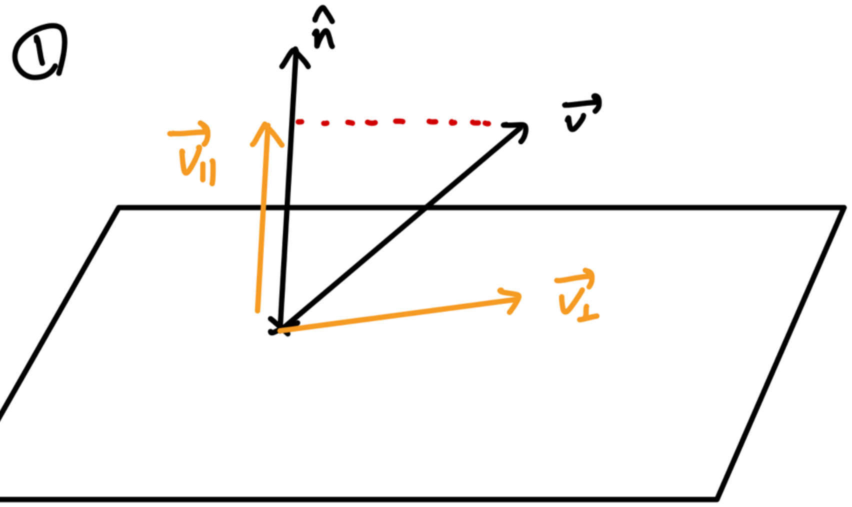


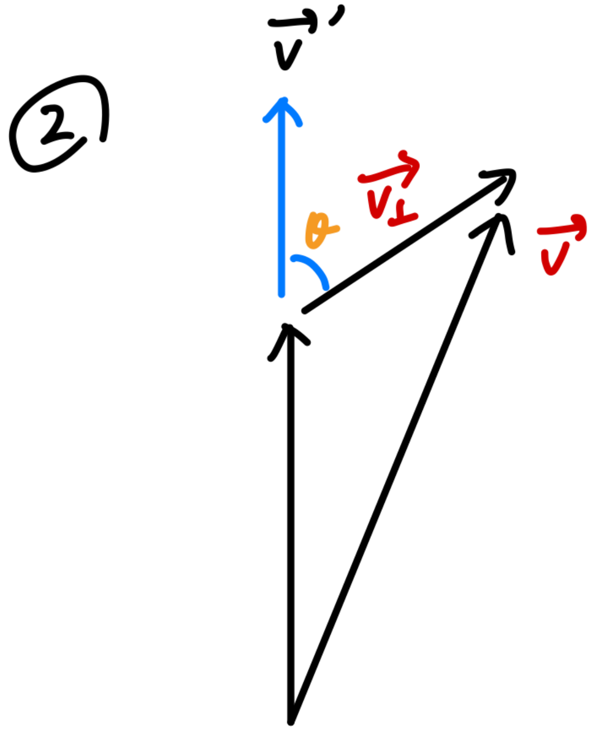

```
v = np.array([1, -1, 0])
n = np.array([1, 1, 1])
theta = np.pi / 6

n_mag = np.linalg.norm(n)
n_hat = n / n_mag

#formula
v_prime = np.cos(theta)*v + np.sin(theta) * np.cross(n_hat, v)
```

