OpenGL Rotation About Arbitrary Axis

The 4x4 transformation matrix for rotating about an arbitrary axis in OpenGL is defined as

$$M_R = \begin{pmatrix} (1-c)x^2 + c & (1-c)xy - sz & (1-c)xz + sy & 0 \\ (1-c)xy + sz & (1-c)y^2 + c & (1-c)yz - sx & 0 \\ (1-c)xz - sy & (1-c)yz + sx & (1-c)z^2 + c & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\text{where } c = \cos\theta, \quad s = \sin\theta, \quad \vec{r} = (x, y, z)$$

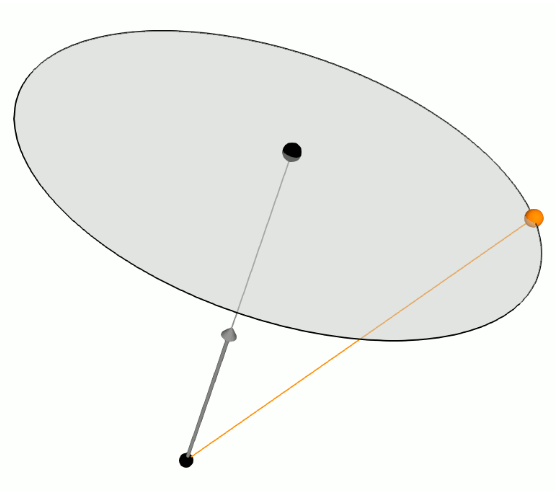
This page explains how to derive this rotation matrix using Rodrigues' formula. Suppose a 3D point *P* is rotating to Q by an angle θalong a unit vector $\vec{r} = (x, y, z)$.

The vector form of P is broken up the sum of $\overrightarrow{OR}$ and $\overrightarrow{RP}$, and Q is the sum of $\overrightarrow{OR}$ and $\overrightarrow{RQ}$ respectively:

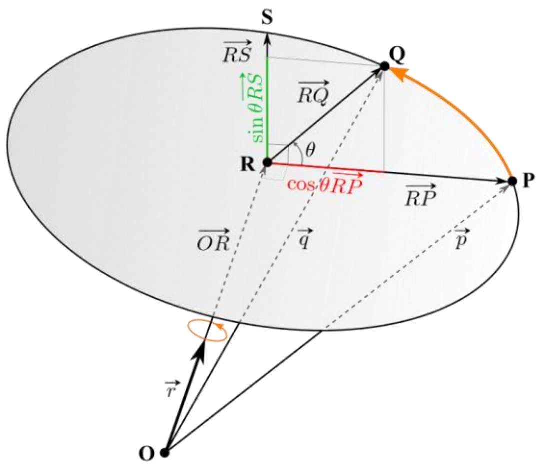
$$\vec{p} = \overrightarrow{OR} + \overrightarrow{RP}, \vec{q} = \overrightarrow{OR} + \overrightarrow{RQ}$$

Therefore, we need to find $\overrightarrow{OR}$ and $\overrightarrow{RQ}$ first in order to get Q. The vector $\overrightarrow{OR}$ can be acquired from $\vec{p}$, and $\overrightarrow{RQ}$comes from the circular plane, where the point Q lies on. $\overrightarrow{OR}$ is parallel to the unit vector $\vec{r}$, and its length can be computed by projecting $\vec{p}$ to $\vec{r}$ using inner product. So, it can be written as:

$$\overrightarrow{OR} = (\vec{p} \cdot \vec{r})\vec{r}$$



Animation Of Rotating A Vertex About Arbitrary Axis



Rotating P to Q About Arbitrary Axis

$\overrightarrow{RQ}$ can be determined by 2 basis vectors on the rotation plane. We use $\overrightarrow{RP}$ as the first basis vector, and the other basis vector $\overrightarrow{RS}$ is perpendicular to $\overrightarrow{RP}$and has equal length because they are both the radius of the circular plane. Therefore, $\overrightarrow{RS}$ can be computed by the cross product of 2 perpendicular vectors; $\vec{r}$ and $\overrightarrow{RP}$

