

DM6122

Assignment 2 (Plotting B-spline curves)

One of the good ways to develop a working knowledge of B-spline techniques is to actually write programming codes. This assignment is to write a program using C programming language and OpenGL (or any other programming language), which can plot an arbitrary degree B-spline curve. The program accepts an input file as follows, which defines a B-spline curve:

```
degree
cnt_num
u0 u1 u2 ...
x0 y0
x1 y1
...
```

where “degree” is the degree of the B-spline curve, “cnt_num” is the number of the control points, “u0, u1, ...” are the knot sequence of the curve, “x0 y0” are the x- and y-coordinates of the first control point, “x1 y1” are the x- and y-coordinates of the second control point, and so on.

Below “cubic.txt” is an example file of a cubic B-spline curve with 4 control points.

cubic.txt ---

```
3
4

0  0.5  1  2  3  3.5  4.5  5

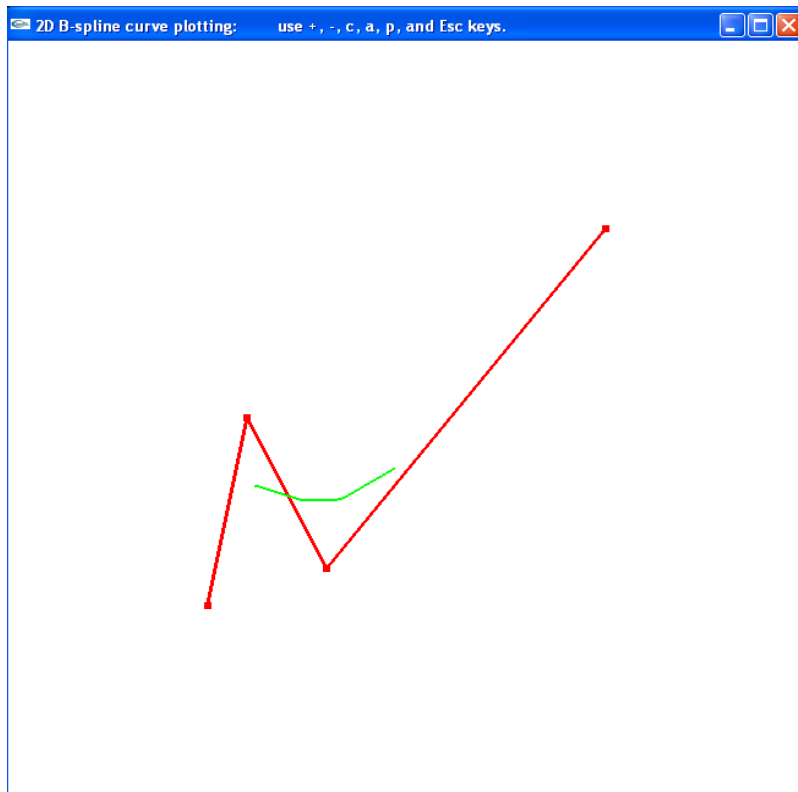
0  0
10 50
30 10
100 100
```

The program will display both the control polygon and the curve. The curve will be plotted using two rendering methods: uniform and adaptive, as explained below.

(1). With the uniform method, the program tessellates the curve by evenly sampling the parameter in the parameter domain of the curve. The program uses the number of the sampling points to control the tessellation.

(2). With the adaptive method, the program first converts each curve segment of the B-spline curve into a Bezier curve and then adaptively tessellates the Bezier curve using the method given in the lecture (see Module 3. Bezier curves). The program uses the approximation tolerance to control the tessellation.

The following figure is a snapshot of the result when the program loads “cubic.txt”.



Your program should at least have the following features:

1. The student's name should be displayed in the title bar of the application window.
2. Key “c” will switch between the control polygon and no control polygon
3. Key “p” will toggle displaying sampling points
4. Key “a” will switch between adaptive and uniform rendering modes.
5. Key “+” will increase the number of tessellation line segments.
6. Key “-” will decrease the number of tessellation line segments.
7. Key “ESC” will exit the program.

The implementation of other features such as menu or toolbar is also encouraged.

Submission:

- Copy your source **source code (.c file)**, the **.exe file**, and a readme.txt file if necessary, into folder “assignment2” that you created in your first assignment.
- Create two B-spline curves of degrees 3 and 5, respectively, with at least 15 knots and store them in two files “deg3Bspline.txt” and “deg5Bspline.txt” in a format

similar to that of the above “cubic.txt”. You should use your program to display the two curves. Take a snapshot of your curves displayed together with control polygons and save the image as the .jpeg format. Try to make the curves to define some nice shapes. Also copy the two B-spline files and the two images into folder “assignment2”.

- Zip the whole folder that contains “assignment1” and “assignment2” and upload your zipped folder to **DM6122-Projects** in the **Content/Assignments** of the NTULearn course-site.

Submission deadline: 1 November 2016 (Tuesday)

IN CLASS on 1 November 2016 (Tuesday), also hand in a hardcopy of the two images of your two curves.

The assessment criteria:

- 20 points for the two B-spline curves
- 70 points for the B-spline plotting program with the requirement functions/features.
- 10 points for other extra features

To simplify your work, you can download “assignment2Supp.zip”, which contains some useful files for this assignment. In particular,

- “demo.exe” is a demo file for your reference.
- “sample.c” is a template file. You can start with it, modify the codes and add your own codes to finish the assignment.
- “cubic.txt”, “cubic1.txt”, and “quartic.txt” are some sample data files that you can use to test your program.