Rodrigues Formula: General Case



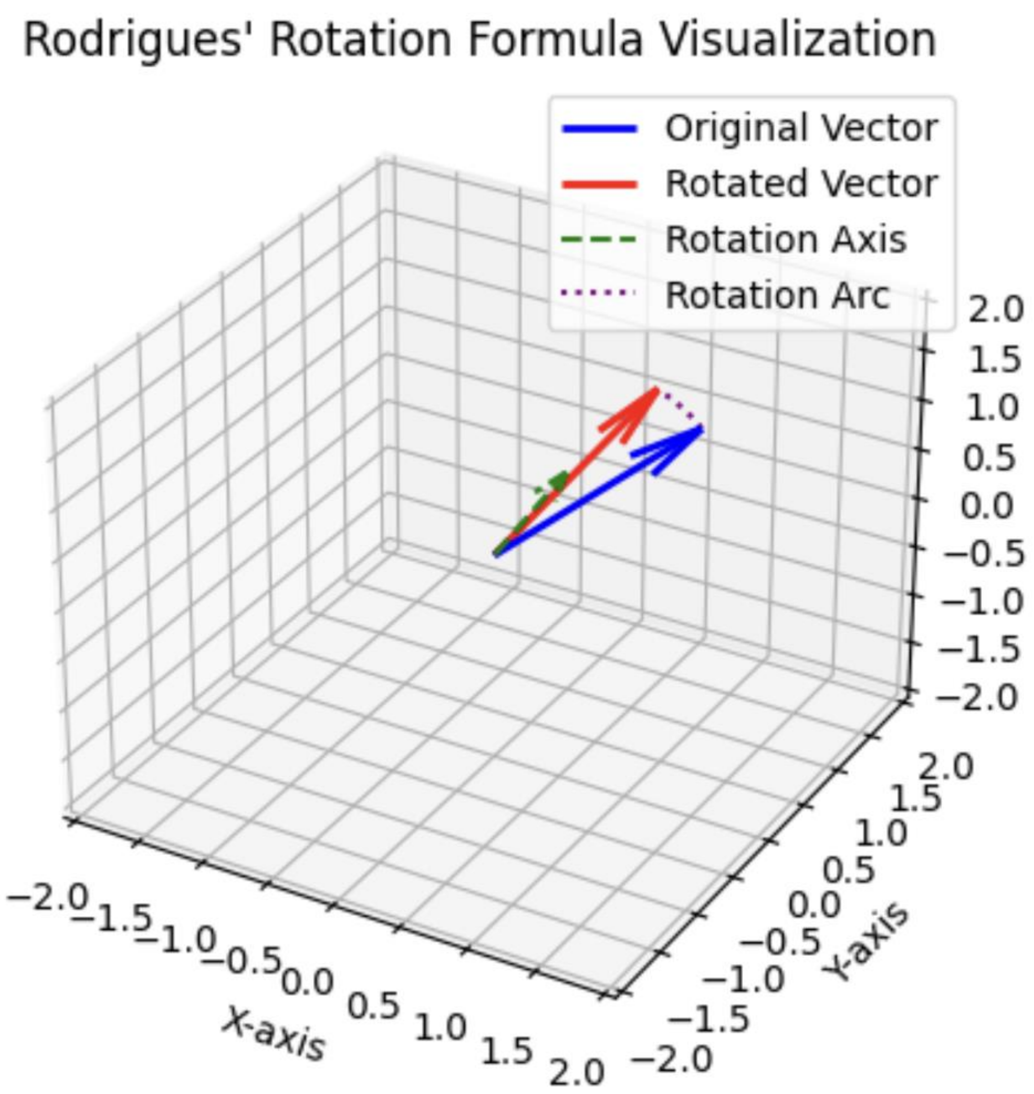
$$\vec{v}' = (1 - \cos\theta)(\vec{v} \cdot \hat{n})\hat{n} + \cos\theta \cdot \vec{v} + \sin\theta \cdot (\hat{n} \times \vec{v})$$





Rodrigues Formula: General Case

Roatate $\vec{v} = (1, 1, 1)$ by $\frac{\pi}{6}$ radians about the axis $\vec{n} = (0.5, 0.2, 1.0)$




```
v = np.array([1, 1, 1])
n = np.array([0.5, 0.2, 1.0])
theta = np.pi / 6

n_mag = np.linalg.norm(n)
n_hat = n / n_mag
```

```
v_prime = (
    np.cos(theta) * v
    + np.sin(theta) * np.cross(n_hat, v)
    + (1 - np.cos(theta)) * np.dot(n_hat, v) * n_hat
)
```

Q&A