$$\begin{aligned}\overrightarrow{RS} &= \vec{r} \times \overrightarrow{RP} \\ &= \vec{r} \times (\vec{p} - \overrightarrow{OR}) \qquad (\because \vec{p} = \overrightarrow{OR} + \overrightarrow{RP}) \\ &= \vec{r} \times \vec{p} - \vec{r} \times \overrightarrow{OR} \\ &= \vec{r} \times \vec{p} \qquad\qquad\qquad (\because \vec{r} \times \overrightarrow{OR} = 0)\end{aligned}$$

Now, $\overrightarrow{RQ}$ is represented with the composition of these basis vectors and trigonometric functions:

$$\begin{aligned}\overrightarrow{RQ} &= \cos\theta \overrightarrow{RP} + \sin\theta \overrightarrow{RS} \\ &= \cos\theta \overrightarrow{RP} + \sin\theta (\vec{r} \times \vec{p})\end{aligned}$$

Finally, the rotated vector $\vec{q}$ is written by the sum of $\overrightarrow{OR}$ and $\overrightarrow{RQ}$

$$\begin{aligned}\vec{q} &= \overrightarrow{OR} + \overrightarrow{RQ} \\ &= \overrightarrow{OR} + \cos\theta \overrightarrow{RP} + \sin\theta(\vec{r} \times \vec{p}) \qquad (\because \overrightarrow{RQ} = \cos\theta \overrightarrow{RP} + \sin\theta(\vec{r} \times \vec{p})) \\ &= \overrightarrow{OR} + \cos\theta(\vec{p} - \overrightarrow{OR}) + \sin\theta(\vec{r} \times \vec{p}) \qquad (\because \vec{p} = \overrightarrow{OR} + \overrightarrow{RP}) \\ &= (1 - \cos\theta)\overrightarrow{OR} + \cos\theta \vec{p} + \sin\theta(\vec{r} \times \vec{p}) \\ &= (1 - \cos\theta)(\vec{p} \cdot \vec{r})\vec{r} + \cos\theta \vec{p} + \sin\theta(\vec{r} \times \vec{p}) \quad (\because \overrightarrow{OR} = (\vec{p} \cdot \vec{r})\vec{r})\end{aligned}$$

This equation is called Rodrigues' rotation formula:

$$\vec{q} = (1 - \cos\theta)(\vec{p} \cdot \vec{r})\vec{r} + \cos\theta \vec{p} + \sin\theta(\vec{r} \times \vec{p})$$

It can be represented by an equivalent matrix form. First, convert $\overrightarrow{OR}$ and $\overrightarrow{RS}$ components to 3x3 matrix forms for P = (p_x, p_y, p_z) and r = (x, y, z)

$$\begin{aligned}(\vec{p} \cdot \vec{r})\vec{r} &= (p_x x + p_y y + p_z z)\vec{r} \\ &= (p_x x + p_y y + p_z z)\begin{pmatrix} x \\ y \\ z \end{pmatrix} \\ &= \begin{pmatrix} p_x x^2 + p_y xy + p_z xz \\ p_x xy + p_y y^2 + p_z yz \\ p_x xz + p_y yz + p_z z^2 \end{pmatrix} \\ &= \begin{pmatrix} x^2 & xy & xz \\ xy & y^2 & yz \\ xz & yz & z^2 \end{pmatrix}\begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix}\end{aligned}$$

$$\begin{aligned}\vec{r} \times \vec{p} &= \begin{pmatrix} yp_z - zp_y \\ zp_x - xp_z \\ xp_y - yp_x \end{pmatrix} \\ &= \begin{pmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{pmatrix}\begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix}\end{aligned}$$

