Transformações Geométricas 3D

Disciplina: Computação Gráfica (BCC35F)
Curso: Ciência da Computação

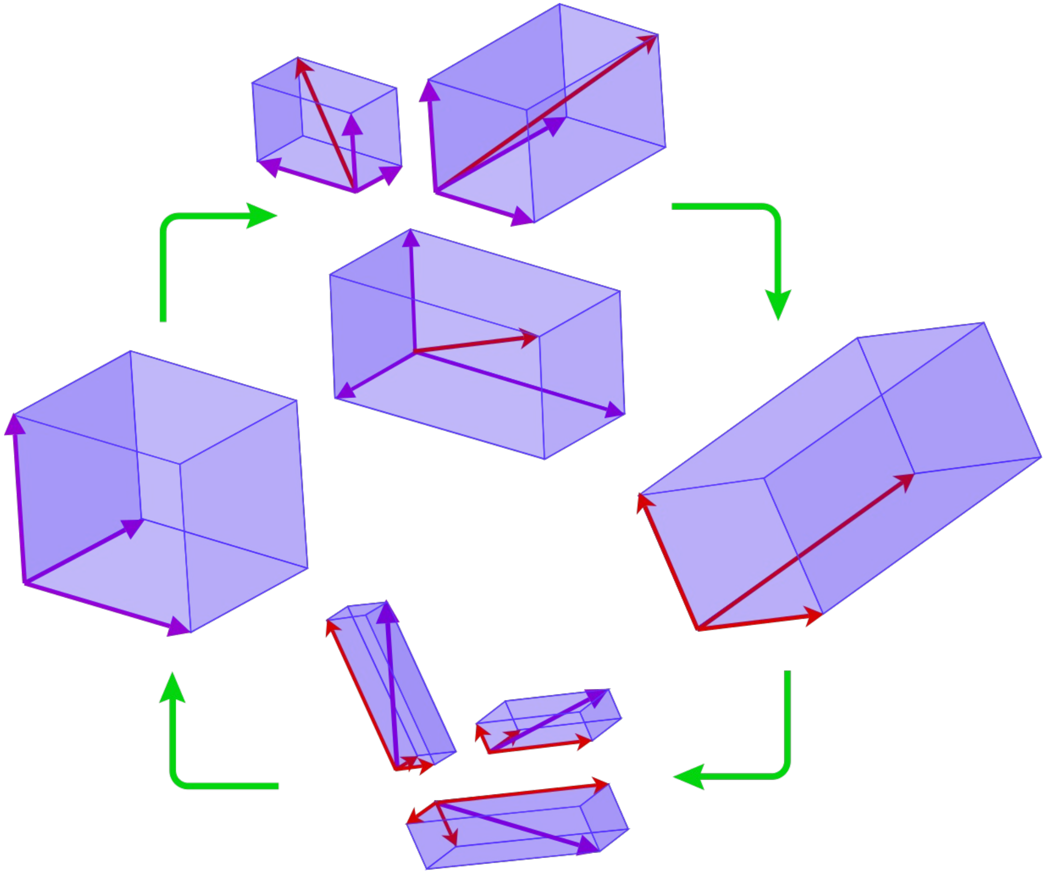
Prof. Walter T. Nakamura
waltertakashi@utfpr.edu.br

Campo Mourão - PR

Baseados nos materiais elaborados pelas professoras Aretha Alencar (UTFPR) e Rosane Minghim (USP)

Introdução

- Métodos para **transformações geométricas 3D** são extensões de **métodos 2D**, incluindo a coordenada z
- A translação e a escala são simples adaptações, mas a **rotação** é mais complexa
  - Em 2D somente são consideradas rotações em torno de um eixo perpendicular ao plano xy, em 3D pode-se pegar qualquer orientação espacial para o eixo de rotação
- Uma **posição 3D** expressa em coordenadas homogêneas é representada usando vetores coluna de 4 elementos, portanto as transformações 3D são matrizes 4 × 4



Transformações Geométricas 3D

Translação 3D

- Um objeto é movimentado **adicionando-se offsets** a cada uma das três direções Cartesianas:

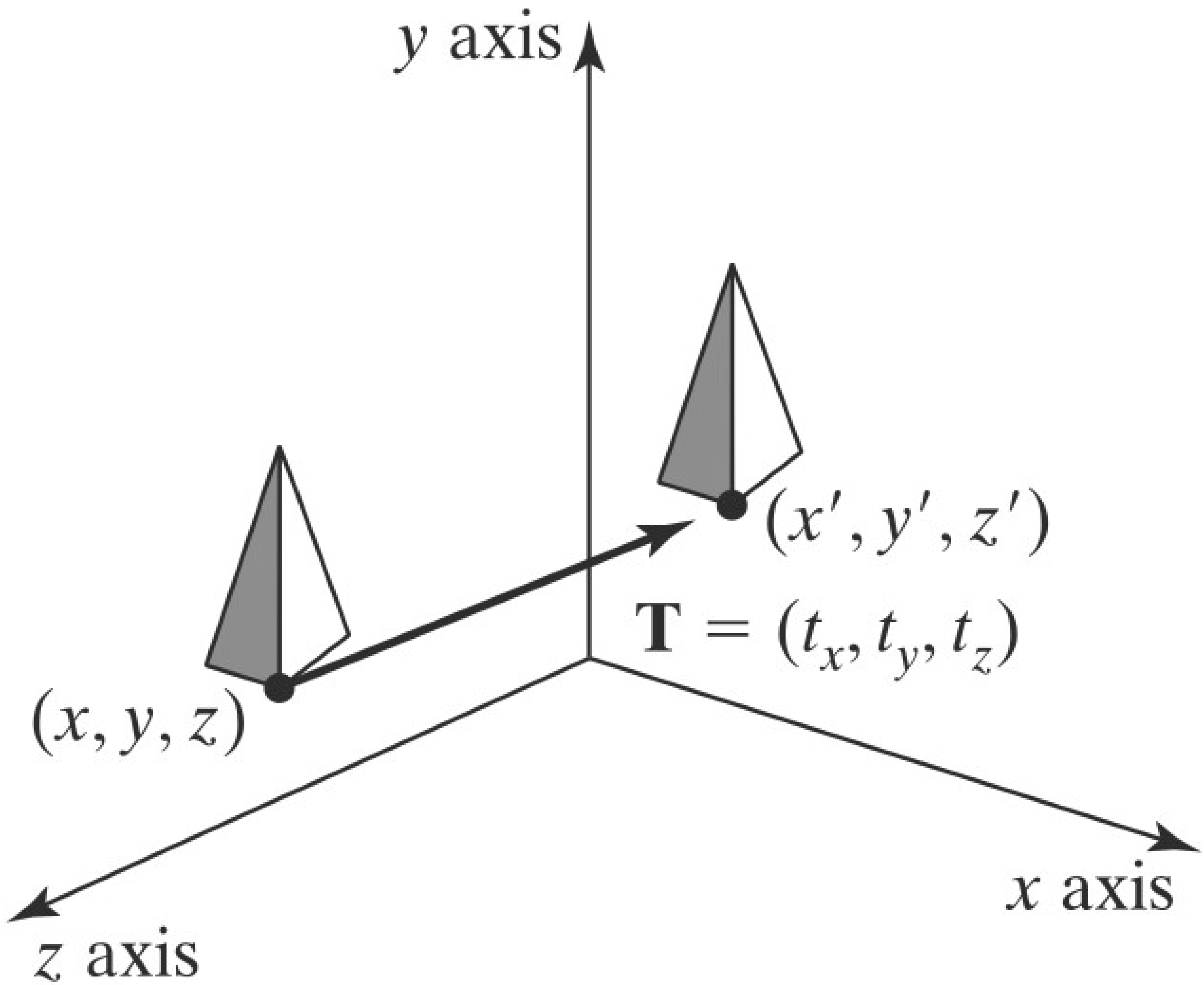
$$x' = x + t_x$$
$$y' = y + t_y$$
$$z' = z + t_z$$

- Representando matricialmente usando coordenadas homogêneas, temos:

$$\boldsymbol{P'} = \boldsymbol{T} \cdot \boldsymbol{P}$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Translação 3D



Translação 3D Inversa

- A translação inversa 3D é dada de forma semelhante a 2D, **negando os offsets** de translação:

$$\boldsymbol{T}^{-1}(t_x, t_y, t_z) = \boldsymbol{T}(-t_x, -t_y, -t_z)$$

$$\boldsymbol{T}^{-1}(t_x, t_y, t_z) = \begin{bmatrix} 1 & 0 & 0 & -t_x \\ 0 & 1 & 0 & -t_y \\ 0 & 0 & 1 & -t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Escala 3D

- A matriz de escala 3D é uma simples **extensão da 2D**, incluindo a variável z. Considerando os fatores de escala $s_x > 0$, $s_y > 0$ e $s_z > 0$, temos:

$$x' = x \cdot s_x$$
$$y' = y \cdot s_y$$
$$z' = z \cdot s_z$$

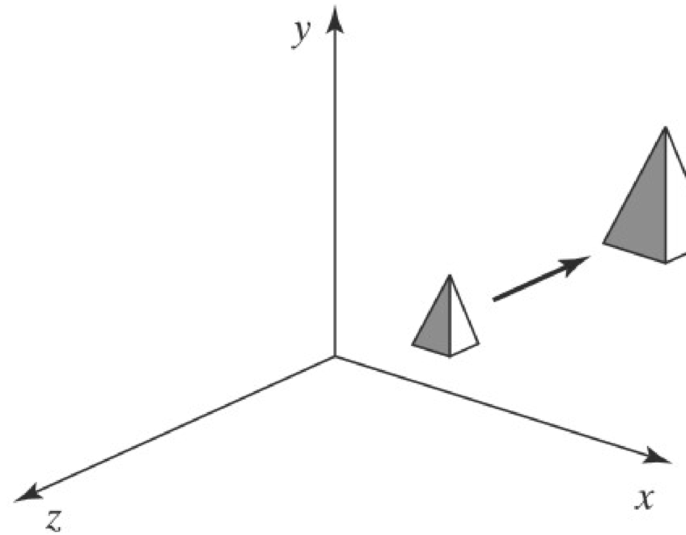
- Que definem a matriz de transformação:

$$\boldsymbol{P'} = \boldsymbol{S} \cdot \boldsymbol{P}$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Escala 3D

- Essa definição de escala muda a posição do objeto com relação a origem das coordenadas:
  - Valores > 1 afastam da origem;
  - Valores < 1 (exceto zero) aproximam da origem.
    - **Cuidado:** valores negativos fazem a reflexão sobre o eixo especificado.



- Se $s_x = s_y = s_z$, então temos uma escala uniforme, caso contrário o objeto apresenta escala diferencial.

