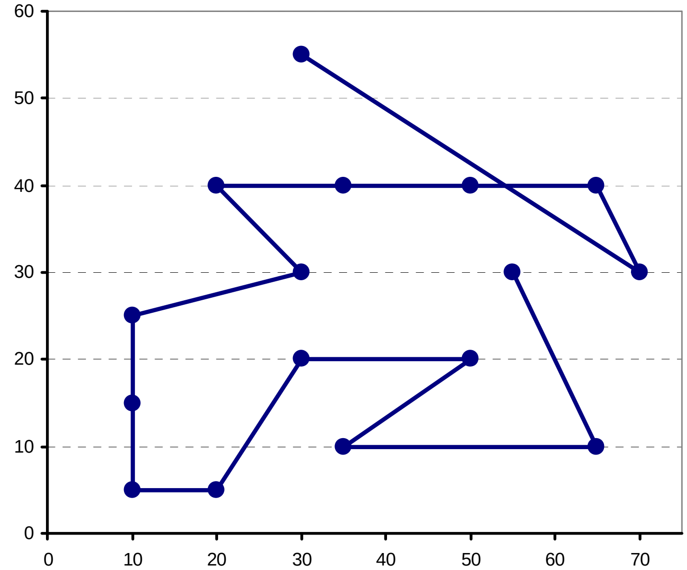
Laboratory Assignment 7: Experiments with Curves

In this assignment, we use Excel to compare the shapes of a Bezier, cubic B-spline, and Kochanek-Bartels spline curves. You are asked to use the set of sixteen 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 Templateasgn7.xlsm for Excel 2007 available on shareout: contains the coordinates of the control points and the chart shown to the right.



To complete this assignment, you must superimpose on the chart present in the starting worksheet (in different colors) the curves of each of these three types generated using these control points. Submit your final Excel file to sharein using a filename of the form *name.set.*asgn7.xls. Submit a single sheet printout containing only the final chart and an adjacent textbox answering the brief questions at the end of this handout to the instructor (put your name on it, as well, of course!) No other printout is required.

The absolute deadline for this assignment is 4:30 p.m. on Wednesday, April 15, 2009.
YOU MUST BEGIN THIS ASSIGNMENT WITH A COPY OF THE STARTING EXCEL WORKBOOK THAT YOU DOWNLOAD FROM THE SHAREOUT: DIRECTORY YOURSELF. YOU MAY NOT BORROW A COPY OF THE FILE FROM ANY OTHER STUDENT, EVEN IF THEY HAVE NOT DONE ANY WORK ON IT. VIOLATION OF THIS RULE WILL RESULT IN A GRADE OF ZERO FOR BOTH THE DONOR AND THE RECIPIENT STUDENT!

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:

$$\mathbf{X} = \begin{bmatrix} x_0 & y_0 \\ x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix} = \mathbf{t \bullet B \bullet} \begin{bmatrix} (p_0)_x & (p_0)_y \\ (p_1)_x & (p_1)_y \\ \vdots & \vdots \\ (p_L)_x & (p_L)_y \end{bmatrix} = \mathbf{t \bullet B \bullet p} \qquad (1)$$

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 t^0, t^1, t^2, … up to t^L.

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.

<u>The Bezier Curve</u>

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 sixteen row by two column range of (x, y) coordinates of the control points.

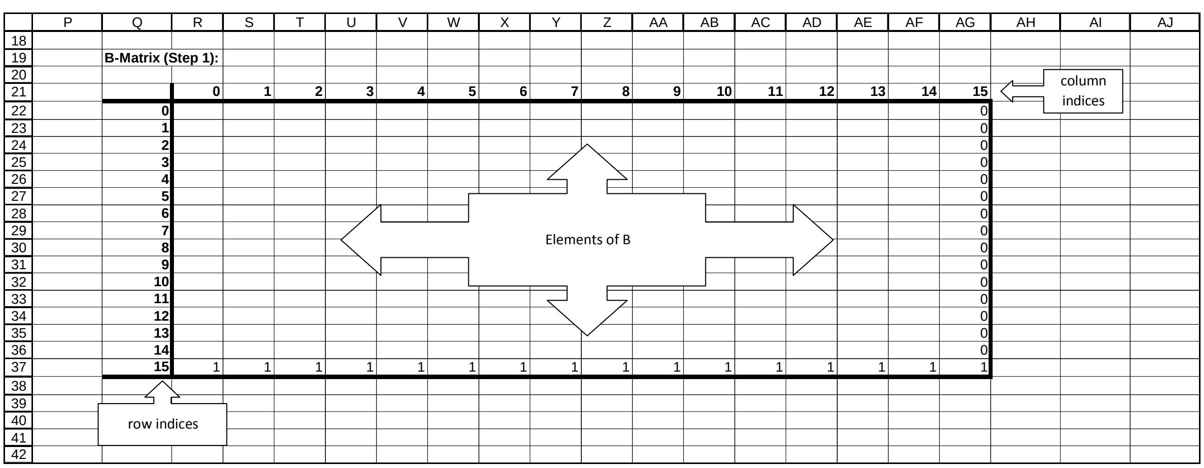
(ii) the matrix **t** consists of 14 columns (corresponding to the powers of t from 0 through 13) and N rows, where N is the number of points along the Bezier curve to be computed. Enter a list of t values starting from 0, going to 1 in steps of 0.005. (This will give 201 points along the Bezier curve, which should be adequate here. The easiest way to generate such a list in Excel is to enter the values 0 and 0.005 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 t^0). To the right of this column of t values, create columns containing t^2, t^3, …, t^{15} for each row. You should end up with a table with 201 rows and 16 columns looking something like:

t^0	t^1	t^2	t^3	....	t^{14}	t^{15}
1	0	0	0	....	0	0
1	0.005	0.000025	1.25E-07	....	6.1E-33	3.05E-35
1	0.010	0.0001	0.000001	....	1E-28	1E-30
1	0.015	0.000225	3.38E-06	....	2.92E-26	4.38E-28
1	0.020	0.0004	0.000008	....	1.64E-24	3.28E-26
....	....	....	....	....	....	....
1	0.990	0.9801	0.970299	....	0.868746	0.860058
1	0.995	0.990025	0.985075	....	0.93223	0.927569
1	1	1	1	....	1	1

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 requires the three steps described in class. Again, most of this 16 x 16 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 256 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 16 columns wide and 16 rows high. Use a row and column to number off these rows and columns from 0 to 15 – 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. This will have to be done manually, one cell at a time. 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. **<u>The sign alternation MUST be built into your formula</u>!** The final result should look something like:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	15	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	105	14	1	0	0	0	0	0	0	0	0	0	0	0	0	0
3	455	91	13	1	0	0	0	0	0	0	0	0	0	0	0	0
4	1365	364	78	12	1	0	0	0	0	0	0	0	0	0	0	0
5	3003	1001	286	66	11	1	0	0	0	0	0	0	0	0	0	0
6	5005	2002	715	220	55	10	1	0	0	0	0	0	0	0	0	0
7	6435	3003	1287	495	165	45	9	1	0	0	0	0	0	0	0	0
8	6435	3432	1716	792	330	120	36	8	1	0	0	0	0	0	0	0
9	5005	3003	1716	924	462	210	84	28	7	1	0	0	0	0	0	0
10	3003	2002	1287	792	462	252	126	56	21	6	1	0	0	0	0	0
11	1365	1001	715	495	330	210	126	70	35	15	5	1	0	0	0	0
12	455	364	286	220	165	120	84	56	35	20	10	4	1	0	0	0
13	105	91	78	66	55	45	36	28	21	15	10	6	3	1	0	0
14	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	1	15	105	455	1365	3003	5005	6435	6435	5005	3003	1365	455	105	15	1

0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	-15	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	105	-210	105	0	0	0	0	0	0	0	0	0	0	0	0	0
3	-455	1365	-1365	455	0	0	0	0	0	0	0	0	0	0	0	0
4	1365	-5460	8190	-5460	1365	0	0	0	0	0	0	0	0	0	0	0
5	-3003	15015	-30030	30030	-15015	3003	0	0	0	0	0	0	0	0	0	0
6	5005	-30030	75075	-100100	75075	-30030	5005	0	0	0	0	0	0	0	0	0
7	-6435	45045	-135135	225225	-225225	135135	-45045	6435	0	0	0	0	0	0	0	0
8	6435	-51480	180180	-360360	450450	-360360	180180	-51480	6435	0	0	0	0	0	0	0
9	-5005	45045	-180180	420420	-630630	630630	-420420	180180	-45045	5005	0	0	0	0	0	0
10	3003	-30030	135135	-360360	630630	-756756	630630	-360360	135135	-30030	3003	0	0	0	0	0
11	-1365	15015	-75075	225225	-450450	630630	-630630	450450	-225225	75075	-15015	1365	0	0	0	0
12	455	-5460	30030	-100100	225225	-360360	420420	-360360	225225	-100100	30030	-5460	455	0	0	0
13	-105	1365	-8190	30030	-75075	135135	-180180	180180	-135135	75075	-30030	8190	-1365	105	0	0
14	15	-210	1365	-5460	15015	-30030	45045	-51480	45045	-30030	15015	-5460	1365	-210	15	0
15	-1	15	-105	455	-1365	3003	-5005	6435	-6435	5005	-3003	1365	-455	105	-15	1