Finally, the equivalent matrix form by substituting the above matrix components is

$$
\begin{aligned}
\vec{q} &= (1-\cos\theta)(\vec{p}\cdot\vec{r})\vec{r} + \cos\theta\,\vec{p} + \sin\theta(\vec{r}\times\vec{p}) \\
&= (1-c)(\vec{p}\cdot\vec{r})\vec{r} + c\vec{p} + s(\vec{r}\times\vec{p}) \qquad (\text{Let } c=\cos\theta,\ s=\sin\theta) \\
&= (1-c)\begin{pmatrix} x^2 & xy & xz \\ xy & y^2 & yz \\ xz & yz & z^2 \end{pmatrix}\begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} + c\begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} + s\begin{pmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{pmatrix}\begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} \\
&= \left[(1-c)\begin{pmatrix} x^2 & xy & xz \\ xy & y^2 & yz \\ xz & yz & z^2 \end{pmatrix} + c\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} + s\begin{pmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{pmatrix}\right]\begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} \\
&= \begin{pmatrix} (1-c)x^2+c & (1-c)xy-sz & (1-c)xz+sy \\ (1-c)xy+sz & (1-c)y^2+c & (1-c)yz-sx \\ (1-c)xz-sy & (1-c)yz+sx & (1-c)z^2+c \end{pmatrix}\begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix}
\end{aligned}
$$

And, the 3x3 rotation matrix alone is

$$
M_R = \begin{pmatrix} (1-c)x^2+c & (1-c)xy-sz & (1-c)xz+sy \\ (1-c)xy+sz & (1-c)y^2+c & (1-c)yz-sx \\ (1-c)xz-sy & (1-c)yz+sx & (1-c)z^2+c \end{pmatrix}
$$

$$
\text{where } c=\cos\theta, \quad s=\sin\theta, \quad \vec{r}=(x,y,z)
$$

Or, as 4x4 matrix

$$
M_R = \begin{pmatrix} (1-c)x^2+c & (1-c)xy-sz & (1-c)xz+sy & 0 \\ (1-c)xy+sz & (1-c)y^2+c & (1-c)yz-sx & 0 \\ (1-c)xz-sy & (1-c)yz+sx & (1-c)z^2+c & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}
$$

$$
\text{where } c=\cos\theta, \quad s=\sin\theta, \quad \vec{r}=(x,y,z)
$$

Example: Rodrigues' Rotation Formula

The following C++ code snippet is rotating a 3D point P to Q along the rotation axis $\vec{r}$ using Rodrigues' formula. Download the complete implementation from rotate.zip.

$$
\vec{q} = (1-\cos\theta)(\vec{p}\cdot\vec{r})\vec{r} + \cos\theta\,\vec{p} + \sin\theta(\vec{r}\times\vec{p})
$$

```
// minimal implementation of Vector3
struct Vector3
{
    float x, y, z;
    // ctor
    Vector3() : x(0), y(0), z(0) {}

    // inner and cross products
    float   dot(Vector3& v)    {
      return x*v.x + y*v.y + z*v.z;
    }
    Vector3 cross(Vector3& v) {
      return Vector3(y*v.z-z*v.y, z*v.x-x*v.z, x*v.y-y*v.x);
    }
    // scalar product
    friend Vector3 operator*(float s, Vector3 v) {
      return Vector3(s*v.x, s*v.y, s*v.z);
    }
    Vector3& normalize() {
        float invLength = 1.0f / sqrtf(x*x + y*y + z*z);
        x *= invLength;
        y *= invLength;
        z *= invLength;
        return *this;
    }
}
...

// define the rotation vector r and angle
Vector3 r = Vector3(1, 1, 1).normalize();   // make unit length
float a = 30 / 180 * PI;                     // rotation angle as radian

// define a vector p to rotate
Vector3 p = Vector3(1, 2, 3);

// compute the rotated vector q using Rodrigues' formula
Vector3 q = (1 - cos(a)) * p.dot(r) * r + cos(a) * p + sin(a) * r.cross(p);
```