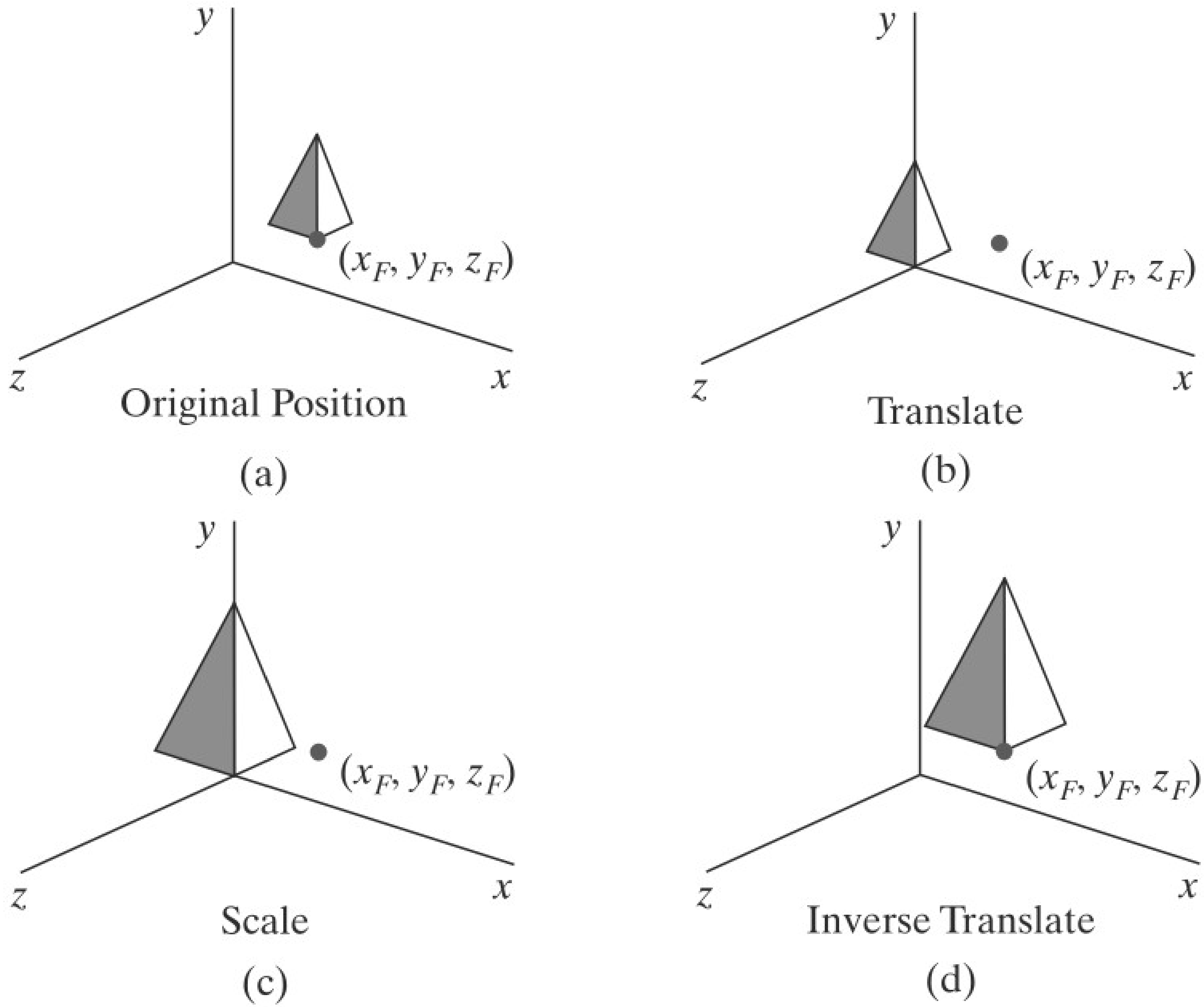
Escala 3D com Ponto Fixo

- Para se evitar esse problema podemos definir a escala com relação a uma posição fixa (x_f , y_f , z_f):
    1) Translado o ponto fixo para a origem;
    2) Aplico a transformação de escala;
    3) Translado o ponto fixo de volta a sua posição original.

$$\boldsymbol{P}'=\boldsymbol{T}(x_f, y_f, z_f)\cdot\boldsymbol{S}(s_x, s_y, s_z)\cdot\boldsymbol{T}(-x_f, -y_f, -z_f)\cdot\boldsymbol{P}$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}=\begin{bmatrix} 1 & 0 & 0 & x_f \\ 0 & 1 & 0 & y_f \\ 0 & 0 & 1 & z_f \\ 0 & 0 & 0 & 1 \end{bmatrix}.\begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.\begin{bmatrix} 1 & 0 & 0 & -x_f \\ 0 & 1 & 0 & -y_f \\ 0 & 0 & 1 & -z_f \\ 0 & 0 & 0 & 1 \end{bmatrix}.\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Escala 3D com Ponto Fixo



Escala 3D Inversa

- A matriz de escala inversa 3D é **trocando os fatores de escala** por seus inversos:

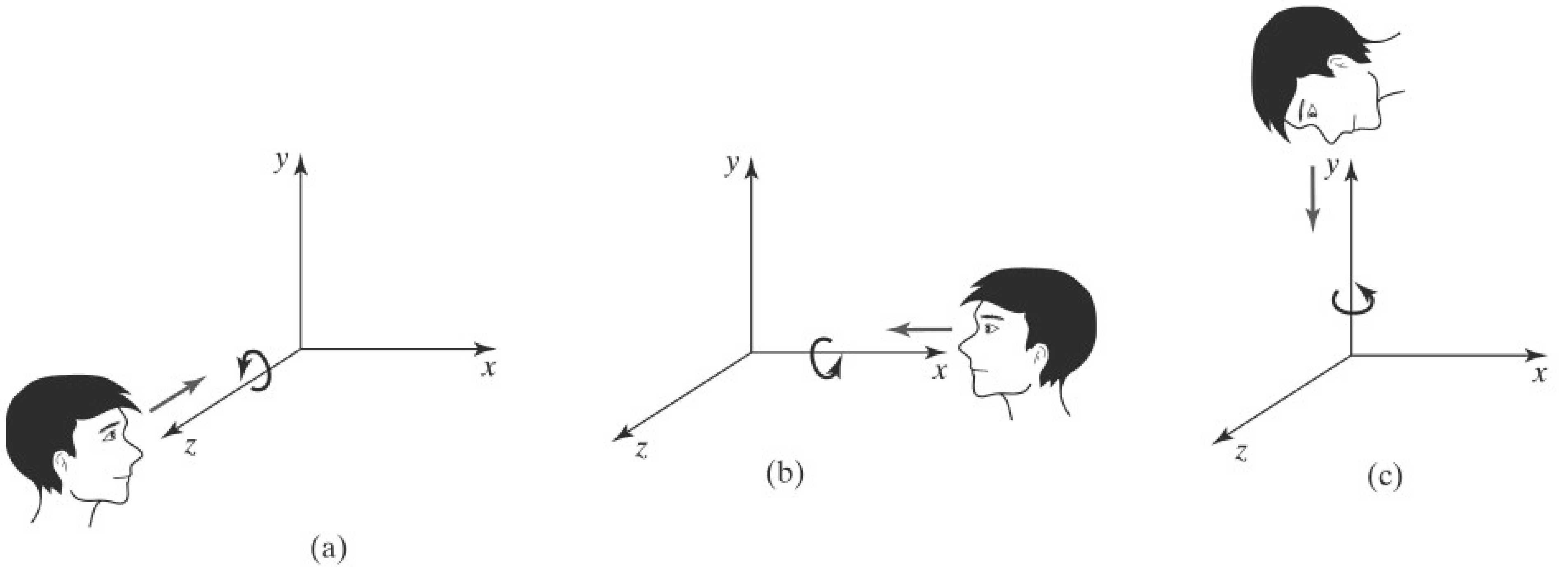
$$\boldsymbol{S}^{-1}(s_x, s_y, s_z) = \begin{bmatrix} 1/s_x & 0 & 0 & 0 \\ 0 & 1/s_y & 0 & 0 \\ 0 & 0 & 1/s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotação 3D

- É possível **rotacionar um objeto** ao redor de <u>qualquer eixo</u> no espaço 3D, porém, as rotações **mais fáceis** são executadas ao redor dos <u>eixos de coordenadas Cartesianas</u>
  - É possível combinar rotações em tornos dos eixos Cartesianos para se obter rotações em torno de qualquer eixo no espaço
- Por convenção, **ângulos positivos** produzem <u>rotações no sentido anti-horário</u>, assumindo que estamos olhando na direção negativa ao longo do eixo de rotação

Rotação 3D

- **Rotação positivas** sobre um eixo de coordenadas são <u>anti-horárias</u>, quando olhando a partir da parte positiva em direção a origem



Rotação 3D nos Eixos Coordenados

- Uma rotação 2D é facilmente estendida para uma rotação 3D ao redor do eixo z:

$$x' = x \cdot \cos(\theta) - y \cdot sen(\theta)$$

$$y' = x \cdot sen(\theta) + y \cdot \cos(\theta)$$

$$z' = z$$

- Na forma matricial, usando coordenadas homogêneas

$$\boldsymbol{P}' = \boldsymbol{R_z}(\theta) \cdot \boldsymbol{P}$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} cos\theta & -sen\theta & 0 & 0 \\ sen\theta & cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Rotação 3D nos Eixos Coordenados