(iv) Now, to create **X** and plot the Bezier curve. First, form the matrix product **B•p** (the 16 x 2 product of the 16 x 16 matrix **B** and the 16 x 2 matrix **p**) in a range of 16 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 201 x 16 **t** matrix onto the 16 x 2 matrix **B•p**, and put the results into the 201 x 2 range of cells below the labels x and y. These will be the (x, y)-coordinates of 201 successive points along the Bezier curve. (The matrix multiplication **t•B•p** could be done as a single matrix multiplication formula in which mmult invokes a second call to mmult as one of the function parameters.)

Finally, add this 201 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 **<u>no markers</u>** 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 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.

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 16 control points (L = 15), the cubic B-spline will be made up of 13 (= 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:

$$\mathbf{X} = \begin{bmatrix} x_0 & y_0 \\ x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix} = \mathbf{t}\bullet\mathbf{N}\bullet\begin{bmatrix} (p_k)_x & (p_k)_y \\ (p_{k+1})_x & (p_{k+1})_y \\ (p_{k+2})_x & (p_{k+2})_y \\ (p_{k+3})_x & (p_{k+3})_y \end{bmatrix} = \mathbf{t}\bullet\mathbf{N}\bullet\mathbf{p}_k \qquad (2)$$

Here k = 0, 1, 2, 3, 4, 5, 6, 7 ,8, 9, 10, 11, 12, and 13 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: t^0, t^1, t^2, and t^3. Thus **t** is an n x 4 matrix of powers of t. Also, since the entire curve will consist of 13 segments, we need only about 10 or 20 points per segment here to get adequate smoothness. (In the illustration below, n = 10 is used.) Since each segment makes use of only four of the control points, the matrix $\mathbf{p}_k$ above is successive sets of four control points from the set of sixteen control points generating the entire curve (that is, $\mathbf{p}_k$ is the 4 x 2 matrix consisting of rows k, k+1, k +2, and k + 3 of the table of control points). 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 twelve 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.

Calculations for the cubic B-spline			
16 points --> 13 segments			
	t	x	y
Segment 0	0		
	0.1		
	0.2		
	0.3		
	0.4	**$t\bullet N_0 \bullet p_0$**	
	0.5		
	0.6		
	0.7		
	0.8		
	0.9		
Segment 1	0		
	0.1		
	0.2		
	0.3		
	0.4	**$t\bullet N_1 \bullet p_1$**	
	0.5		
	0.6		
	0.7		
	0.8		
	0.9		
Segment 2	0		
	0.1		
	0.2		
		.....	.....
	0.8		
	0.9		
Segment 12	0		
	0.1		
	0.2		
	0.3		
	0.4	**$t\bullet N_{12} \bullet p_{12}$**	
	0.5		
	0.6		
	0.7		
	0.8		
	0.9		
	1		

The matrices $\mathbf{p}_0$, $\mathbf{p}_1$, …, $\mathbf{p}_{12}$ 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 twelve 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:

$$\mathbf{N}_0 = \begin{bmatrix} 12 & 0 & 0 & 0 \\ -36 & 36 & 0 & 0 \\ 36 & -54 & 18 & 0 \\ -12 & 21 & -11 & 2 \end{bmatrix} \bullet \frac{1}{12}$$

(First Segment, k = 0)

$$\mathbf{N}_1 = \begin{bmatrix} 3 & 7 & 2 & 0 \\ -9 & 3 & 6 & 0 \\ 9 & -15 & 6 & 0 \\ -3 & 7 & -6 & 2 \end{bmatrix} \bullet \frac{1}{12}$$

(Second Segment, k = 1)

$$\mathbf{N}_G = \begin{bmatrix} 1 & 4 & 1 & 0 \\ -3 & 0 & 3 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \bullet \frac{1}{6}$$

(Generic Segment, k = 2, ..., L - m - 1)

$$\mathbf{N}_{L-4} = \begin{bmatrix} 2 & 8 & 2 & 0 \\ -6 & 0 & 6 & 0 \\ 6 & -12 & 6 & 0 \\ -2 & 6 & -7 & 3 \end{bmatrix} \bullet \frac{1}{12}$$

(Second last Segment, k = L-4)

$$\mathbf{N}_{L-3} = \begin{bmatrix} 2 & 7 & 3 & 0 \\ -6 & -3 & 9 & 0 \\ 6 & -15 & 9 & 0 \\ -2 & 11 & -21 & 12 \end{bmatrix} \bullet \frac{1}{12}$$

