Bezier Curves

Octave Project 3

# Description of the problem:

The mathematical problem we are trying to solve with these methods is interpolating 2D or 3D curves using a set of control points. Bezier curves are defined as a linear combination of Bernstein polynomials as follows. At the end we are using those control points to create the Bernstein polynomials that define the curve.

# Explanation of the method:

**Direct Evaluation:**

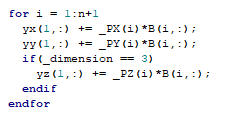
Given points we will compute the Bernstein polynomials where is , we will have a mesh of t values in . Finally compute the Bezier curve . Normally we will compute the curve first and then compute for each of the t values. In the code we first compute the values of the Bernstein polynomials by plugging the t values for each one of them.

A picture containing text

Description automatically generated

Figure

In *Figure 1* we can se the computation of each of the Bernstein polynomials, in the code the t represents an array of values from .



Figure

In *Figure 2* we are computing .

**De Casteljau:**

Given P0, P1 … Pn points in or the Bezier curve can be computed constructing a regular mesh of m + 1 nodes in [0, 1] and then applying linear interpolation recursively on for each node. The recursion is noted as follows:

Texto, Carta

Descripción generada automáticamenteAnd in code it looks like this

**Midpoint:**

# Examples:

**Direct Evaluation:**

**De Casteljau:**

Given .

For :

For :

Gráfico, Gráfico de líneas

Descripción generada automáticamenteFor :

Gráfico, Gráfico de líneas

Descripción generada automáticamenteOutput of the program compared with the one obtained by hand.

# Bibliography:

MAT300 Lecture Notes: lecture11.