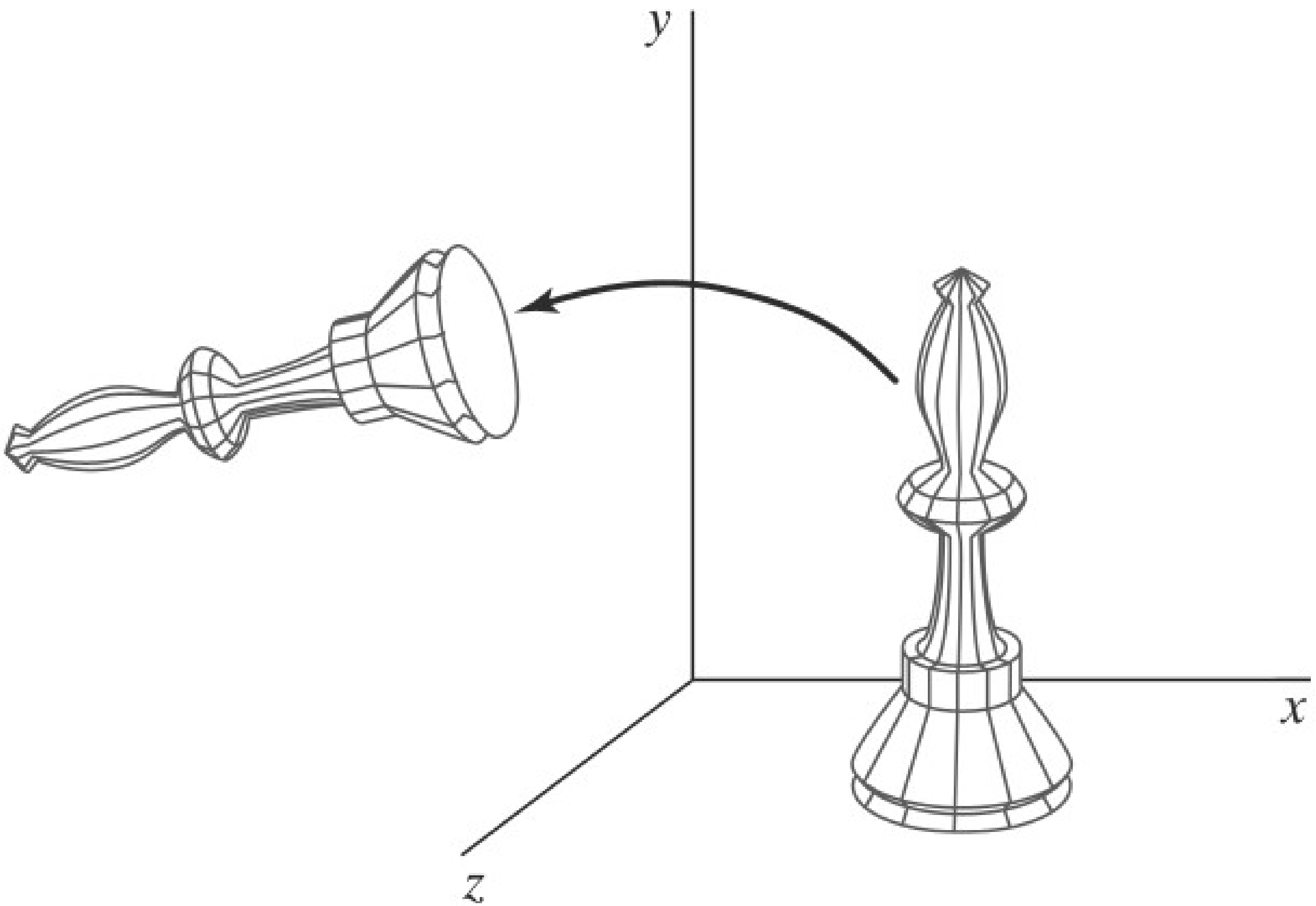
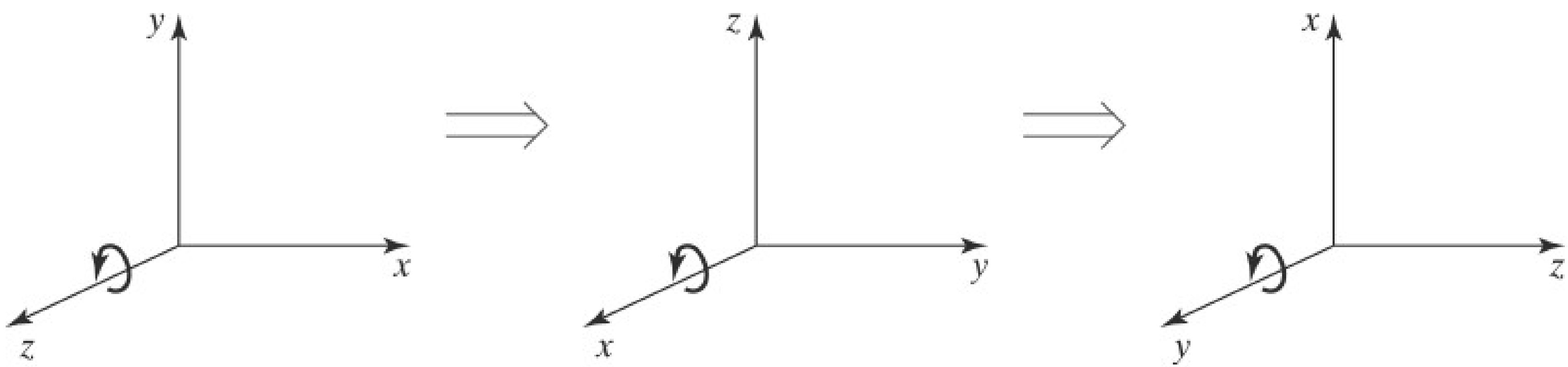


Figura: Rotação de um objeto em torno do eixo z.

Rotação 3D nos Eixos Coordenados

- As transformação de **rotação** para os <u>outros eixos de coordenadas</u> podem ser obtidas por meio de uma **permutação cíclica** das coordenadas x, y e z:

$$x \rightarrow y \rightarrow z \rightarrow x$$



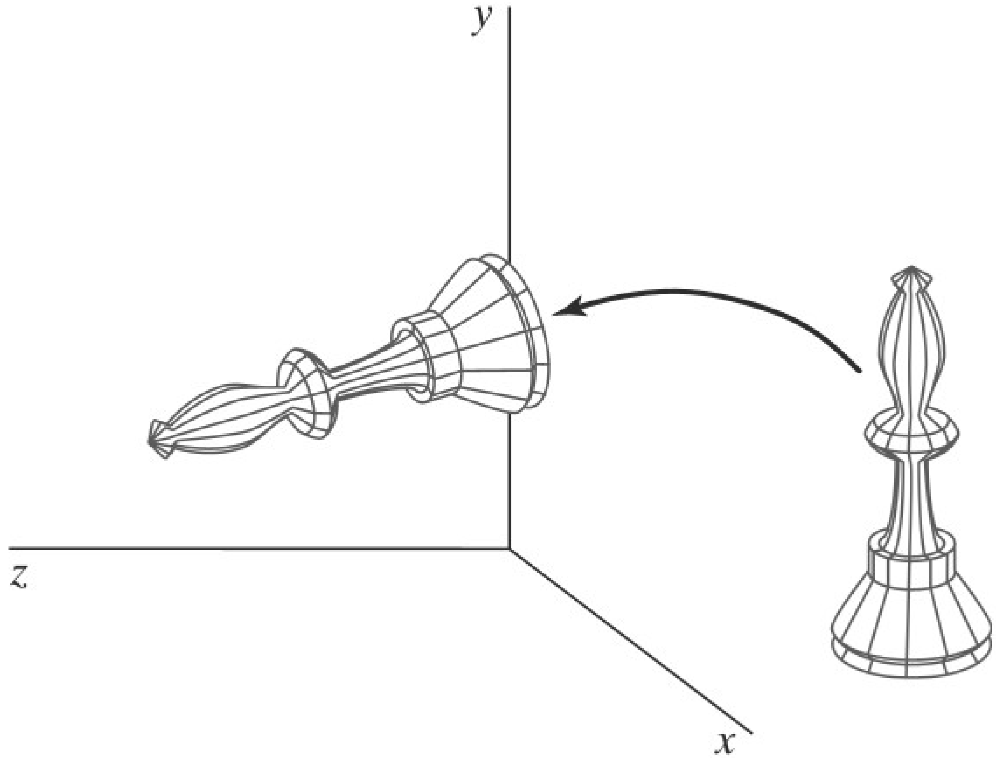
Rotação 3D nos Eixos Coordenados

- Considerando essa permutação e substituindo na equação da rotação 3D, compomos a rotação em torno do eixo x:

$$y' = y \cdot \cos(\theta) - z \cdot sen(\theta)$$

$$z' = y \cdot sen(\theta) + z \cdot \cos(\theta)$$

$$x' = x$$



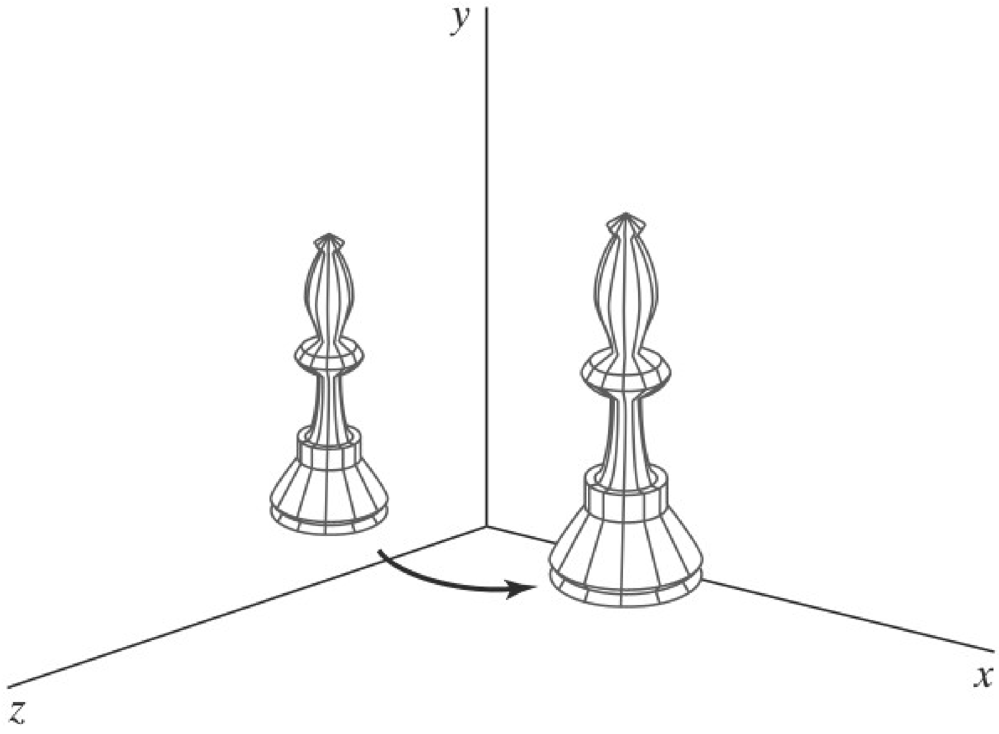
Rotação 3D nos Eixos Coordenados

- Considerando essa permutação e substituindo na equação da rotação 3D, compomos a rotação em torno do eixo y:

$$z' = z \cdot \cos(\theta) - x \cdot sen(\theta)$$

$$x' = z \cdot sen(\theta) + x \cdot \cos(\theta)$$

$$y' = y$$



Rotação 3D nos Eixos Coordenados

- Portanto as matrizes de rotação em torno dos eixos x, y e z são, respectivamente:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -sen\theta & 0 \\ 0 & sen\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & sen\theta & 0 \\ 0 & 1 & 0 & 0 \\ -sen\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -sen\theta & 0 & 0 \\ sen\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Exemplo

- Dado o ponto P = (4, 3, 7) rotacione-o em torno do eixo x em 60º.
  Dica: $\cos(60º) = 1/2$ e $\text{sen}(60º) = \sqrt{3}/2$

Exemplo

- Dado o ponto P = (4, 3, 7) rotacione-o em torno do eixo x em 60º.
  Dica: cos(60º) = 1/2 e sen(60º) = √3/2

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -sen\theta & 0 \\ 0 & sen\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/2 & -\sqrt{3/2} & 0 \\ 0 & \sqrt{3/2} & 1/2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 4 \\ 3 \\ 7 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ -4.562178 \\ 6.098076 \\ 1 \end{bmatrix}$$

