CS 3451 Project 05 – COON PATCH

For this assignment you will be implementing Coons Patch between two 2D curves.

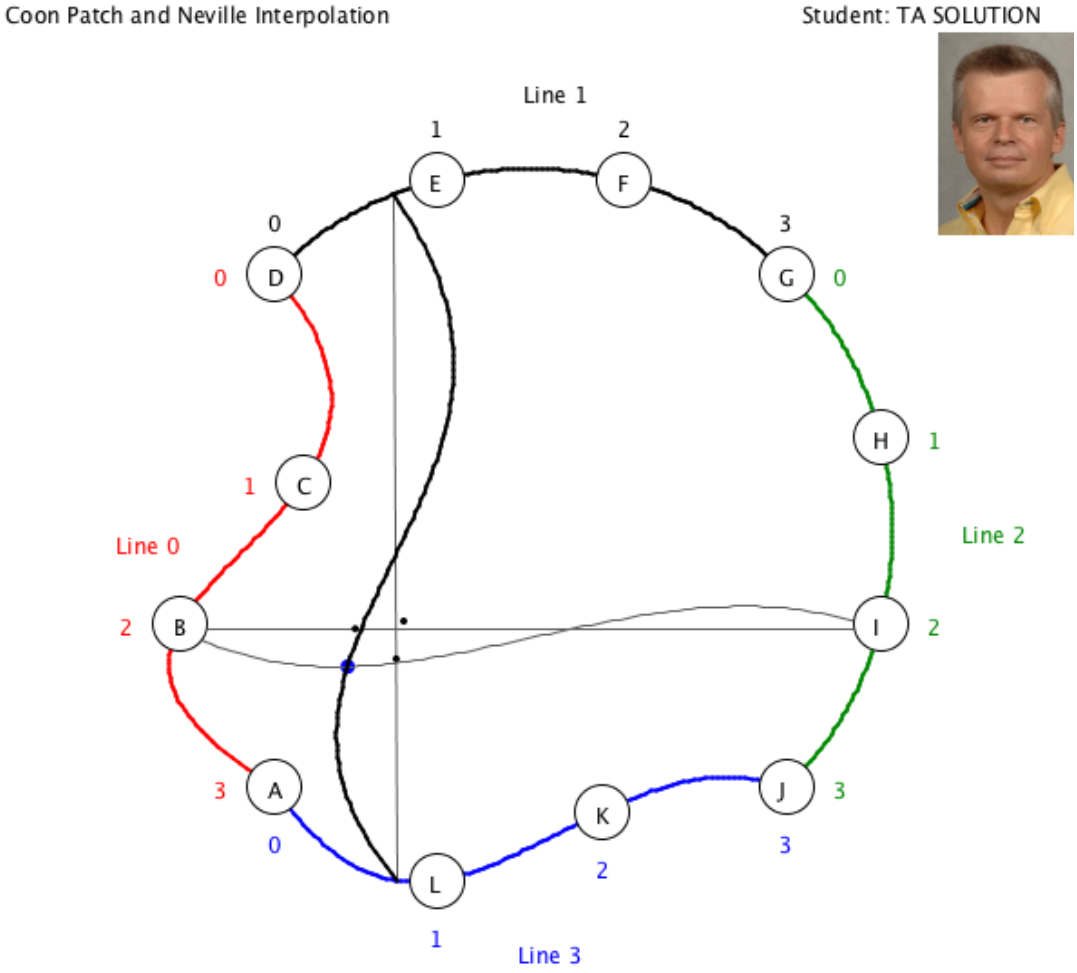
`http://en.wikipedia.org/wiki/Coons_patch`

Requirements

- You are give four control curves. You must interpolate between the left and right curves, using the top and bottom curves as guides. To do this, you should use the algorithm known as Coons Patch.
- Coons Patch interpolates a point (s, t) on a surface by generating three points, and computing a weighted average of them.
    1. It generates one point by first interpolating *down* the vertical curves by t, then interpolating across the line that joins them (the grey horizontal line in the picture) by s.
    2. It generates one point by first interpolating *across* the horizontal control curves by s, then interpolating down the line that joins them (the grey vertical line) by t.
    3. It generates one point by the bilinear interpolation of the four outer control points (D, G, A, and J in the picture), shown near the intersection of the two lines.
    4. The final control point (shown in blue) is calculated as the first two points minus the bilinearly interpolated point.
- You are given the control points for the curves, but they are currently drawn as blocky polylines. Instead, draw the horizontal curves as interpolating cubic curves, using Neville's algorithm (which is like Bezier cubic curves, but actually interpolates the control points), and the vertical lines as non-interpolating quartic bezier curves (which means you need to add another point to the left and right curves, to have 5 points each on the left and right!).

  In addition to just drawing the control curves, you will use these interpolation functions to compute the values along the control curves in order to do the rest of the project.
- You should draw the interpolated vertical curve at s for the curve interpolation as a function of mouseX, so that you can move the curve with the mouse.

  In other words, draw the curve created from something akin to a constant $s = \frac{\text{mouseX}}{\text{width}}$, and a variable t ranging from 0...1 (with a reasonable number of t values such that the curve doesn't look blocky).
- You should draw the interpolated *horizontal* curve at t in the same way you drew the vertical curve (shown in gray in the image).
- Also calculate a t from the mouseY, to use for drawing the four calculated points and their guides at (s, t) (as shown in the picture below). You should probably implement this *first* (before the actual curve) to help you debug.



Important note! The above picture shows the left and right curves using 4 points interpolated with Neville's algorithm. This is *different* from what you need to do! You need to use 5 points to create *quartic bezier* curves for the left and right edges.

Deliverables

- Code
- Video (showing your code working, in a variety of interesting control curve configurations.)
- Writeup (describing problem statement and your solution)