CSE 167 (FA22)
Computer Graphics:
3D Rotations

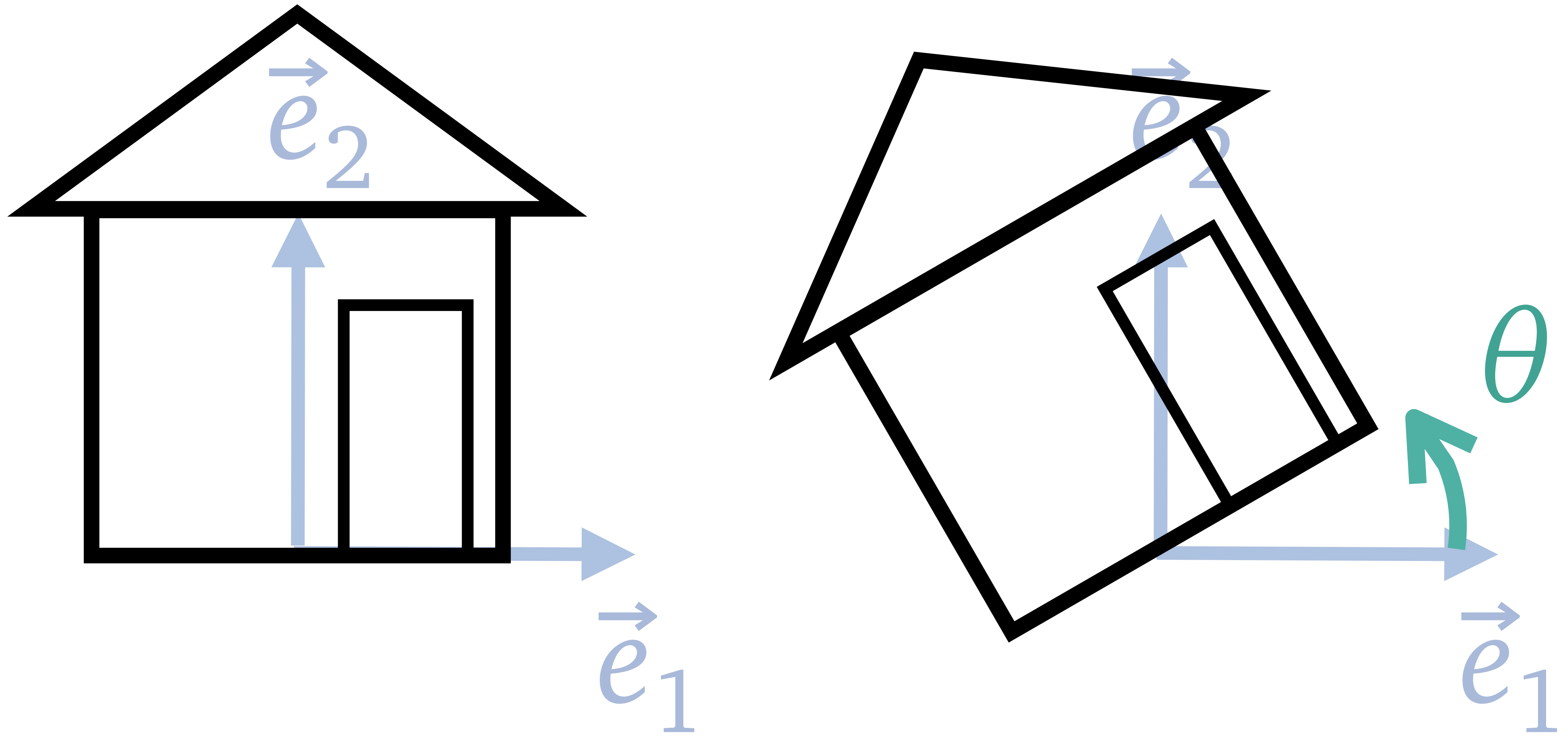
Albert Chern

UC San Diego
JACOBS SCHOOL OF ENGINEERING

Recall 2D rotations

- The 2D rotation matrix is

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



2D rotation using complex number

real part

real part

$$\begin{bmatrix} \boxed{\cos\theta} & -\boxed{\sin\theta} \\ \boxed{\sin\theta} & \boxed{\cos\theta} \end{bmatrix} \begin{bmatrix} \boxed{a} \\ \boxed{b} \end{bmatrix} = \begin{bmatrix} \boxed{\cos(\theta)a - \sin(\theta)b} \\ \boxed{\sin(\theta)a + \cos(\theta)b} \end{bmatrix}$$

imaginary part

imaginary part

- We can view **2D vectors** as complex numbers

  and **rotation matrices** also as complex numbers

$$(\cos\theta + \mathrm{i}\sin\theta)(a + \mathrm{i}\,b) = \big(\cos(\theta)a - \sin(\theta)b\big) + \mathrm{i}\big(\sin(\theta)a + \cos(\theta)b\big)$$