Rotação 3D Inversa

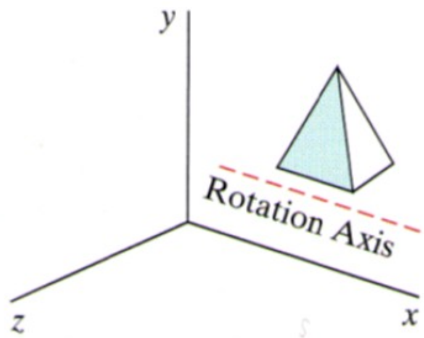
- A inversa de uma rotação é obtida trocando θ por −θ.
- Como somente o sinal do seno é alterado, a **inversa** pode ser obtida <u>trocando as linhas pelas colunas</u>, isto é $R^{-1} = R^T$
  - Exemplo:

$$R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\theta & -sen\theta & 0 \\ 0 & sen\theta & cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \Rightarrow R^T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\theta & sen\theta & 0 \\ 0 & -sen\theta & cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

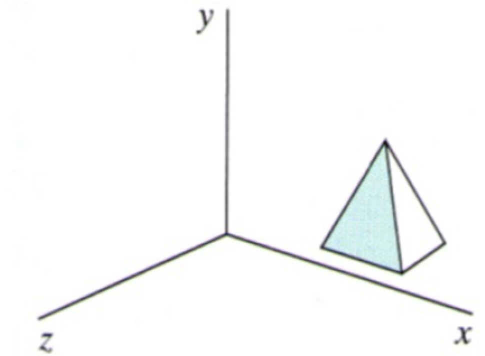
Rotação 3D Geral

- A **rotação** em torno de qualquer eixo pode ser obtida como a combinação de **rotações** e **translações**.
- No caso especial quando o **eixo de rotação** é paralelo a algum **eixo de coordenadas**, obtemos a rotação desejada fazendo:
  1) Translado o objeto de forma que o eixo de rotação coincida com o eixo paralelo de coordenadas;
  2) Executo a rotação;
  3) Translado o objeto de forma que o eixo de rotação é movido de volta a posição original.

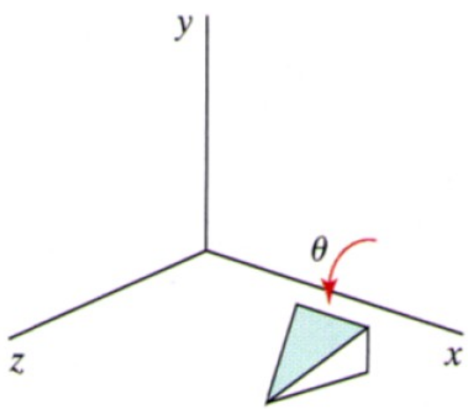
Rotação 3D Geral



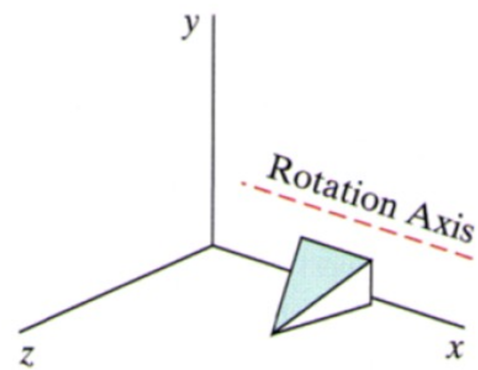
(a) Posição Original do Objeto



(b) Translade o eixo de rotação para coincidir com o eixo x



(c) Rotacione objeto usando o ângulo θ



(d) Translade o eixo de rotação para a sua posição original

Rotação 3D Geral

- Para um **eixo de rotação** paralelo ao eixo x, a sequência de transformações sobre um ponto P é dada por:

$$\boldsymbol{P}' = \boldsymbol{T} \cdot \boldsymbol{R}_x(\theta) \cdot \boldsymbol{T}^{-1} \cdot \boldsymbol{P}$$

- Ou seja, a matriz composta de rotação é:

$$\boldsymbol{R}(\theta) = \boldsymbol{T} \cdot \boldsymbol{R}_x(\theta) \cdot \boldsymbol{T}^{-1}$$

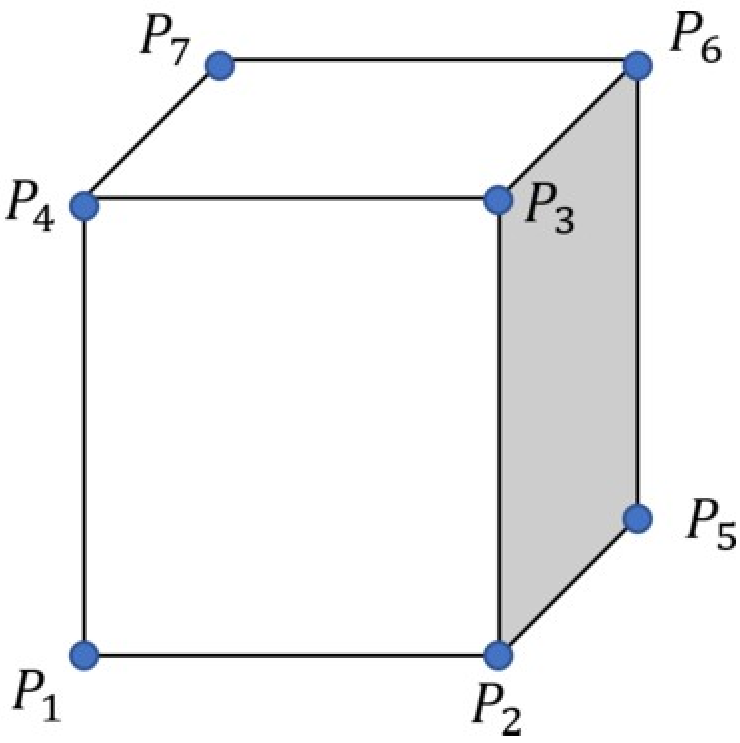
Rotação 3D Geral

- Quando o **eixo de rotação** não é **paralelo** aos eixos de coordenadas, algumas transformações adicionais são necessárias
  - São necessárias rotações para alinhar o eixo de rotação com o eixo de coordenadas escolhido e para trazer de volta o eixo de rotação para a posição original[1]

[1] Ver Apêndice ao final dos slides

Rotação 3D Geral - Exemplo

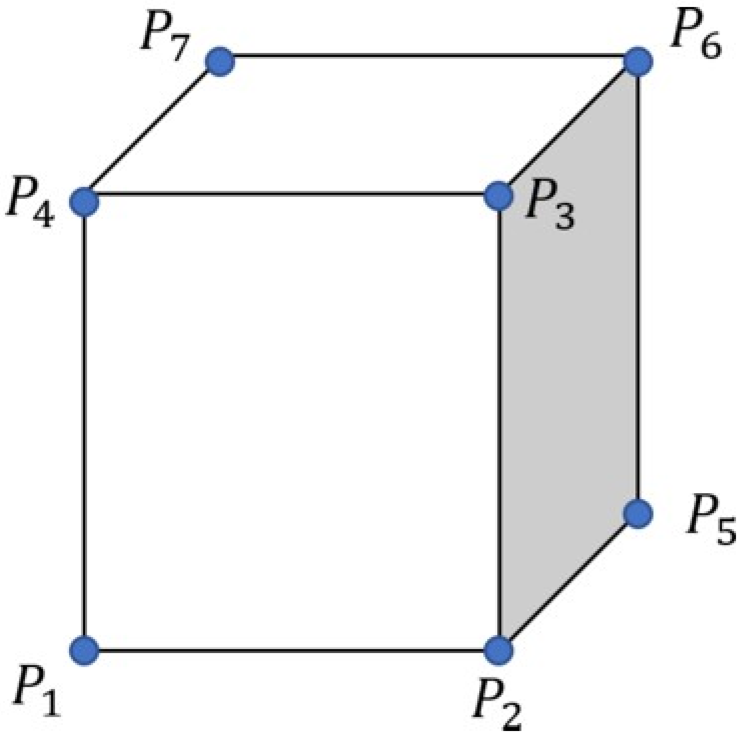
- Considere o objeto tridimensional abaixo, e a tabela de vértices que define a sua geometria:



$$P_1 = (4, -1, 2)$$
$$P_2 = (6, -1, 2)$$
$$P_3 = (6, 1, 2)$$
$$P_4 = (4, 1, 2)$$
$$P_5 = (6, -1, 0)$$
$$P_6 = (6, 1, 0)$$
$$P_7 = (4, 1, 0)$$

Rotação 3D Geral - Exemplo

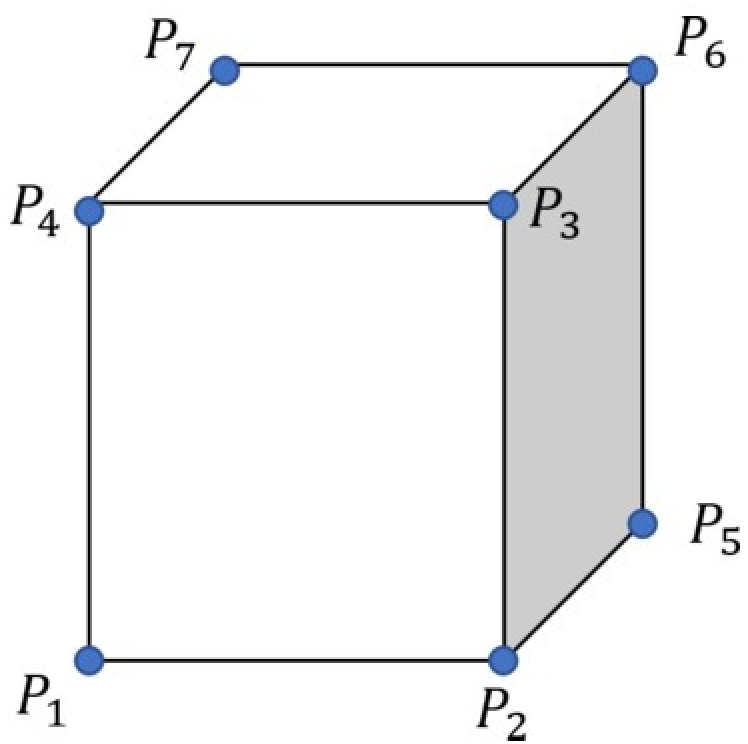
- Dê a matriz composta de transformação geométrica necessária para aplicar ao objeto uma rotação de 90º no sentido anti-horário em torno do eixo definido pela aresta P_3P_2
- Dê também coordenadas finais do ponto P_6 depois de aplicada essa rotação.



$$
\begin{aligned}
P_1 &= (4, -1, 2) \\
P_2 &= (6, -1, 2) \\
P_3 &= (6, 1, 2) \\
P_4 &= (4, 1, 2) \\
P_5 &= (6, -1, 0) \\
P_6 &= (6, 1, 0) \\
P_7 &= (4, 1, 0)
\end{aligned}
$$

Rotação 3D Geral - Exemplo

- Dê a matriz composta de transformação geométrica necessária para aplicar ao objeto uma rotação de 90º no sentido anti-horário em torno do eixo definido pela aresta P_3P_2
- Lembrando: $P_2 = (6,-1, 2)$ e $P_3 = (6, 1, 2)$. Dica: cos(90º) = 0 e sen(90º) = 1.



