CSC 260, Spring 1999 - Assignment #2

Due in class on March 22. Worth 10% of the course grade.

Note that this assignment is to be done by each student individually. You may discuss it in general terms with other students, but the work you hand in should be your own.

In this assignment, you will write a set of Maple procedures that let you draw closed curves that are defined using Catmull-Rom cubic splines. There are four components to this task: (1) writing a procedure that will find the piecewise cubic function satisfying the conditions required for a "circular" Catmull-Rom spline, (2) writing a procedure that uses this procedure to find the basis functions for this spline, along with a procedure that uses these basis functions to compute the splines more quickly than the first procedure would, (3) using these splines to define a closed parametric curve of a certain shape, (4) discussing whether this procedure can be improved.

As discussed in the lecture, a Catmull-Rom cubic spline based on data $(t_0, x_0), (t_1, x_1), \ldots, (t_n, x_n)$, where $t_0 < t_1 < \cdots < t_n$, is a piecewise cubic polynomial function in which the boundaries of the n pieces are at the t_i locations. The values of each piece at its two endpoints match the given x values. Furthermore, the first derivative of the two pieces that end at location t_i match the slope of the line segment joining (t_{i-1}, x_{i-1}) to (t_{i+1}, x_{i+1}) (the two end-points need to be treated specially).

In this assignment, you will implement a variation on this scheme that we will call a "circular" Catmull-Rom spline, using t values that are uniformly spaced around a "circle" of values from 0 to 1. If there are n datapoints, such a spline will have n pieces. The first piece handles t values from $0 = 0/n$ to $1/n$, the second piece handles t values from $1/n$ to $2/n$, and so forth, up to the nth piece, which handles t values from $(n-1)/n$ to $n/n = 1$. These pieces are cubic polynomials whose values at their endpoints match the n data points given. The first data point given corresponds to a t value of $1/n$, the second to a t value of $2/n$, etc. up to the nth value, which corresponds to *both* a t value of $n/n = 1$ *and* a t value of $0/n = 0$. The first derivatives at the ends of each piece must match the slope of the line segment connecting the two adjacent data points. For instance, the first derivative of the second piece at $t = 2/n$ must match the slope of the line segment from $(1/n, x_1)$ to $(3/n, x_3)$. The first derivatives at the start of the first piece and the end of the last piece must match the slope of the line segment obtained by imagining that things wrap around. Specifically, the first derivative of the first piece at $t = 0$ must match the slope of the line segment from $(-1/n, x_n)$ to $(1/n, x_1)$, and the first derivative of the last piece at $t = 1$ must match the slope of the line segment from $((n-1)/n, x_{n-1})$ to $((n+1)/n, x_1)$.

The reason for defining these circular splines is so that they can be used to define smooth closed curves. For this purpose, two such splines are needed, one giving the x values for a parametric representation of the curve, the other giving the y values. Both splines will be used with t ranging from 0 to 1. The circular nature of the splines ensures that the curve they define is closed, and has continuous first derivatives in all positions.

In an interactive graphics application, the splines defining such a curve would need to be found rapidly, so that the curve could quickly be re-displayed when the user changed the location of one of the datapoints used to define it. To allow this, you will find the basis functions for these splines (for a given number of datapoints), and use them to rapidly compute the splines for any set of x values at these locations.

Question 1: [50 marks] Write and hand in a Maple procedure called `circspline` that finds a circular piecewise cubic spline as described above. The list of x values should be given as first argument of `circspline`. The second argument will be the unknown variable for the cubic polynomials. The result returned should be expressed using the `piecewise` function.

To help you get started, a Maple procedure that finds first-order splines is given in the accompanying handout. You should follow the general pattern of this procedure — in particular, for this exercise, you should *not* work out solutions to equations by hand, even thought that isn't all that hard to do, but should instead get Maple to solve them, as in the example provided. However, you will need to change the way things are organized a bit to adapt this procedure to "circular" splines, and to impose the Catmull-Rom conditions. You can get to this procedure on CDF at `/u/radford/linspline.mws` (eg, you can "Open" it with that file name). You may use this as a starting point for your own procedure if you wish.

Here is an example of the output of this procedure:

```
> circspline([1,4,7,6],t);
```

$$\begin{cases} 6-12\,t-128\,t^2+384\,t^3 & t \le \frac{1}{4} \\ 14-116\,t+320\,t^2-256\,t^3 & t \le \frac{1}{2} \\ 22-116\,t+224\,t^2-128\,t^3 & t \le \frac{3}{4} \\ -14+52\,t-32\,t^2 & t \le 1 \end{cases}$$

Question 2: [30 marks] Write and hand in a Maple procedure called `circbasis` that finds the basis functions for the circular splines with a given number (n) of datapoints (you can use your `circspline` procedure in doing this). Recall that the n basis functions are simply the results of fitting the splines to x values that are all zero except for a single 1; the n possible positions for this 1 value give the n basis functions. You should store the sequence of these n basis functions in a global variable called `basis`. The arguments of `circbasis` will be the number of datapoints (n) and the unknown variable to use in the cubic polynomials. It doesn't matter what `circbasis` returns; returning NULL will make the result be invisible.

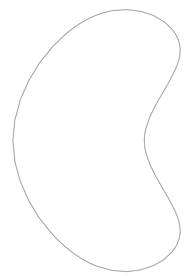
You should also write and hand in a procedure called `csb` that uses these pre-computed basis functions to quickly find the circular spline when given a list of x values for these n locations. You may need to use the `simplify` procedure in order to get Maple to combine a sum of piecewise functions into one piecewise function.

For example, the following commands should produce essentially the same result as above:

```
> circbasis(4,t);
> csb([1,4,7,6]);
```

Note that if you change the number of datapoints you are using, you will need to call `circbasis` again.

Question 3: [10 marks] Use the procedures above to design a closed curve that looks something like this:



You will probably need to use something like six or eight points to define a curve like this. For each point, you will need to specify the x and y positions. You will use commands something like this:

```
> circbasis(6,t)
> x := csb([ ... ]);
> y := csb([ ... ]);
> plot([x,y,t=0..1],scaling=constrained);
```

where the ... represents the x and y positions you select to try to get the desired curve. Hand in the points you decided to use in the end, along with the plot of the resulting curve.

Question 4: [10 marks] Finally, try to think of a way to improve this method, so that it takes less time to compute and/or to use the basis functions. You might try plotting the basis functions to see what they look like. Discuss briefly how you would go about implementing this improvement. You don't have to actually write procedures for an improved method, however.

All your procedures should be nicely formatted, with proper indentation. You should include comments where they will help someone understand your program, but you don't have to repeat information that is in this assignment sheet.