CSE 5443 Lab 4
Autumn, 2021

Implement a program that reads two files, each containing the knot vector and control points of a B-spline curve, generates a B-spline surface, and then output the B-spline surface in Geomview OFF (ASCII) format.

Program details:

1. Name the program `bsplinesurf`.
2. Run the program as:

   ```
   bsplinesurf <infile1> <infile2> <outfile>
   ```

   (a) `<infile1>` is the name of a file containing the definition of an open B-spline curve ζ_1, including control points $(q'_0, q'_1, \ldots, q'_m)$ and knot vector $(t'_0, t'_1, \ldots, t'_{m+k})$;

   (b) `<infile2>` is the name of a second file containing the definitiion of an open B-spline curve ζ_2, including control points $(q''_0, q''_1, \ldots, q''_n)$ and knot vector $(t''_0, t''_1, \ldots, t''_{n+h})$;

   (c) Use the format for the B-splines in files `<infile1>` and `<infile2>` in the instructions for Lab 2. Create the files using your lab 2.
   (Note: There should be NO commas in your input files.)

   (d) `<outfile>` is the name of the output file. The output file should be in Geomview OFF (ASCII) format and should have the suffix ".off".

3. The control point in `<infile1>` should be viewed as lying in the y-z plane where the first coordinate of each control point is the y coordinate and the second coordinate is the z-coordinate. The x-coordinate of each control point is 0.
4. The control points in `<infile2>` should be viewed as lying in the x-y plane. The z-coordinate of each control point is 0.
5. For $i = 0, \ldots, m$ and $j = 0, \ldots, n$, let q_{ij} equal $(q'_i + q''_j)$. The $q_{i,j}$ are the control points of the surface.
6. Generate the B-Spline surface as follows:

   (a) Sample $[t'_{k-1}, t'_{m+1}]$ at a "sufficiently" high rate to form a sequence $(u_0, u_1, \ldots, u_{n_u})$ of $n_u + 1$ points, where $u_0 = t'_{k-1}$ and $u_{n_u} = t'_{m+1}$;

   (b) Sample $[t''_{h-1}, t'_{n+1}]$ at a "sufficiently" high rate to form a sequence $(v_0, v_1, \ldots, v_{n_v})$ of $n_v + 1$ points, where $v_0 = t''_{h-1}$ and $v_{n_v} = t''_{n+1}$;

   (c) For each $a \in [0, 1, \ldots, n_u]$ and $b \in [0, 1, \ldots, n_v]$, compute the point $p_{a,b} = p(u_a, v_b)$.

   (d) To compute $p(u_a, v_b)$, use multiple applications of de Bohr's algorithm as described in the BSplineSurfaces.pdf notes, p. 2:

      i. Apply de Bohr's algorithm $m + 1$ times to compute $\hat{q}_i = \sum_{j=0}^{n} N_{j,h}(v_b) q_{i,j}$ for $i = 0, 1, \ldots, m$.

      ii. Apply de Bohr's algorithm again to compute $p(u_a, v_b) = \sum_{i=0}^{m} N_{i,k}(u_a)\hat{q}_i$.

   (e) Note: Do not confuse the B-Spline control points $q_{i,j}$ with the points $p_{a,b}$ computed on the B-Spline surface $p(u, v)$. There are $(m + 1) \times (n + 1)$ control points q_{ij}. There are thousands (maybe tens of thousands) of surface points p_{ij}.

   (f) For each $u_a \in [0, 1, \ldots, n_u - 1]$ and $v_b \in [0, 1, \ldots, n_v - 1]$, construct a quad mesh element, (mesh face,) consisting of 4 vertices, $p_{ab}, p_{a+1,b}, p_{a+1,b+1}, q_{a,b+1}$. Note that the four vertices are listed in order around the quad. The order should be consistent for all quads, i.e. it should be counter-clockwise for all quads (or clockwise for all quads, depending on which side of the surface is being viewed.) The 4 vertices represent the polygonal curve bounding the quad mesh element. (The 4 vertices may not be co-planar, so the quad mesh element is not a true quadrilateral.)

(g) The program should output a list of the coordinates of the points p_{ab} followed by a list of vertices of each mesh faces (quad). The output should be in Geomview (ASCII) OFF format. See below for references and examples of the format.

(h) Use the program `meshlab` to read and view the output GEOMVIEW OFF file. `meshlab` can be downloaded from: `https:meshlab.net`

7. Include a README file that contains any details of running your program, INCLUDING the OPERATING system (Windows 10, Max, Linux,) on which you ran and tested your program.

8. Optional arguments/features:

   (a) You can add optional command line arguments of the form `-argA` or `-argA <valA>` or `-argA <valAI> <valAII>` or to your program.

   (b) Usage should be:

   ```
   bsplinesurf [OPTIONAL ARGUMENTS] <infile1> <infile2> <outfile>
   ```

   (c) Optional arguments could include arguments such as `-nump <nump1> <nump2>` where `nump1` and `nump2` are n_u and n_v, respectively;

   (d) You can define any other arguments at your discretion, but your program should run reasonably with the default command format:

   ```
   bsplinesurf <infile1> <infile2> <outfile>
   ```

   (e) Entering "`bsplinesurf`" with no arguments should output a usage message with all possible options.

   (f) If you do have optional arguments, entering "`bsplinesurf -h`" should output a help message with a a brief explanation of each option.

   (g) The third file name argument `<outfile>` can be made optional. If only two filenames are included in the command line, then the default output file name should be "`out.off`". Even if `<outfile>` is made optional, specifying the third file name should set the output file to the specified file name.

   (h) Optional command line arguments are totally optional. There is no extra credit for adding such arguments.

Geomview OFF format

While Geomview OFF format is very basic, it is an exceptionally simple format for representing polygonal meshes. A wiki page gives a simple explanation and example:
`https://en.wikipedia.org/wiki/OFF_(file_format)`

Other explanations/examples are at:
`http://www.geomview.org/docs/html/OFF.html`
`https://people.sc.fsu.edu/~jburkardt/data/off/off.html`

Even though Geomview OFF format is very basic, we still don't need many/most of the features in the format, such as normals, colors, texture coordinates or 4 or higher dimenional coordinates.

(No extra credit for this lab.)