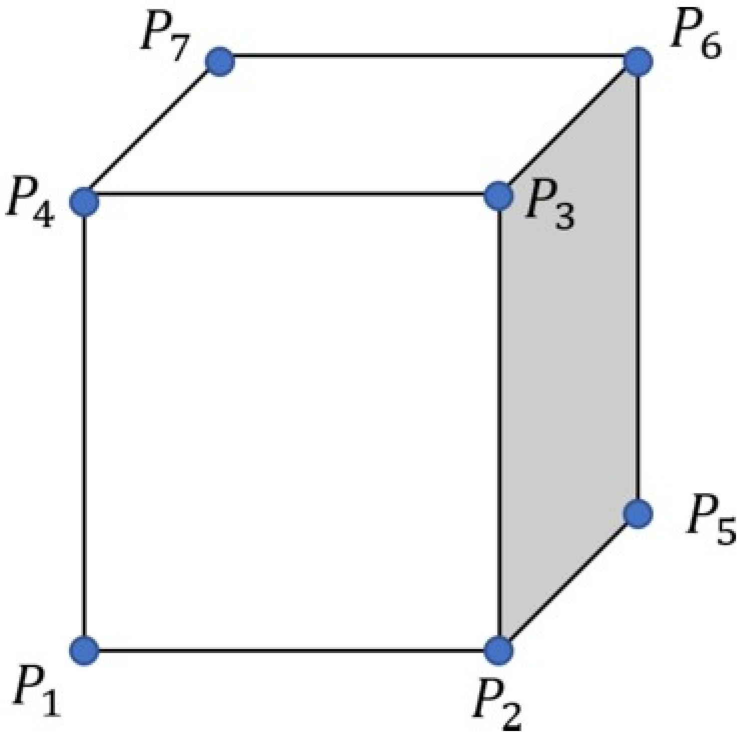
$$M = T(6,0,2) \cdot R_y(90^\circ) \cdot T(-6,0,-2)$$

$$M = \begin{bmatrix} 1 & 0 & 0 & 6 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -6 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M = \begin{bmatrix} 0 & 0 & 1 & 6 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -6 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotação 3D Geral - Exemplo

- Dê a matriz composta de transformação geométrica necessária para aplicar ao objeto uma rotação de 90º no sentido anti-horário em torno do eixo definido pela aresta P_3P_2

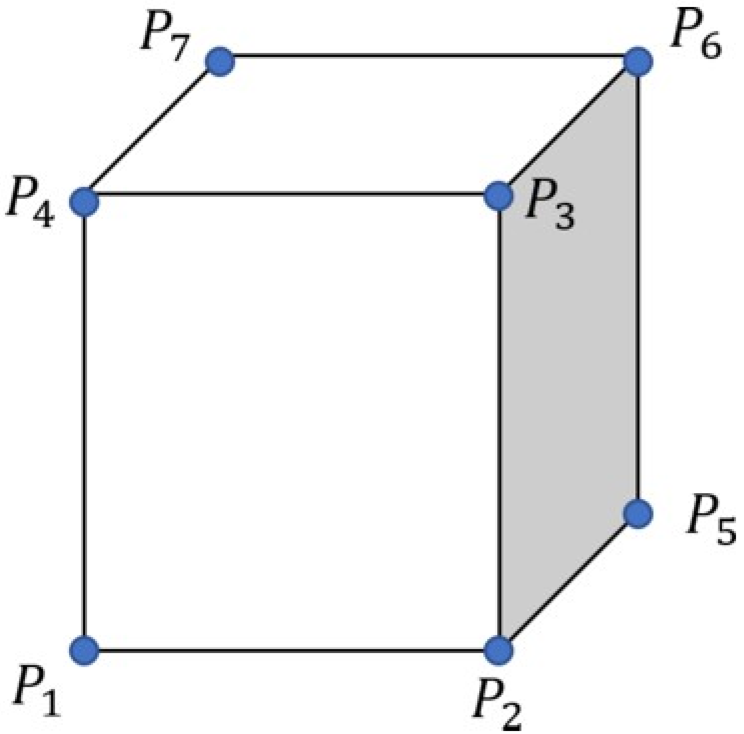


$$M = \begin{bmatrix} 0 & 0 & 1 & 6 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -6 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M = \begin{bmatrix} 0 & 0 & 1 & 4 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 8 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotação 3D Geral - Exemplo

- Dê também coordenadas finais do ponto P_6 depois de aplicada essa rotação.
- Lembrando: $P_6 = (6, 1, 0)$.



$$P'_6 = \begin{bmatrix} 0 & 0 & 1 & 4 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 8 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 6 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

$$P'_6 = \begin{bmatrix} 4 \\ 1 \\ 2 \\ 1 \end{bmatrix}$$

Reflexão 3D

- É semelhante a reflexão 2D: **rotação de 180º** sobre um eixo de rotação

$$M_{x_{reflect}} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad M_{y_{reflect}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad M_{z_{reflect}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Cisalhamento 3D

- É semelhante ao cisalhamento 2D: **distorcendo um objeto** em <u>direção de algum eixo</u>

$$M_{x_{cis.}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ sh_{xy} & 1 & 0 & 0 \\ sh_{xz} & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad M_{y_{cis.}} = \begin{bmatrix} 1 & sh_{yx} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & sh_{yz} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad M_{z_{cis.}} = \begin{bmatrix} 1 & 0 & sh_{zx} & 0 \\ 0 & 1 & sh_{zy} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$x' = x \qquad x' = x + sh_{yx} \cdot y \qquad x' = x + sh_{zx} \cdot z$$

$$y' = y + sh_{xy} \cdot x \qquad y' = y \qquad y' = y + sh_{zy} \cdot z$$

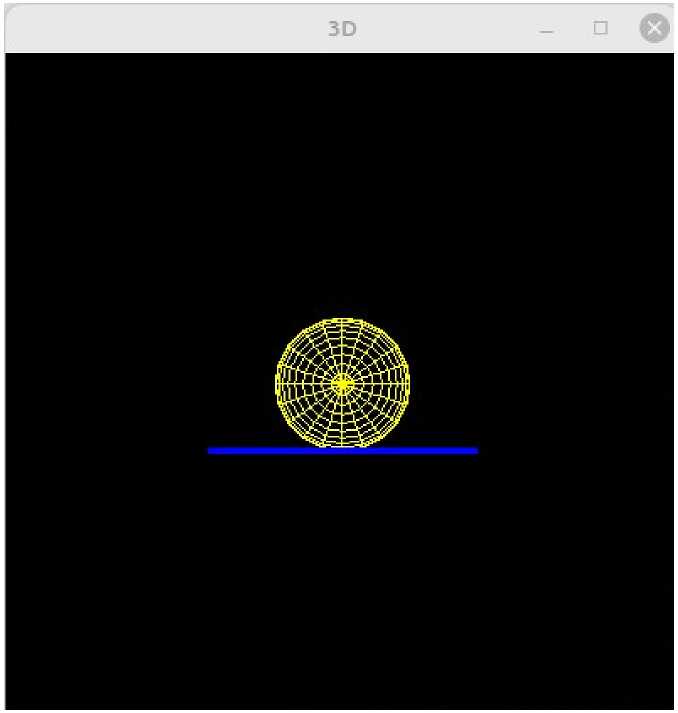
$$z' = z + sh_{xz} \cdot x \qquad z' = z + sh_{yz} \cdot y \qquad z' = z$$

Compondo Transformações 3D

- Assim como nas transformações 2D, as transformações 3D são **compostas** multiplicando matrizes
- Normalmente, a transformação **mais à direita** será a primeira a ser aplicada
- É necessário observar se a API gráfica utilizada é **pós-multiplicada** (da direita para a esquerda) ou **pré-multiplicada** (da esquerda para a direita)
  - **OpenGL** usa pós-multiplicação

Programação OpenGL – Exemplo 1

- Crie um programa em que seja desenhado uma esfera e um plano. O usuário poderá interagir com os objetos da seguinte forma:
  - Ao pressionar as teclas direcionais, os objetos deverão ser rotacionados;
  - Ao pressionar as teclas Page Up ou Page Down, os objetos devem ser ampliados ou reduzidos, respectivamente (zoom-in e zoom-out).
  - **Dicas:**
    - Utilize as funções:
      - glRotatef(angulo, x, y, z);
      - glScalef(x, y, z);



Programação OpenGL – Exemplo 1

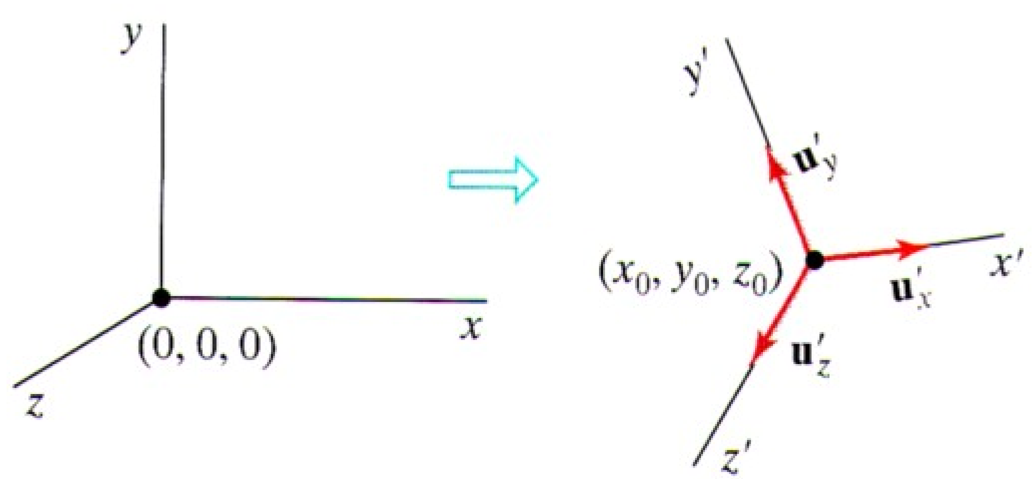
```
#include <stdio.h>
#include <GL/glut.h>

int init(void){
    glClearColor(0, 0, 0, 0);     // define a cor de fundo
    glEnable(GL_DEPTH_TEST);      // remoção de superfície oculta
    // define que a matriz é a de projeção
    glMatrixMode(GL_PROJECTION);

    glLoadIdentity();             // carrega a matriz identidade
    glOrtho(-5, 5, -5, 5, -5, 5); // define uma projeção ortográfica
}
```

