

QUESTÃO 2 (2 valores)

Admita que temos as seguintes funções de mistura Bézier.

$$B_0 = 2u$$

$$B_1 = u^2 - u + 1$$

$$B_2 = u^2 - u \quad \text{com } u \in [0, 1].$$

Sendo os três pontos de controlo $P_0 = (1, 0, 0)$, $P_1 = (1, 0, 1)$, $P_2 = (-1, 1, 0)$ calcule o ponto $P = (p_x, p_y, p_z)$ da curva Bézier em $u = 1$

R:

2)

Vamos recorrer ao cálculo matricial referente a Bézier de grau 2 (quadráticas), que são controladas por 3 pontos, tal que:

$$p(u) = UMP \text{ em que } U = [u^2 \ u \ 1]; \quad M = \begin{bmatrix} 0 & 1 & 1 \\ 2 & -1 & -1 \\ 0 & 1 & 0 \end{bmatrix} \text{ e } P = [P_0 \ P_1 \ P_2]^T$$

Assim vem que:

$$p(u) = [u^2 \ u \ 1] \cdot \begin{bmatrix} 0 & 1 & 1 \\ 2 & -1 & -1 \\ 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} P_0 \\ P_1 \\ P_2 \end{bmatrix} = [u^2 \ u \ 1] \cdot \begin{bmatrix} 0 & 1 & 1 \\ 2 & -1 & -1 \\ 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix}$$

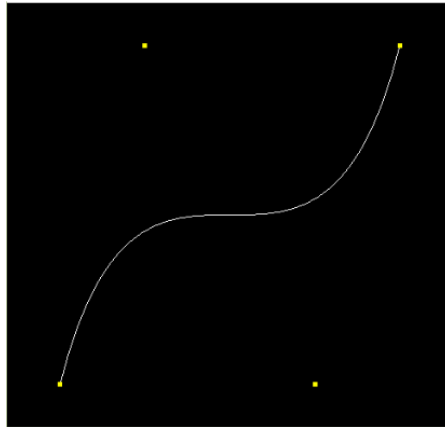
Se substituirmos por $u = 1$ vem:

Veja-se que esta expressão permite obter qualquer ponto da curva com $u \in [0, 1]$. Seja $u = 0.5 = \frac{1}{2}$

$$\begin{aligned} p(u=1) &= [1^2 \ 1 \ 1] \cdot \begin{bmatrix} 0 & 1 & 1 \\ 2 & -1 & -1 \\ 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix} = \\ &= [1 \ 1 \ 1] \cdot \begin{bmatrix} 0 & 1 & 1 \\ 2 & -1 & -1 \\ 1 & 0 & 1 \end{bmatrix} = \\ &= [3, 0, 1] \end{aligned}$$

QUESTÃO 3 (6 valores)

Codifique **apenas** as funções/métodos *init()* e *display()* de um programa em JOGL que gere o resultado visual ilustrado. Deverá criar uma curva de Bézier, definindo devidamente os pontos de controlo. Comente o código devidamente.



R:

```

public void init(GLAutoDrawable drawable)
{
    GL gl = drawable.getGL();
    glu = new GLU();
    glut = new GLUT();
    for (int i = 0; i < ctrlpoints.length; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            ctrlpointBuf.put(ctrlpoints[i][j]);
        }
    }
    ctrlpointBuf.rewind();
    gl.glClearColor(0.0f, 0.0f, 0.0f, 0.0f);
    gl.glShadeModel(GL.GL_FLAT);
    gl.glMap1f(GL.GL_MAP1_VERTEX_3, 0.0f, 1.0f, 3, 4, ctrlpointBuf);
    gl.glEnable(GL.GL_MAP1_VERTEX_3);
}

public void display(GLAutoDrawable drawable)
{
    GL gl = drawable.getGL();

    gl.glClear(GL.GL_COLOR_BUFFER_BIT);
    gl.glColor3f(1.0f, 1.0f, 1.0f);
    gl.glBegin(GL.GL_LINE_STRIP);
    for (int i = 0; i <= 30; i++)
    {
        gl.glEvalCoord1f((float) i / (float) 30.0);
    }
    gl.glEnd();
    gl.glPointSize(5.0f);
    gl.glColor3f(1.0f, 1.0f, 0.0f);
    gl.glBegin(GL.GL_POINTS);
    for (int i = 0; i < 4; i++)
    {
        gl.glVertex3fv(ctrlpointBuf);
        ctrlpointBuf.position(i * 3);
    }
    gl.glEnd();
    gl.glFlush();
}

```