|  |  |  |
| --- | --- | --- |
| COMP 4560 | **Computer Graphics for Computer Systems Technology** | bcit_logo_color |

**Laboratory Assignment 7: Experiments with Curves**

In this assignment, we use Excel to create a Bezier curves. You are asked to use the set of fourteen control points shown in the figure to the right. The resulting control polygon has regions of considerable twisting, and as well sequences of three and four control points in straight lines. The file or asgn7student.xlsm for available on shareout: contains the coordinates of the control points and the chart shown to the right.

To complete the graded part of this assignment, you must superimpose on the chart present in the starting worksheet a Bezier curve.

**You may work in groups of up to 3 on this assignment. Show your completed Bezier curve (pages 1-3) to your lab instructor by the end of today’s lab in order to get credit.**

**The Strategy**

As discussed in class, coordinates of points along a Bezier or spline curve in two dimensions can be generated using a formula of the following type:

 (1)

where

(i) the rows of the matrix **X** to the left of the main equals sign form coordinates of points along the curve. Joining these points by straight line segments produces an approximation to the actual curve. The larger the value of n used, the smoother the final curve will appear to be.

(ii) the (L + 1) x 2 matrix, **p**, on the extreme right of the formula is the matrix of (x, y) coordinates of the L + 1 control points producing the current segment of the overall curve.

(iii) The matrix **B** is the (L + 1) x (L + 1) “shape matrix,” which embodies the geometry of the particular curve generation method being used.

(iv) the matrix **t** is made up of n + 1 rows, corresponding to n + 1 values of the parameter t, ranging from 0 to 1 in steps of 1/n. It contains L+ 1 columns. Starting from the left, the columns contain values of t0, t1, t2, … up to tL.

Constructing any kind of Bezier or spline curve then just consists of constructing the matrix on the left above as one or a succession of operations of the type shown above, and then plotting the sequence of line segments. The calculations are easily implemented in an Excel spreadsheet, and the result is easily graphed using an Excel xy-scattergraph-type chart.

**The Bezier Curve**

The Bezier curve for a set of control points is constructed as a single segment, taking into account all (L + 1) control points at once. Proceed as follows:

(i) the matrix, **p,** of control points already exists in the template file. It is the fourteen row by two column range of (x, y) coordinates of the control points.

(ii) the matrix **t** consists of 14 columns (corresponding to the powers of t from 0 through 13) and N rows, where N is the number of points along the Bezier curve to be computed. Enter a list of t values starting from 0, going to 1 in steps of 0.01. (This will give 101 points along the Bezier curve, which should be adequate here. The easiest way to generate such a list in Excel is to enter the values 0 and 0.01 in two consecutive cells in a column. Select those two cells, and pull down on the fill handle of the lower cell to generate the rest of the sequence.) To the left of this column, enter 1 in every cell (corresponding to t0). To the right of this column of t values, create columns containing t2, t3, …, t13 for each row. You should end up with a table with 101 rows and 14 columns looking something like:



Remember, this should involve very little typing on your part, since most of the entries in the table are easily generated by copy and paste, or using the fill handle.

(iii) Construction of the **B**  matrix will be done recursively (described below) rather than than with the explicit formula described in class. Again, most of this 14 x 14 matrix is generated by formula. You could type in every value by hand, but that would be a waste of your valuable time, you would learn nothing in the process, and you would likely mistype at least one of the 196 matrix elements, which could mess up the entire curve (since the Bezier curve is produced as a unit all at once).

Start by identifying a range of cells 14 columns wide and 14 rows high. Use a row and column to number off these rows and columns from 0 to 13 – the row and column indices you create will be useful shortly. Enter the value 1 in each element of the last row of this range and zeros in each remaining element of the last column of this range. This will leave you with a table looking like this:



The values in each of the remaining blank cells of the table are computed by summing the value immediately to the right, and the value immediately to the right and down one row. This can be accomplished by entering a single formula and copying it to the remaining blank cells.