Programação OpenGL – Exemplo 1

```
void display(void){
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glMatrixMode(GL_MODELVIEW); // define que a matriz é a de modelo
    glLoadIdentity();               // carrega a matriz identidade

    glColor3f(1, 1, 0);
    //desenha esfera wired (raio, slices, stacks)
    glutWireSphere(1.0f, 20, 20);

    glColor3f(0, 0, 1);
    glTranslatef(0, -1, 0);
    glScale(4, 0.1f, 4);
    glutSolidCube(1.0f); //desenha cubo sólido (tamanho do lado)

    glFlush();                      // desenha os comandos não executados
}

int main(int argc, char** argv){
    glutInit(&argc,argv);           //inicializa o GLUT
    //configura o modo de display
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);
    GlutInitWindowPosition(200,0); //seta a posição inicial da janela
    GlutInitWindowSize(400,400);   //largura e altura da janela
    glutCreateWindow("3D");         //cria a janela de exibição

    init();                         //executa função de inicialização
    glutSpecialFunc(keyPressed_special);
    glutDisplayFunc(display);
    glutMainLoop();                 //mostre tudo e espere
    return 0;
}
```

Transformações entre Sistemas de Coordenadas 3D

- Assim como em 2D, a transformação entre sistemas de coordenadas 3D se dá **sobrepondo** <u>sistemas de coordenadas diferentes</u>
- Para transformar um sistema Cartesiano de coordenadas xyz em outro x'y'z', dado que x'y'z' é definido com respeito a xyz fazemos:
  1) Translade a origem de x'y'z' para a origem de xyz;
  2) Execute uma sequencia de rotações para alinhar os eixos x'y'z' com xyz.



Transformações entre Sistemas de Coordenadas 3D

- Nesse exemplo, a origem de x'y'z' é sobreposta a origem de xyz transladando de:

$$T(-x_0, -y_0, -z_0)$$

- E a matriz de rotação pode ser obtida usando-se os vetores unitários u_x' , u_y' e u_z'

$$R = \begin{bmatrix} u'_{x1} & u'_{x2} & u'_{x3} & 0 \\ u'_{y1} & u'_{y2} & u'_{y3} & 0 \\ u'_{z1} & u'_{z2} & u'_{z3} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Então a transformação completa é dada por R .T

Transformações entre Sistemas de Coordenadas 3D – Exemplo

- Dado os eixos do sistema de coordenadas x'y'z': x' = (−1, 0, 0), y' = (0, 3, 0) e z' = (0, 0,−4). Dado também a origem do sistema de coordenadas P_0 = (0, 0,−5) de x'y'z'. Faça a transformação do ponto P = (2, 1, 1) do sistema de coordenadas xyz para o sistema x'y'z'.

$$u'_x = \frac{x'}{|x|} = \frac{(-1,0,0)}{\sqrt{(-1)^2+0^2+0^2}} = \frac{(-1,0,0)}{1} = (-1,0,0)$$

$$u'_y = \frac{y'}{|y|} = \frac{(0,3,0)}{\sqrt{0^2+3^2+0^2}} = \frac{(0,3,0)}{3} = (0,1,0)$$

$$u'_z = \frac{z'}{|z|} = \frac{(0,0,-4)}{\sqrt{0^2+0^2+(-4)^2}} = \frac{(0,0,-4)}{4} = (0,0,-1)$$

Transformações entre Sistemas de Coordenadas 3D – Exemplo

- Dado os eixos do sistema de coordenadas x'y'z': x' = (−1, 0, 0), y' = (0, 3, 0) e z' = (0, 0,−4). Dado também a origem do sistema de coordenadas P_0 = (0, 0,−5) de x'y'z'. Faça a transformação do ponto P = (2, 1, 1) do sistema de coordenadas xyz para o sistema x'y'z'.

$$T(-x_0, -y_0, -z_0) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Transformações entre Sistemas de Coordenadas 3D – Exemplo

- Dado os eixos do sistema de coordenadas x'y'z': x' = (−1, 0, 0), y' = (0, 3, 0) e z' = (0, 0,−4). Dado também a origem do sistema de coordenadas P_0 = (0, 0,−5) de x'y'z'. Faça a transformação do ponto P = (2, 1, 1) do sistema de coordenadas xyz para o sistema x'y'z'.

$$M = R \cdot T = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$P' = M \cdot P = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & -5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -2 \\ 1 \\ -6 \\ 1 \end{bmatrix}$$

$$P' = (-2, 1, -6)$$

Exercício 1

- Crie um programa em OpenGL que contém dois objetos um cubo e um bule que estão lado a lado
  - Se o usuário digitar a tecla ‘1’ ele estará selecionando o cubo
  - Se o usuário digitar a tecla ‘2’ ele estará selecionando o bule
- Ao pressionar as teclas ← e →, o objeto selecionado será rotacionado nos sentidos horário e anti-horário respectivamente, utilizando o centroide como ponto fixo
  - O cubo deve sempre rotacionar em torno do eixo <1,1,1>
  - O bule deve sempre rotacionar em torno do eixo <1,0,1>

Exercício 1

- Dica – Teclas não Especiais:

```
void keyboard (unsigned char key, int x, int y) {
    if (key == 1)
        // ação para a tecla 1
}
```

- Na função `main()`:

```
glutKeyboardFunc(keyboard);
```

- Desenho dos objetos:

```
void glutWireTeapot(GLdouble size);
void glutWireCube(GLdouble size);
```

Exercício 2

- Faça uma animação que simule o sistema solar contendo o Sol, a Terra e a Lua, sabendo que:
  - A Terra faz um giro completo ao redor do Sol em 365 dias (não considere anos bissextos);
  - A Lua gira completamente em torno de si mesma e em torno da Terra em 28 dias.
  - O Sol gira em torno do seu eixo em 27 dias.

Exercício 2

```
#include <stdio.h>
#include <GL/glut.h>

void Timer(int value){
    /*

    Aqui vai algum código para animar os objetos da cena

    */

    // atualiza a cena após a modificação dos objetos
    glutPostRedisplay();
    // chama a própria função para continuar animando os objetos
    glutTimerFunc(40, Timer, 1);
}


int init(void){
    glClearColor(0, 0, 0, 0);     // define a cor de fundo
    // define que a matriz é a de projeção
    glMatrixMode(GL_PROJECTION);

    glLoadIdentity();               // carrega a matriz identidade
    glOrtho(-5, 5, -5, 5, -5, 5); // define uma projeção ortográfica
}
```

Exercício 2

```
void display(void){
    //limpa o buffer
    glClear(GL_COLOR_BUFFER_BIT);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    GluLookAt(0.0,1.0,0.0,    //posição da câmera
              0.0,0.0,0.0,    //para onde a câmera aponta
              1.0,0.0,0.0);   //vetor view-up

    glColor4f(1.0, 1.0, 0.0, 1.0);
    GlutWireSphere(1,20,20);        // desenha o Sol

    glColor4f(0.0, 0.0, 1.0, 1.0);
    glTranslatef(4,0,0);
    GlutWireSphere(0.5,10,10);      // desenha a Terra

    glColor4f (0.81, 0.78, 0.79, 1.0);
    glTranslatef(1.0,0.0,0.0);
    GlutWireSphere(0.2,5,5);        // desenha a Lua

    glutSwapBuffers();
}
```

Exercício 2

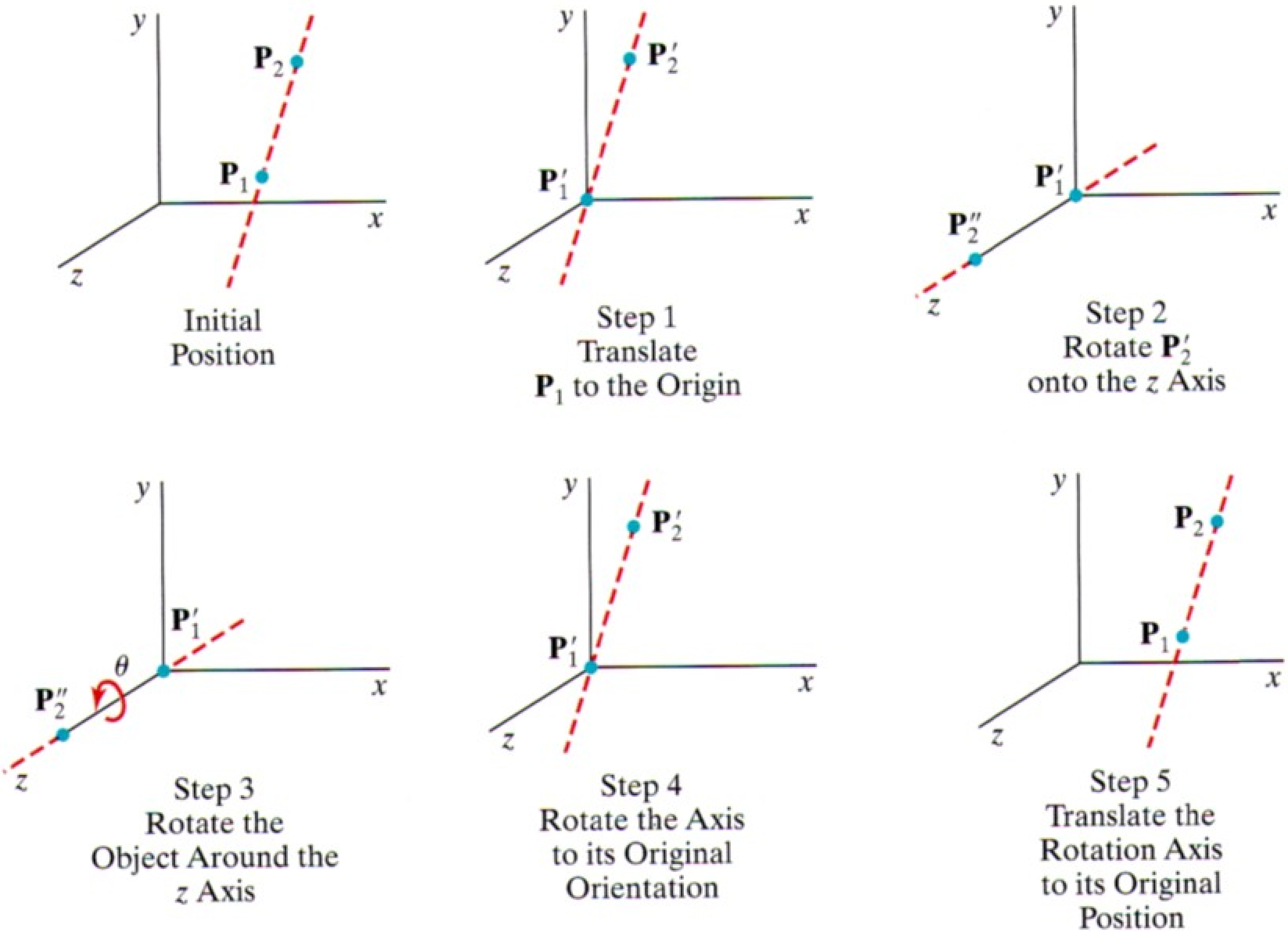
```
int main(int argc, char** argv){
    glutInit(&argc,argv);     //inicializa o GLUT
    //configura o modo de display
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
    //seta a posição inicial da janela
    glutInitWindowPosition(200,0);
    //configura a largura e altura da janela de exibição
    glutInitWindowSize(800,800);
    //cria a janela de exibição
    glutCreateWindow("Sistema Solar");

    init();                  //executa função de inicialização
    glutDisplayFunc(display);
    glutTimerFunc(40, Timer, 1);
    glutMainLoop();          //mostre tudo e espere
    return 0;
}
```

