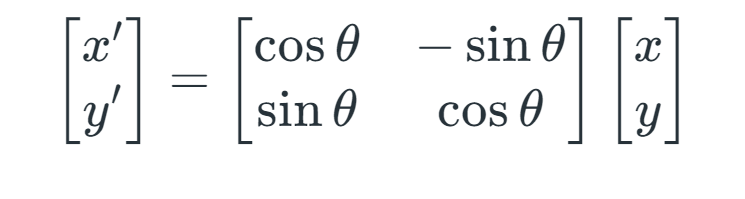
**IMAGE TRANSFORMATIONS - ROTATION**

Rotation is a process in which a image is simply rotated around the origin or an image center by a given angle. This rotates the image or changes the orientation of an image depending on the angle it has been set to.

Its equation is:

****

**Rotation about arbitrary point:**

Let’s say you want to rotate a point around a pivot point that is not the origin.

**Step 1: Identify the Point and Pivot**

Choose the point you want to rotate → example: PPP

Choose the pivot point (center of rotation) → example: CCC

**Step 2: Move the Pivot to the Origin**

Shift the entire setup so that the pivot point becomes the origin.

This is done by subtracting the pivot’s coordinates from the point.

Now your point is in a new temporary position as if the pivot is at the origin.

**Step 3: Rotate Around the Origin**

Apply the rotation to the shifted point (since the pivot is now at the origin).

Use a counter-clockwise or clockwise rotation, depending on the angle and direction.

You’re rotating as if around the origin now.

**Step 4: Move the Pivot Back to Its Original Location**

Undo the shift you did in Step 2.

Add the pivot’s original coordinates back to the rotated point.

Now you get the final, rotated position of the original point.

**Translation: x’ = x - p**

**Rotation: x’’= R(x’) = Rx - Rp**

**Translation Back: x’’’ = x’’ + p = Rx - Rp + p**

**TYPES OF 3D ROTATION:**

**Rotation about x axis:**

The matrix shown is a 4×4 homogeneous rotation matrix that performs a rotation around the X-axis by an angle θ\thetaθ in 3D space. This transformation is commonly used in 3D graphics and computer vision to rotate points or objects along the X-axis. In this matrix, the X-coordinate remains unchanged, while the Y and Z coordinates are transformed using the cosine and sine of the rotation angle, as if they lie in the YZ-plane. This type of rotation effectively tilts the object upward or downward along the X-axis, depending on the sign of the angle, following the right-hand rule for counterclockwise rotation. The use of a 4×4 matrix allows for homogeneous coordinates, which makes it possible to incorporate other transformations such as translation and scaling within the same framework. When this matrix multiplies a point represented as [x,y,z,1]T[x, y, z, 1]^T[x,y,z,1]T, the result is a new point rotated by θ\thetaθ about the X-axis, while preserving the structure needed for further affine transformations.

**Rotation about y axis:**

The matrix shown is the 4×4 homogeneous rotation matrix for rotating a point in 3D space about the Y-axis by an angle θ\thetaθ. This transformation is commonly used in computer graphics, robotics, and 3D geometry to rotate objects around the vertical (Y) axis. In this matrix, the Y-coordinate of a point remains unchanged, while the X and Z coordinates are transformed as if the rotation occurs in the XZ-plane. The top-left 3×3 portion of the matrix performs the actual rotation, with cosine and sine terms controlling how much of the X and Z components mix during the rotation. The full 4×4 matrix format is used because it supports homogeneous coordinates, allowing not only rotation but also the integration of translation and scaling within the same matrix. When a point [x,y,z,1]T[x, y, z, 1]^T[x,y,z,1]T is multiplied by this matrix, the resulting coordinates represent the new position of that point after it has been rotated around the Y-axis by the specified angle θ\thetaθ, following the right-hand rule for counter-clockwise rotation

**Rotation about z axis:**

The matrix shown is a 4×4 homogeneous transformation matrix used to perform a rotation around the Z-axis by an angle θ\thetaθ in 3D space. This type of rotation affects only the X and Y coordinates of a point, leaving the Z-coordinate unchanged, effectively rotating the point within the XY-plane. The top-left 2×2 portion of the matrix contains the cosine and sine terms that control the rotation, while the third row and column preserve the Z value, and the last row and column handle homogeneous coordinate transformations. This form of rotation is particularly useful in computer graphics, robotics, and geometric modeling, where rotating objects about the Z-axis is common. Using the right-hand rule, a positive angle θ\thetaθ results in a counterclockwise rotation when looking from the positive Z-axis toward the origin. When this matrix is applied to a point represented in homogeneous coordinates as [x,y,z,1]T[x, y, z, 1]^T[x,y,z,1]T, the X and Y components are rotated by angle θ\thetaθ, while the Z and homogeneous components remain unchanged, enabling consistent affine transformations.

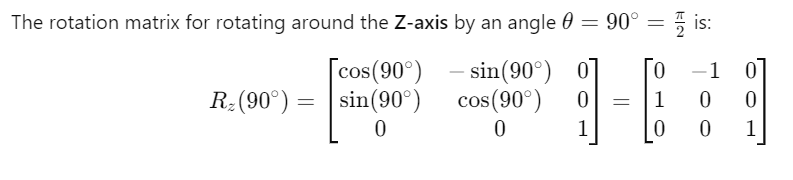
**EXAMPLE PROBLEM 1 :**

Rotate the point

P=(2,1,0)

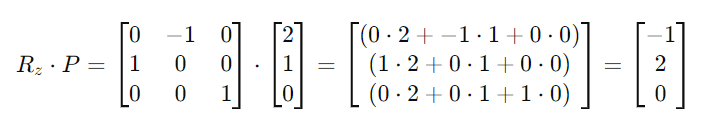
by 90° counter-clockwise around the Z-axis.

**Step 1: Write the 3×3 Z-axis Rotation Matrix:**



**Step 2: Multiply the Matrix by the Point:**

Now perform the matrix multiplication:



**Final Answer:**

The rotated point is



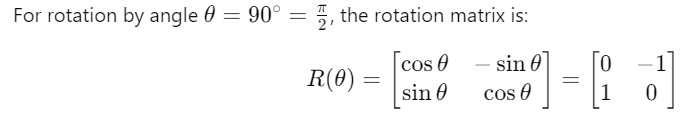
**EXAMPLE PROBLEM 2 :**

Rotate the point

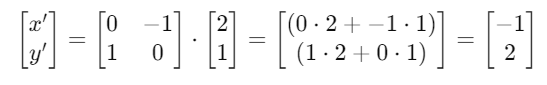


by **90° counter-clockwise** around the **origin (0,0)**.

**Step 1: Write the Rotation Matrix:**



**Step 2: Multiply Matrix with Point:**



**Final Answer:**

The rotated point is