Now, enter formulas which result in the values of the first column of the table being listed in order along the row just below the table with the same indices (shown in the middle of the figure on the next page). This can be done by copying the first column and using a combination of Paste -> Paste Values and Paste -> Transpose. Then create a new table from the existing one (just below it) in which the elements are the product of the corresponding element in the first table and the value in the same column of this row of numbers just produced, and in which the elements alternate in sign. **The sign alternation MUST be built into your formula!** The final result should look something like:



(iv) Now, to create **X** and plot the Bezier curve. First, form the matrix product **B•p** (the 14 x 2 product of the 14 x 14 matrix **B** and the 14 x 2 matrix **p**) in a range of 14 rows and 2 columns just below the **B** matrix completed as above.

Then, to use as a guide, copy and paste the column of t-values generated in step (ii) above into column A, somewhere below where the **p** matrix is stored. Label the columns adjacent to this x and y. Now, multiply the 101 x 14 **t** matrix onto the 14 x 2 matrix **B•p**, and put the results into the 101 x 2 range of cells below the labels x and y. These will be the (x, y)-coordinates of 101 successive points along the Bezier curve.

Finally, add this 101 x 2 range of (x, y) coordinates as a new data series in the chart. The Bezier curve should appear in the chart. Right click on the curve, choose “Format Data Series,” and fine tune the appearance of the curve to involve **no markers** and an approximately 2 pt wide line of a distinctive color of your choice. This will give you your completed Bezier curve.

Before leaving this exercise, take a minute to relate what you’ve done here to the algebraic method described in class. Also, take a minute to look at the shape of the Bezier curve. You can experiment with how the location of the control points affects its shape by simply changing numbers manually in the **p** matrix of by selecting that point on the chart and dragging it. Of course, make sure that you’ve saved your work so far before proceeding – in fact, you should be saving your work frequently as you step through these exercises.

**Show your exercise to your instructor to receive credit.**

**The Cubic B-Spline Curve**

For comparison, we’ll construct a cubic (m = 4) B-spline curve for these control points, using the standard open knot vector formulas. In this case, since there are 14 control points (L = 13), the cubic B-spline will be made up of 11 (= L – m + 2) segments. Points along each segment are generated by a formula very similar to equation (1) earlier, but specifically for the kth segment by:

 (2)

Here k = 0, 1, 2, 3, 4, 5, 6, 7 ,8, 9, 10 and 11 together generate the whole curve in this example. The matrix **t** has the same structure as for the Bezier curve, but with two differences in detail: since this is a cubic curve, we need only four columns: t0, t1, t2, and t3. Thus **t** is an n x 4 matrix of powers of t. Also, since the entire curve will consist of 11 segments, we need only about 10 or 20 points per segment here to get adequate smoothness. So, n can be something like 10. Since each segment makes use of only four of the control points, the matrix **p**k above is successive sets of four control points from the set of fourteen control points generating the entire curve. Finally, we can set up the matrix **X** of coordinates along each segment so that they are contiguous down the column (no blank rows separating coordinates of points for one segment from the coordinates of points for the next segment.

The figure to the right illustrates the suggested setup. Notice that for the first ten segments, we just need points for t = 0 through t = 0.9, say, since the t = 0 for the next segment is at the same location as the t = 1 point for the previous segment. (You can put t = 1 into each segment to confirm for yourself that this last statement is correct – it won’t affect the shape of the final graph.) The only thing you must avoid is blank rows, which will result in gaps in your plotted curve. Once coordinates of points along each segment are filled in to this long list, the entire list can be added to the chart as a new series, producing a picture of the cubic B-spline curve for this set of control points.

The matrices **p**0, **p**1, …, **p**10 are just successive groups of four rows in the matrix **p** of control points already present in the workbook. You only need to set up one copy of the **t** matrix, with rows corresponding to t = 0 to t = 1 in steps of 0.1. For the first nine segments, use just the first 10 rows of this **t** matrix. For the last segment, use all 11 rows, so that the last segment of the cubic B-spline reaches the last control point.

You will have to enter the five different 4 x 4 **N** matrices required here manually into your worksheet. The values of the elements of each of them are:







 

Every element of these entire matrices are to be multiplied by the fractions on the right in each case.

**Questions**

At this point you should have the original control polyline plus the three curves you’ve calculated all superimposed on the same chart, each of the last three as a line only with no markers, and all four in distinct colors. Now, create a textbox to the immediate right of the chart, and answer the following questions briefly but specifically:

Compare and contrast the Bezier and cubic B-spline.