APÊNDICE

Rotação 3D Geral

- Quando o eixo de rotação não é paralelo aos eixos de coordenadas, algumas transformações adicionais são necessárias
  - São necessárias rotações para alinhar o eixo de rotação com o eixo de coordenadas escolhido e para trazer de volta o eixo de rotação para a posição original
  - Dado o eixo de rotação e o ângulo de rotação, isso pode ser feito com os seguintes passos:
    1) Transladar o objeto de forma que o eixo de rotação passe pela origem do sistema de coordenadas;
    2) Rotacionar o objeto para que o eixo de rotação coincida com um dos eixos de coordenadas;
    3) Realizar a rotação sobre o eixo de coordenadas escolhido;
    4) Aplicar a rotação inversa para trazer o eixo de rotação para sua orientação original;
    5) Aplicar a translação inversa para trazer o eixo de rotação para sua posição espacial original.
- Por conveniência, o eixo de coordenadas escolhido para o alinhamento normalmente é o eixo z.

Rotação 3D Geral



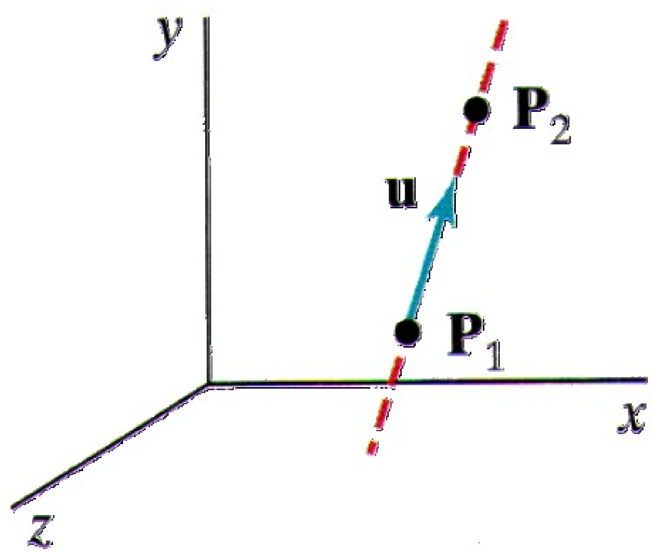
Rotação 3D Geral

- Assumindo que o eixo de rotação é definido por dois pontos (P_2 e P_1) e que a rotação se dá em sentido anti-horário em relação a esse eixo, podemos calcular suas componentes como[1]:

$$V = P_2 - P_1 = (x_2 - x_1, y_2 - y_1, z_2 - z_1)$$

- E o vetor unitário do eixo de rotação:

$$u = \frac{V}{|V|} = (a, b, c)$$

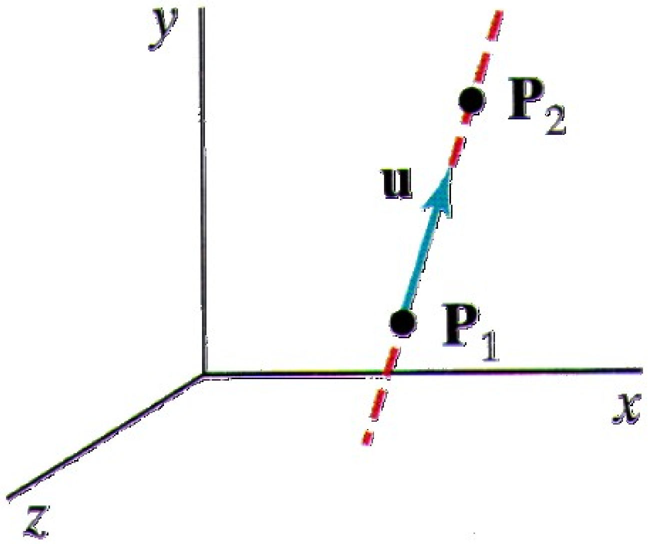


[1] Se a rotação fosse no sentido horário, então seria necessário inverter o vetor V e o vetor unitário u para que eles apontassem na direção de P_2 para P_1.

Rotação 3D Geral – Passo 1

- O primeiro passo da sequencia de rotação é definir uma matriz de translação para **reposicionar o eixo de rotação de forma que esse passe pela origem**:
  - Como a rotação se dá no sentido anti-horário, movemos o ponto P_1 para a origem[1], ou seja

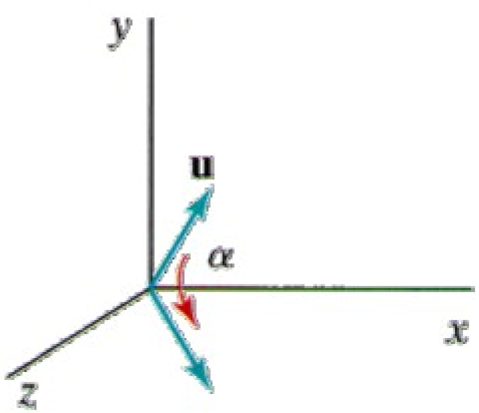
$$T = \begin{bmatrix} 1 & 0 & 0 & -x_1 \\ 0 & 1 & 0 & -y_1 \\ 0 & 0 & 1 & -z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



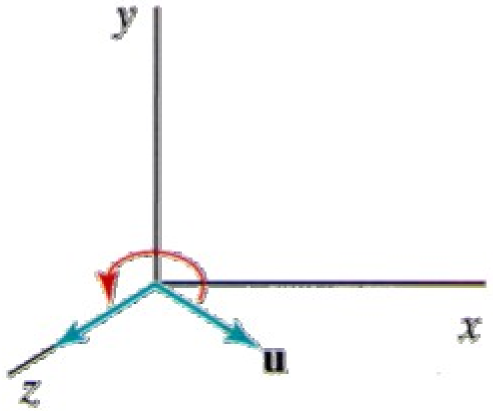
[1] Se a rotação fosse no sentido horário, então seria necessário mover P_2 para a origem.

Rotação 3D Geral – Passo 2

- Após isso, encontramos a **transformação que coloca o eixo de rotação sobre o eixo z**:
  - Existem várias maneiras de se realizar esse alinhamento, por exemplo, primeiro rotacionamos sobre o eixo x, depois sobre o eixo y
  - A rotação sobre o eixo x define o vetor u no plano xz, e a rotação no eixo y rotaciona u até sobrepor o eixo z.



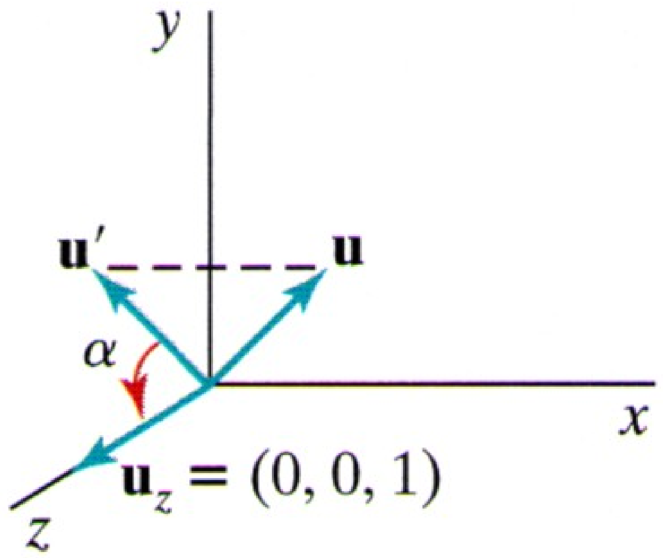
(a) Rotação sobre o eixo x



(b) Rotação sobre o eixo y

Rotação 3D Geral – Passo 2

- A **rotação em torno do eixo x** pode ser definida determinando os senos e cossenos do ângulo de rotação necessário para projetar u no plano xz
- Esse ângulo de rotação (α) é igual ao ângulo entre a projeção de u no plano yz com o eixo z positivo



Rotação 3D Geral – Passo 2

- Dada a projeção de u no plano yz, u' = (0, b, c), então o cosseno do ângulo de rotação α pode ser determinado a partir do produto escalar de u' com o vetor unitário u_z = (0, 0, 1) ao longo do eixo z:

$$\cos\alpha = \frac{u' \cdot u_z}{|u'||u_z|} = \frac{(0,b,c)\cdot(0,0,1)}{\sqrt{0^2+b^2+c^2}\times\sqrt{0^2+0^2+1^2}} = \frac{c}{d}$$

- Onde d é a magnitude de u':

$$d = \sqrt{b^2+c^2}$$

Rotação 3D Geral – Passo 2

- Similarmente é possível determinar o seno de α igualando a forma independente de coordenadas do produto vetorial

$$u' \times u_z = u_x |u'||u_z| \operatorname{sen} \alpha$$

- Usando a forma Cartesiana do produto vetorial[1] temos:

$$u' \times u_z = u_x \cdot b$$

$$u' \times u_z = u_x |u'||u_z| \operatorname{sen} \alpha = u_x \cdot b$$

- Como $|u_z| = 1$ e $|u'| = d$, então:

$$\operatorname{sen} \alpha = \frac{b}{d}$$

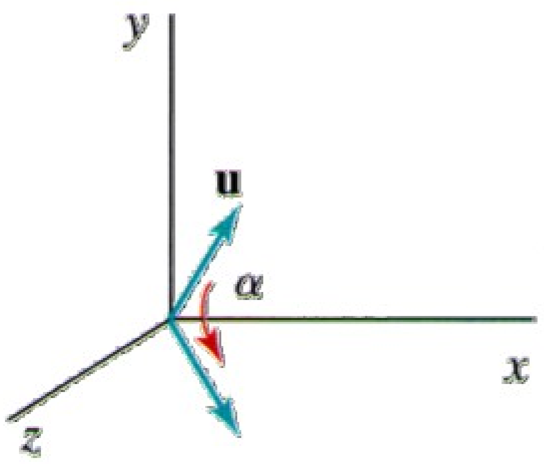
[1]Ver Apêndice (slide 78)

Rotação 3D Geral – Passo 2

- Com os senos e cossenos determinados (sen α = b/d e cos α = c/d), podemos definir a matriz para a rotação de u sobre o eixo x no plano xz:

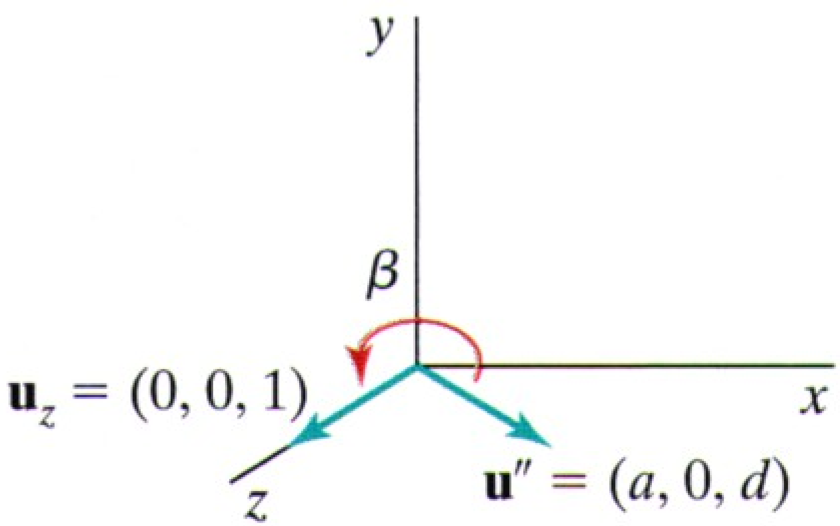
$$\mathbf{R_x}(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\operatorname{sen}\alpha & 0 \\ 0 & \operatorname{sen}\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c/d & -b/d & 0 \\ 0 & b/d & c/d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Rotação 3D Geral – Passo 2

- O próximo passo é determinar a matriz de rotação que vai rotacionar o vetor unitário u'' (resultante da rotação anterior) no plano xz em torno do eixo y até sobrepor o eixo z:
  - Como u = (a, b, c), então u'' = (a, 0, d) pois a rotação em torno do eixo x não altera a coordenada x, a coordenada y é zerada pela projeção no plano xz e a coordenada z = d porque |u''| = |u|



Rotação 3D Geral – Passo 2

- Com isso podemos novamente encontrar os senos e cossenos do ângulo β fazendo:

$$\cos\beta = \frac{u'' \cdot u_z}{|u''||u_z|} = \frac{(a,0,d)\cdot(0,0,1)}{\sqrt{a^2+0^2+d^2}\cdot\sqrt{0^2+0^2+1^2}}$$

- Como $|u''| = |u_z| = 1$:

$$\cos\beta = d$$

Rotação 3D Geral – Passo 2

- Igualando a forma independente de coordenadas do produto vetorial:

$$u'' \times u_z = u_y |u''||u_z| \operatorname{sen} \beta$$

- Com a forma Cartesiana[1]:

$$u'' \times u_z = u_y \cdot (-a)$$

$$u'' \times u_z = u_y |u''||u_z| \operatorname{sen} \beta = u_y \cdot (-a)$$

- Como $|u_z| = |u''| = 1$, então:

$$\operatorname{sen} \beta = -a$$

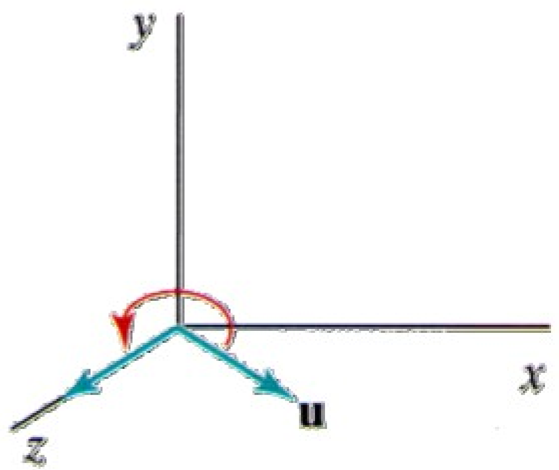
[1]Ver Apêndice (slide 79)

Rotação 3D Geral – Passo 2

- Com os senos e cossenos determinados (sen β = −a e cos β = d), a matriz de rotação de u'' sobre o eixo y é:

$$\mathbf{R_y}(\beta) = \begin{bmatrix} \cos\beta & 0 & \operatorname{sen}\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\operatorname{sen}\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} d & 0 & -a & 0 \\ 0 & 1 & 0 & 0 \\ a & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Rotação 3D Geral – Passo 3

- Com essas rotação em α e β nós alinhamos o eixo de rotação sobre o eixo z, então agora a rotação com um ângulo θ sobre o eixo z finalmente pode ser aplicada:

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

Rotação 3D Geral – Passos 4 e 5 (Completo)

- Assim, a matriz de rotação completa sobre um eixo arbitrário fica:

$$\mathbf{R}(\theta) = \mathbf{T}^{-1} \cdot \mathbf{R}_{\mathbf{x}}^{-1}(\alpha) \cdot \mathbf{R}_{\mathbf{y}}^{-1}(\beta) \cdot \mathbf{R}_{\mathbf{z}}(\theta) \cdot \mathbf{R}_{\mathbf{y}}(\beta) \cdot \mathbf{R}_{\mathbf{x}}(\alpha) \cdot \mathbf{T}$$

Exemplo

- Dar a matriz de rotação em termo de um eixo arbitrário em 3D dado por P_1P_2 no sentido anti-horário em 90º. Dados $P_1 = (2, 2, 2)$ e $P_2 = (6, 6, 6)$. Dica: cos 90º = 0 e sen 90º = 1.

Exemplo

- Dar a matriz de rotação em termo de um eixo arbitrário em 3D dado por P_1P_2 no sentido anti-horário em 90º. Dados P_1 = (2, 2, 2) e P_2 = (6, 6, 6). Dica: cos 90º = 0 e sen 90º = 1.

$$V = P_2 - P_1 = (6-2, 6-2, 6-2) = (4, 4, 4)$$

$$u = \frac{V}{|V|} = \frac{(4,4,4)}{\sqrt{4^2+4^2+4^2}} = \frac{(4,4,4)}{\sqrt{16 \times 3}} = \left(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right) = (a, b, c)$$

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & -2 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & -2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Exemplo

$$d = \sqrt{b^2 + c^2} = \sqrt{\left(\frac{1}{\sqrt{3}}\right)^2 + \left(\frac{1}{\sqrt{3}}\right)^2} = \sqrt{\frac{2}{3}} = \frac{\sqrt{2}}{\sqrt{3}}$$

$$\cos\alpha = \frac{c}{d} = \frac{1}{\sqrt{3}} \cdot \frac{\sqrt{3}}{\sqrt{2}} = \frac{1}{\sqrt{2}}$$

$$\operatorname{sen}\alpha = \frac{b}{d} = \frac{1}{\sqrt{3}} \cdot \frac{\sqrt{3}}{\sqrt{2}} = \frac{1}{\sqrt{2}}$$

$$\mathbf{R_x}(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\operatorname{sen}\alpha & 0 \\ 0 & \operatorname{sen}\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Exemplo

$$\cos\beta = d = \frac{\sqrt{2}}{\sqrt{3}}$$

$$\operatorname{sen}\beta = -a = -\frac{1}{\sqrt{3}}$$

$$\mathbf{R_y}(\beta) = \begin{bmatrix} \cos\beta & 0 & \operatorname{sen}\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\operatorname{sen}\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \sqrt{2}/\sqrt{3} & 0 & -1/\sqrt{3} & 0 \\ 0 & 1 & 0 & 0 \\ 1/\sqrt{3} & 0 & \sqrt{2}/\sqrt{3} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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

Exemplo

$$\mathbf{R}(\theta) = \mathbf{T}^{-1} \cdot \mathbf{R_x}^{-1}(\alpha) \cdot \mathbf{R_y}^{-1}(\beta) \cdot \mathbf{R_z}(\theta) \cdot \mathbf{R_y}(\beta) \cdot \mathbf{R_x}(\alpha) \cdot \mathbf{T}$$

$$\mathbf{R}(\theta) = \mathbf{T}^{-1} \cdot \mathbf{R_x^T}(\alpha) \cdot \mathbf{R_y^T}(\beta) \cdot \mathbf{R_z}(\theta) \cdot \mathbf{R_y}(\beta) \cdot \mathbf{R_x}(\alpha) \cdot \mathbf{T}$$

$$\mathbf{R}(\theta) \approx \begin{bmatrix} 0.3333333 & -0.2440169 & 0.9106836 & 0 \\ 0.9106836 & 0.3333333 & -0.2440169 & 0 \\ -0.2440169 & 0.9106836 & 0.3333333 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Produto Vetorial na forma Cartesiana para u' x u_z

$$u' \times u_z = (0, b, c) \times (0, 0, 1)$$

$$u' \times u_z = \begin{vmatrix} \mathbf{u_x} & \mathbf{u_y} & \mathbf{u_z} \\ 0 & b & c \\ 0 & 0 & 1 \end{vmatrix}$$

$$= \left| \begin{matrix} \mathbf{u_x} & \mathbf{u_y} & \mathbf{u_z} \\ 0 & b & c \\ 0 & 0 & 1 \end{matrix} \right| \begin{matrix} \mathbf{u_x} & \mathbf{u_y} \\ 0 & b \\ 0 & 0 \end{matrix} \right|$$

$$= [(\mathbf{u_x} \cdot b \cdot 1) + (\mathbf{u_y} \cdot c \cdot 0) + (\mathbf{u_z} \cdot 0 \cdot 0)]$$
$$- [(0 \cdot b \cdot \mathbf{u_z}) + (0 \cdot c \cdot \mathbf{u_x}) + (1 \cdot 0 \cdot \mathbf{u_y})]$$
$$= \mathbf{u_x} \cdot b$$

Produto Vetorial na forma Cartesiana para u'' x u_z

$$u'' \times u_z = (a, 0, d) \times (0, 0, 1)$$

$$u'' \times u_z = \begin{vmatrix} \mathbf{u_x} & \mathbf{u_y} & \mathbf{u_z} \\ a & 0 & d \\ 0 & 0 & 1 \end{vmatrix}$$

$$= \left|\begin{matrix} \mathbf{u_x} & \mathbf{u_y} & \mathbf{u_z} \\ a & 0 & d \\ 0 & 0 & 1 \end{matrix}\right|\begin{matrix} \mathbf{u_x} & \mathbf{u_y} \\ a & 0 \\ 0 & 0 \end{matrix}\right|$$

$$= [(\mathbf{u_x} \cdot 0 \cdot 1) + (\mathbf{u_y} \cdot d \cdot 0) + (\mathbf{u_z} \cdot a \cdot 0)]$$

$$- [(0 \cdot 0 \cdot \mathbf{u_z}) + (0 \cdot d \cdot \mathbf{u_x}) + (1 \cdot a \cdot \mathbf{u_y})]$$

$$= \mathbf{u_y} \cdot (-a)$$