(Last Segment, k = L - 3)

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

The Kochanek-Bartels Spline Formula

The Kochanek-Bartels spline formula (hereafter KB formula) is based on a cubic approximation, and so the formula is essentially the same as equation (2) on page 3 of this handout for the cubic B-spline, except that the elements of the 4 x 4 **N** matrix will be calculated using different formulas. The coordinates of points forming segment #k of a KB spline are given by the formula:

$$\mathbf{X} = \begin{bmatrix} x_0 & y_0 \\ x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix} = \mathbf{t} \bullet \mathbf{M}_{KB} \bullet \begin{bmatrix} (p_k)_x & (p_k)_y \\ (p_{k+1})_x & (p_{k+1})_y \\ (p_{k+2})_x & (p_{k+2})_y \\ (p_{k+3})_x & (p_{k+3})_y \end{bmatrix} = \mathbf{t} \bullet \mathbf{M}_{KB} \bullet \mathbf{p}_k \qquad (3)$$

where

$$\mathbf{M}_{KB} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -s(1+b)(1-c) & 2s(b-c) & s(1-b)(1+c) & 0 \\ 2s(1+b)(1-c) & -3+s(1-3b+5c+bc) & 3+s(-2-4c+2bc) & -s(1-b)(1-c) \\ -s(1+b)(1-c) & 2+s(-1+b-3c-bc) & -2+s(1+b+3c-bc) & s(1-b)(1-c) \end{bmatrix} \qquad (4)$$

where s, b, and c are the parameters defined in the development of the KB spline formulas. Set up individual cells in your worksheet to hold the values of s, b, and c, and you will probably find it much easier to define names for these cells that you can use directly in the formulas for the elements $\mathbf{M}_{KB}$[1]. You only need one copy of the $\mathbf{M}_{KB}$ matrix – its elements have the same value for every segment of the curve.

The KB spline formula uses the first and last control point solely to establish a directional vector for the beginning and ending control point of the curve. Hence the curve will actually be seen to begin with p_1 and end with p_{14} (this will happen automatically), which still results in an thirteen segment curve in this case. To generate each of the eleven segments, simply set up equation (3) with the matrices $\mathbf{p_k}$ being successive groups of four control points (first segment uses p_0, p_1, p_2 and p_3; the second segment uses p_1, p_2, p_3, and p_4; and so on, with the last segment, the 13th segment using p_{12}, p_{13}, p_{14}, and p_{15}). You might want to have values of t increasing in increments of 0.05 here, since the KB spline is capable of much tighter twists than are typical with Bezier or B-spline curves.

Superimpose the KB spline curve for these control points on the master chart, making sure that the curve is a line only, with no markers. Setting s = 0.5, with b and c both zero gives you the so-called Catmull-Rom spline. Adjusting b and c to nonzero values should produce the sort of changes in shape that were discussed in class and mentioned in the attached notes.

---

[1] Although it should go without saying, the obvious "names" to use for these cells are s, b, and c, respectively. Then the Excel formulas you enter will look exactly like the formulas shown in the matrix (4) above. Consult Excel help for details on how names are associated with cells or ranges of cells in the version of Excel you are using.

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. These answers are worth a significant number of marks. You will get marks only for evidence that you have tried to formulate meaningful, informative, focused sentences!

1) In two or three sentences, compare and contrast the Bezier, cubic B-spline, and Catmull-Rom spline (this is the KB curve with s = 0.5, b = c = 0). Give actual substantive answers here – avoid wishy-washy.
2) In one sentence, compare the shape of the Catmull-Rom spline when s = 0.25 to the shape when s = 0.5. Make a similar comparison between the shapes at s = 1 and s = 0.5. In one sentence, what happens when s = 0?
3) Set s = 0.5 in the KB spline. Compare the shapes of the curve when b = +2, b = 0 and b = -2. (Keep c = 0 throughout.)
4) Set s = 0.5 and b = 0 in the KB spline. Now, compare the shapes of the curve when c = +2, 0, and -2.
5) Set s = 0.5 in the KB spline. Try a couple of combinations of b and c where the values are quite different from zero (say, b = 2, c = 5, or b = 2, c = -5). What happens to the shape of the curve?

Finish!

Print out your chart (with your name, etc.) and the textbox giving your answers to the above questions (on a single sheet is fine). Hand this sheet in to the course instructor. (**<u>Do not</u> print out the ranges of cells in which you did your calculations – this will be many many pages of numbers that mean little to a reader.)**
Upload your Excel file to your COMP 4560 set directory for assignment 7 on the sharein drive.