CPSC 478/578
HW #4
Due October 26, 2015, 11:55pm

Note that the requirements for each question may vary depending on whether you are registered for 478 or for 578. The areas addressed in this assignment are splines and subdivision.

Turn-in Procedure

You should submit your work as a zip file using the classesv2 server. Please name your file as LastNameFirstName-Assignment3.zip
When your file is unzipped there should be subdirectories for each question named q1, q2, etc. Name your files as directed in each question. In each directory you should have:
1. The HTML and Javascript programs you have written, or pdf's of your written response (either typed directly or scanned in). For code, you should use files in the form of the samples given, rather than producing files from scratch. This will help us follow your code.
2. If the question asks you to write code to make images, provide sample images created by your program. You can save these by clicking and saving results in your browser, or by taking a screenshot.
3. A readme.{txt, doc} that lists the input used to create the images you include. You should also list the operating system (e.g. Linux, Windows 7, 8.1, 10, Mac OS 10.4.4) and browser (e.g. Firefox 40.0.2, Safari, IExplorer, Edge) that you used. If you programs fail on the machines used for grading, you may be asked to bring in your system to demonstrate that the files you submitted functioned in the environment you worked in.

1. No programming required for this question.
(478 and 578) a) Find the tangent and normal for a cubic Bezier segment defined by the points (2,5), (3, 7), (4,5), (5,0) and the point where the spline segment has an x value that is the averate of the x values at the endpoints of the segment.

(478 an 578) b) The de Casteljau algorithm can be expressed as producing a new sequence of 8 control points $P'_o, P'_1, \ldots P'_7$ from an original set of 4 points P_o, P_1, P_2, P_3 by multiplying by S defined as:

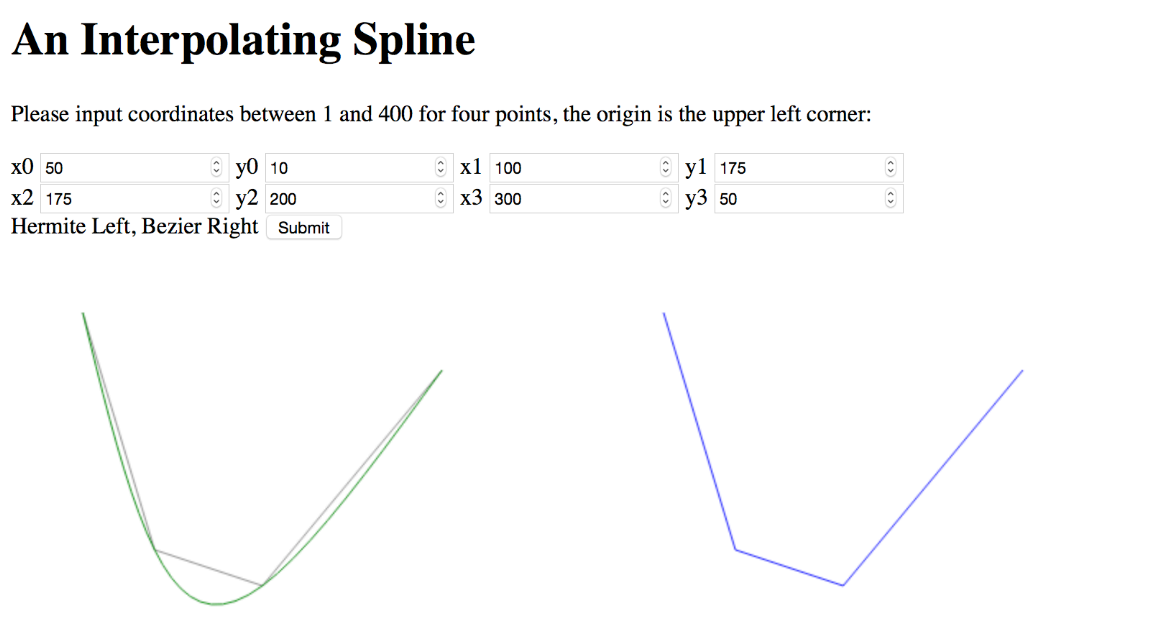
$$S = \left\{ \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} & 0 \\ \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8} \end{pmatrix}, \begin{pmatrix} \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8} \\ 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix} \right\}.$$

i. Show that all 8 new points are within the convex hull defined by original 4 points.
ii. Show that the two new segments have C^2 continuity independent of the values of the original 4 points.

(578) Choose four 2D points that a define Catmull-Rom spline segment. Give a value of u and the corresponding point P on the spline segment that is not in the convex hull of the original 4 points. Explain how you know that the point is not in the convex hull.

2. **(478 and 578)** The file spline-interp.html (which uses file tridiagonal.js) takes as input 4 points and draws the linear spline and Hermite cubic spline that interpolate the points on the canvas on the left (see image below). Modify the code to also compute the control points for the Bezier curve segments that define the same spline. Draw lines connecting the Bezier control points, in order, in blue on the canvas on the right (initially the blue line is just the linear spline). Also draw the Bezier curve itself in black on the right, using the HTML5 built in function bezierCurveTo
http://www.html5canvastutorials.com/tutorials/html5-canvas-bezier-curves/

(578) Create a new version of the code for drawing the Hermit curve that adaptively determines the steps to take in for delta u to make the curve draw to look smooth. Your code should take a parameter that controls how smooth the curve will be (i.e. if the parameter has one extreme value the individual line segments will be clearly visible, if it has another extreme value it will look smooth. In the readme file for this problem, explain the rule you use to determine how large a step in u you can take.



An Interpolating Spline

Please input coordinates between 1 and 400 for four points, the origin is the upper left corner:

x0 50 y0 10 x1 100 y1 175
x2 175 y2 200 x3 300 y3 50
Hermite Left, Bezier Right Submit

3. **(478 and 578)** Write a program that when given a set of 16 three dimensional control points computes 25 points on the corresponding cubic B-Spline patch, and then displays the patch as a set of 32 triangles connecting the computed surface points. For input, the user only needs to fill in the 16 zij values below to define the points.

(0, 0, z00), (1, 0, z10), (2,0,z20), (3,0,z30)
(0, 1, z01), (1, 1, z11), (2,1,z21), (3,1,z31)
(0, 2, z02), (1, 2, z12), (2,2,z22), (3,2,z32)
(0, 2, z03), (1, 3, z13), (2,3,z23), (3,3,z33)

The patch should be displayed in the same way the tetrahedron and sphere primitives were in Assignment 2. Provide images of two views of the output of your program for a patch defined by the following control points:

(0, 0, 0), (1, 0, 0), (2,0,0), (3,0,0)
(0, 1, 0), (1, 1, 2), (2,1,2), (3,1,0)
(0, 2, 0), (1, 2, 2), (2,2,2), (3,2,0)
(0, 3, 0), (1, 3, 0), (2,3,0), (3,3,0)

(578) Write a variation of the program that takes four coordinates at each point (i.e. include the fourth w coordinate) and displays the NURB surface that is defined by the points.

4. **(478 and 578)** Write a program that takes as input four 3D coordinates of a tetrahedron and a number of iterations N, and displays the mesh that results after applying N iterations of Loop subdivision to the tetrahedron.