2D rotation using complex number

real part

real part

$$\begin{bmatrix} \boxed{\cos\theta} & -\boxed{\sin\theta} \\ \boxed{\sin\theta} & \boxed{\cos\theta} \end{bmatrix} \begin{bmatrix} \boxed{a} \\ \boxed{b} \end{bmatrix} = \begin{bmatrix} \boxed{\cos(\theta)a - \sin(\theta)b} \\ \boxed{\sin(\theta)a + \cos(\theta)b} \end{bmatrix}$$

imaginary part

imaginary part

- We can view **2D vectors** as complex numbers

  and **rotation matrices** also as complex numbers

$$\underbrace{\boxed{(\cos\theta + i\sin\theta)}}_{\text{this “rotor” must have length=1}}\underbrace{\boxed{(a + i b)}}_{\text{arbitrary vector being rotated}} = \big(\cos(\theta)a - \sin(\theta)b\big) + i\big(\sin(\theta)a + \cos(\theta)b\big)$$

2D rotation using complex number

$$\boxed{(\cos\theta + \mathbb{i}\sin\theta)}\boxed{(a + \mathbb{i}b)} = \big(\cos(\theta)a - \sin(\theta)b\big) + \mathbb{i}\big(\sin(\theta)a + \cos(\theta)b\big)$$

this "rotor" must have length=1

arbitrary vector being rotated

- Euler formula $e^{\mathbb{i}\theta} = \cos\theta + \mathbb{i}\sin\theta$
- 2D rotation $\boxed{e^{\mathbb{i}\theta}}\boxed{(a + \mathbb{i}b)}$

the rotor

arbitrary vector being rotated

- Rotor–rotation matrix conversion $\begin{bmatrix} \mathrm{Re}(e^{\mathbb{i}\theta}) & -\mathrm{Im}(e^{\mathbb{i}\theta}) \\ \mathrm{Im}(e^{\mathbb{i}\theta}) & \mathrm{Re}(e^{\mathbb{i}\theta}) \end{bmatrix}$

