Curves Assignment - Lab Assignment 2

Instructor: Gershon Elber

T.A.: Haitham Fadila

Handed out: July 8th, 2024 **Due date:** July 29th, 2024

Description

Develop an interactive graphics program that will compute and display Bezier and B-spline curves. Your program should allow the user to display the data (both the curve and its control polygon) and edit it. You are to provide methods to add, delete and modify the control points. The curves should be redrawn accordingly. You also should provide a way to change only the weight of a control point (a possible nice way, is by editing the radius of a circle around the control point, which represents the weight at the control point).

For B-spline curves, you should provide a *graphical mechanism* to edit the knot vector. The user should be able to specify the end conditions of a B-spline curve (open end or floating end conditions), as well as the degree of the curve to be created.

Note: The responsibility of the consistency between number of knots and number of control points for a given degree is of the user - therefore if a knot (or a control point) is added/removed, your program should not draw a curve until the user makes appropriate changes to resume consistency. However, if the knot vector has a special character - uniform float or uniform open end - you should automatically fix the knot vector if the control polygon is changed, such that the knot vector retains this special property.

You should be able to handle more than one curve at a time, and to connect two Bezier/B-spline curves with C^0, C^1, and G^1 continuity into a single B-spline via modifications of the second curve only.

Selection

You should also implement *one of the* following additional options:

- Implement a single knot insertion for B-spline curves.
- Add support to display differential geometry properties (using code from your first lab): Tangent, normal, and osculating circle (curvature). You may consider the evaluation of the derivatives numerically, at a point.
- Implement derivative computations for Bezier/B-spline curves and show the Hodograph (derivative) curve along with the original. Allow the user to select a point on the curve and the corresponding point on the Hodograph will be highlighted as well.
- Raise the degree of a Bezier curve and subdivide a linear/quadratic/cubic Bezier curve into two curves at the parameter $t \in [-1, 2]$. Note subdivision outside [0, 1] should work equally well but will extrapolate the shape.

Data file format

Your program should be able to read in data files in the following format: an integer specifying the order (= number of points if it is a Bezier curve), followed by a knot vector if a B-spline curve as *knots[n]* = k_1, ..., k_n, followed by the control points themselves as pairs or triplets, specifying *(x, y {, w})*. Each control point will appear on a separate line. Blank lines should be skipped. Lines beginning with a pound sign, #, should be treated as comments. Comment lines may appear in the body of the data file. Here is an example:

```
# A cubic Bezier curve.

4
0.000000 -3.000000
2.000000 -1.000000
2.000000 1.000000
0.000000 3.000000
```

In the files in which *w* is specified, the given *x* and *y* coordinates are the original coordinates AFTER they have been multiplied by *w*. Display the rational circle example (circle.dat) to check that your code fits this convention (i.e., the circle looks circular...). Note there can be multiple curves in a single file, having the curves one after the other.

Provide the capability to save your resulting data into a data file. You are also required to **contribute one interesting data file** of your own and this save capability will make it easier for you to create this data file.

There are sample data files here (Bezier) and here (B-Spline).

Submission

As always, we will schedule a time to grade the program with you.

Summary

- Load data files of (piecewise) rational and polynomial 2D curve(s) descriptions;
- Evaluate and display Bezier curves;
- Evaluate and display B-spline curves;
- Allow for a choice of open-end or floating end conditions of a B-spline curve.
- Allow setting the degree of the B-spline curve to be created.
- Display the control polygons of both types of curves;
- Allow modifications (position and weight) of existing control points.
- Allow the insertion and deletion of control points;
- Any modification should be interactive and interactively displayed;
- For B-spline curves, allow graphical editing of the knot vector: insert, delete, modify;
- Allow the handling of multiple curves simultaneously in the scene.
- Select and implement one of the additional options.

Tips

- You can download an exe example here. **In case of a discrepancy between the exe file and this document, this document governs.**