

## גיאומטריה חישובית 2

תרגיל 1

B-Spline to Bezier conversion

מגיש: אליאס קואינקה 015257231

## Program Description

The program reads and displays uniform and non-uniform B-Spline curves from the following file format:

```
# A cubic B-Spline curve
# The number of control points
5
# The knot vector
0,1,2,3,4,5,6,7,8
# The control points (x,y)
0.0 -3.0
2.0 3.0
4.0 5.0
6.0 5.0
8.0 -5.0
```

Comments can be added with the # prefix and the program will ignore them when reading the file.

The program then calculates the B-Spline curve by first finding the basis using the Cox-deBoor algorithm in a recursive function:

```
void COpenGLControl::findBasis(int d)
```

The function calls itself, each time with a successive degree  $d$ , until the degree of the function we are looking for (in this case, cubic) is reached.

Calculation of the curve is done by plugging numbers in the formula

$$Q(u) = \sum_{i=0}^k B_i(u)P_i$$

where the quantity of numbers depends on the level of detail desired (defined within the program).

Conversion to the Bezier curve is done by taking each segment of the B-Spline curve and comparing 4 points (at  $u=0$ ,  $u=1$  and two in the middle) so as to find the control points that correspond to the Bezier representation.

The image can be zoomed by clicking and dragging with the right mouse button, and translated on the x-y plane by clicking and dragging with the middle mouse button. The button labeled *Center Image* returns to the original view.

The curve displays points where each segment begins and ends, and there's options to see the curves  $x(u)$ , and  $y(u)$ . These two last are scaled to the knot vector, so they are impervious to change when zooming or translating.

The program was written in Visual C++ with the OpenGL and MFC libraries.

## Screenshots