What about 3D rotations

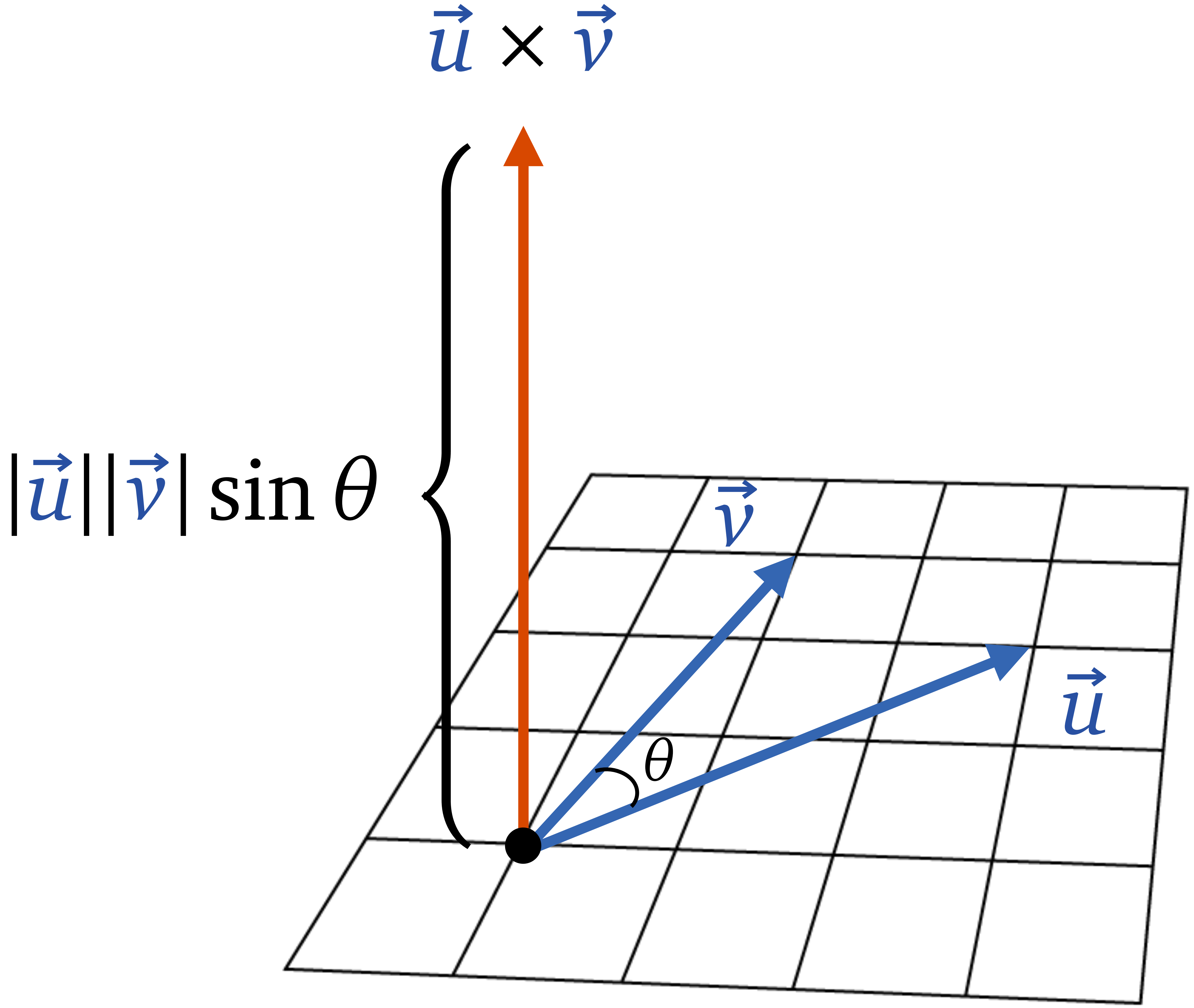
3D rotation

- Each 2D rotation is described by an angle → • Each 3D rotation is described by an **angle** and an **rotation axis**
- 2D rotation matrix → • Rotation matrix (Rodrigues formula)
  → • Euler angle system
- Complex numbers → • Quaternions

- Geometric algebra

Cross Product

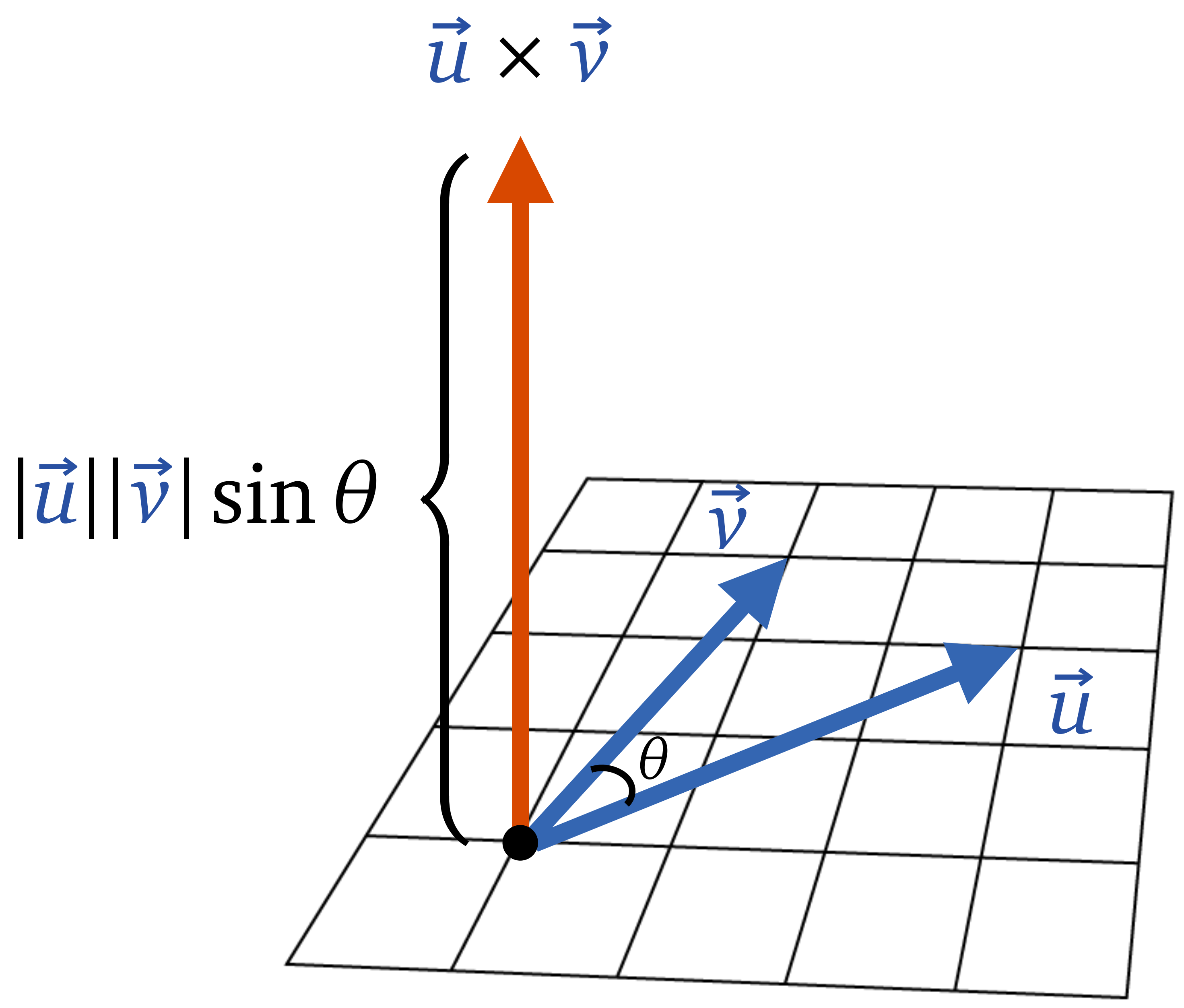
Cross product (geometric)



