Algorithm for 3D Transformation

1. Input

- Coordinates of the rotation axis (x1, y1, z1 and x2, y2, z2).
- Angle of rotation (angle) in degrees.
- The 3D object to be rotated (e.g., a cube's vertices).

2. Calculate Direction Cosines

- Compute the direction vector of the axis:

  $dx = x2 - x1$, $dy = y2 - y1$, $dz = z2 - z1$
- Calculate the length of the axis:

  $L = \sqrt{(dx^2 + dy^2 + dz^2)}$
- Direction cosines of the axis:

  $\cos(\alpha) = dx / L$, $\cos(\beta) = dy / L$, $\cos(\gamma) = dz / L$

3. Create Transformation Matrices

1. Translation to Origin:

$$T1 = \begin{bmatrix} 1 & 0 & 0 & -x1 \\ 0 & 1 & 0 & -y1 \\ 0 & 0 & 1 & -z1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2. Rotation About X-axis to Align with ZX-plane:

$$Rx = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3. Rotation About Y-axis to Align with Z-axis:

$$Ry = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

4. Rotation About Z-axis by Given Angle:

$$R_z = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

5. Inverse Transformations:

Reverse Y-axis Rotation: R_y^{-1}

Reverse X-axis Rotation: R_x^{-1}

Reverse Translation: T2

4. Combine Transformations

Composite transformation matrix:

$$\text{Composite} = T2 * R_x^{-1} * R_y^{-1} * R_z * R_y * R_x * T1$$

5. Apply Transformation

Transform the vertices of the object:

$$\text{Transformed Vertices} = \text{Composite} * \text{Vertices}^T$$

6. Visualize Results

Plot the original and transformed objects using matplotlib.