Cross product (algebraic)

$$\begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} \times \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} u_y v_z - u_z v_y \\ u_z v_x - u_x v_z \\ u_x v_y - u_y v_x \end{bmatrix}$$

Cross product



$$\begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} \times \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} u_y v_z - u_z v_y \\ u_z v_x - u_x v_z \\ u_x v_y - u_y v_x \end{bmatrix}$$

Suppose $\vec{\mathbf{e}}$ is an orthonormal basis, $\vec{u} = \vec{\mathbf{e}}^{\mathsf{T}}\mathbf{u}$, $\vec{v} = \vec{\mathbf{e}}^{\mathsf{T}}\mathbf{v}$.
Then

$$\vec{u} \times \vec{v} = \vec{\mathbf{e}}^{\mathsf{T}}(\mathbf{u} \times \mathbf{v})$$

Cross product (properties)

$$\begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} \times \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} u_y v_z - u_z v_y \\ u_z v_x - u_x v_z \\ u_x v_y - u_y v_x \end{bmatrix}$$

- Skew-symmetric $\mathbf{u} \times \mathbf{v} = -\mathbf{v} \times \mathbf{u}$
- Non-associative. In general, $\mathbf{u} \times (\mathbf{v} \times \mathbf{w}) \neq (\mathbf{u} \times \mathbf{v}) \times \mathbf{w}$
- Bilinear. And

$$\mathbf{u} \times \mathbf{v} = \begin{bmatrix} & -u_z & u_y \\ u_z & & -u_x \\ -u_y & u_x & \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$$

3D Rotations (angle-axis)

3D Rotation (Rodrigues formula)

- We can describe a 3D rotation by an axis $\mathbf{a} \in \mathbb{R}^3, \ |\mathbf{a}| = 1$ and an angle $\theta \in \mathbb{R}$

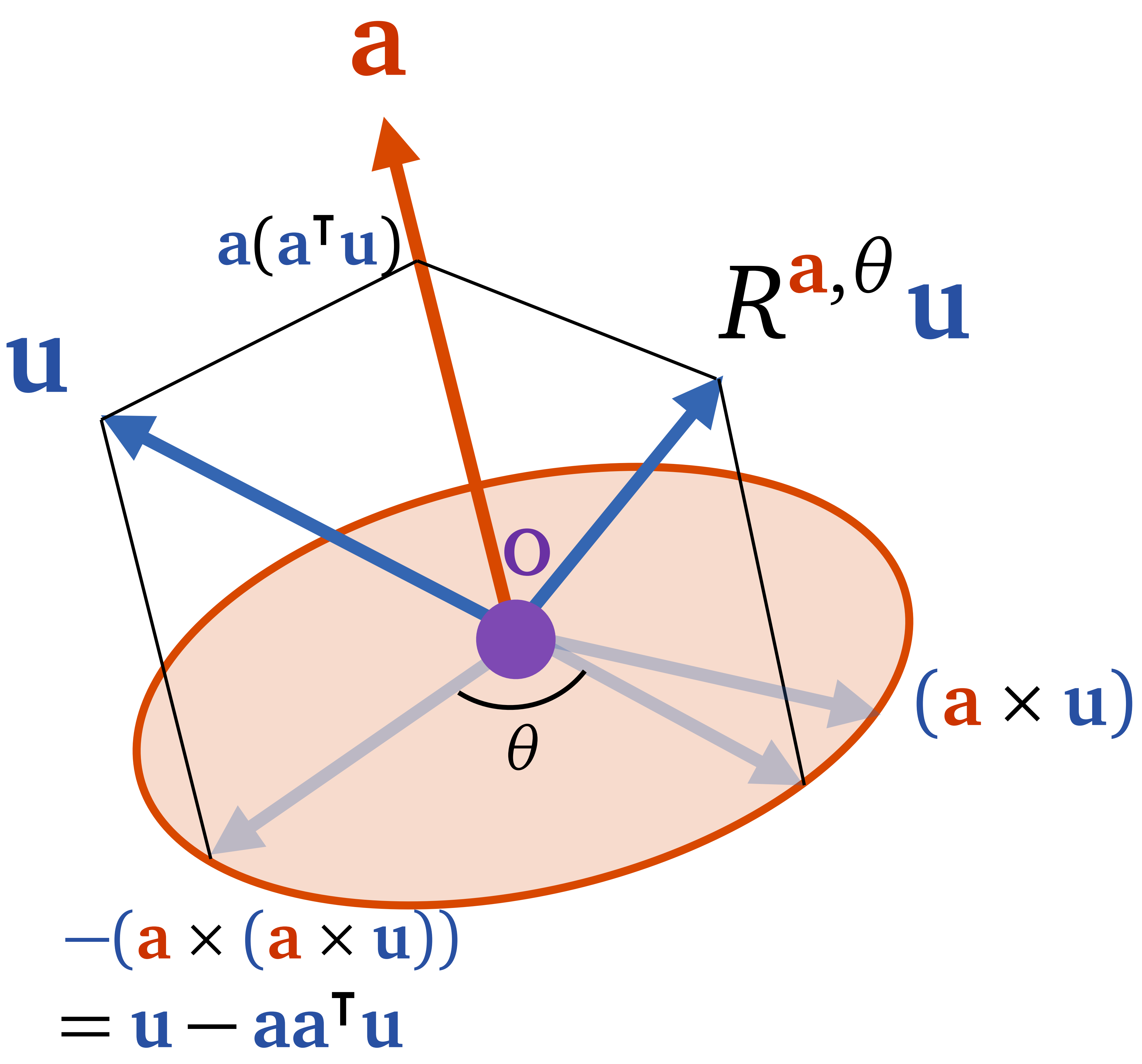
- Rodrigues formula

$$R^{\mathbf{a},\theta}\mathbf{u}$$
$$= \mathbf{a}(\mathbf{a}^\intercal\mathbf{u}) + \cos\theta(\mathbf{u} - \mathbf{a}\mathbf{a}^\intercal\mathbf{u}) + \sin\theta(\mathbf{a} \times \mathbf{u})$$

that is,

$$R^{\mathbf{a},\theta} = \cos\theta\, I_{3\times 3} + (1 - \cos\theta)\mathbf{a}\mathbf{a}^\intercal + \sin\theta[\mathbf{a}\times]$$

$$[\mathbf{a}\times] = \begin{bmatrix} 0 & -a_2 & a_1 \\ a_2 & 0 & -a_0 \\ -a_1 & a_0 & 0 \end{bmatrix}$$



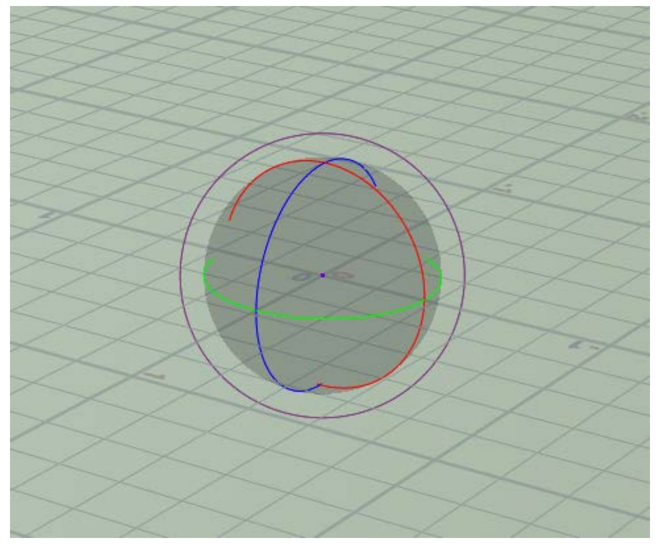
Other representations of 3D rotations

Various ways for 3D rotations

- Rodrigues formula (angle-axis) (previous slides)

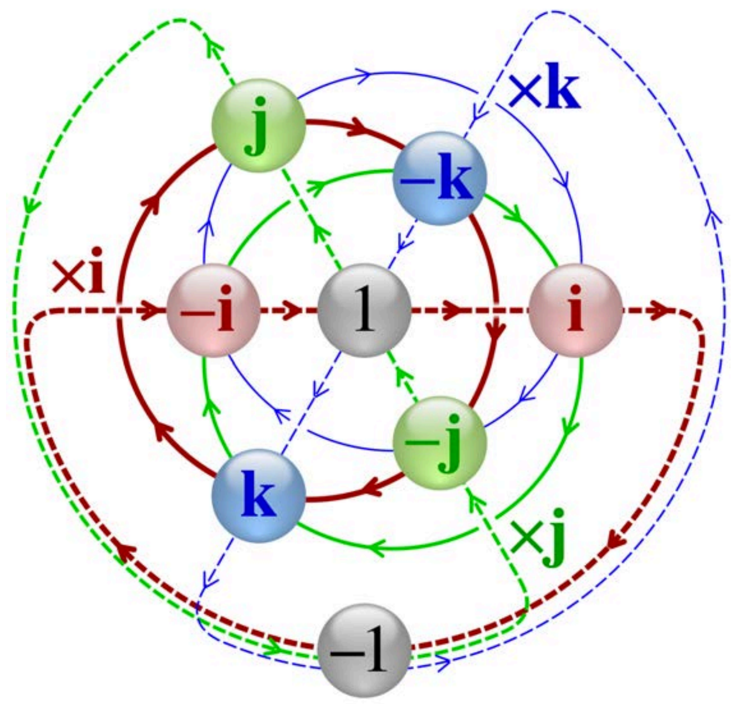
$$R^{\mathbf{a},\theta} = \cos\theta\, I_{3\times 3} + (1-\cos\theta)\mathbf{a}\mathbf{a}^{\intercal} + \sin\theta[\mathbf{a}\times]$$

- Euler angles (3 planar rotations)



$$R^{(\alpha,\beta\gamma)} = R^{\vec{e}_y,\alpha} R^{\vec{e}_z,\beta} R^{\vec{e}_x,\gamma}$$

- Quaternions (angle-axis)



$$R^{\mathbb{a},\theta}\mathbb{w} = e^{\frac{\theta}{2}\mathbb{a}}\mathbb{w}\,e^{-\frac{\theta}{2}\mathbb{a}}$$

- Geometric algebra (a way to understand quaternions and general rotors)

Various ways for 3D rotations

- With any other way of doing 3D rotations, you can convert it to a 3x3 rotation matrix using the following pseudocode

Rotating vectors is enough

```
vec3 rotate(  rotation_parameters ,   vec3   v  ){
   …
   return rotated_vector;
}

mat3 rotationMatrix(  rotation_parameters  ){
   vec3  e1 = (1,0,0); vec3 e2 = (0,1,0); vec3 e3 = (0,0,1);
   vec3  r1 = rotate( rotation_parameters,  e1 );
   vec3  r2 = rotate( rotation_parameters,  e2 );
   vec3  r3 = rotate( rotation_parameters,  e3 );
   mat3 R;  R[0] = r1; R[1] = r2; R[2] = r3; // column major
   return R;
}
```

Euler Angles

Quaternions

Geometric Algebra