## Mathematics 308 — Fall 1997

## Fourth homework—due Wednesday, October 29

**1.** In the figure in the notes on how computers draw Bezier curves, the point $P_1^\bullet$ is $(1/2)P_0 + (1/2)P_{1/3}$. Find similar expressions for all the constructed points in terms of the original four.

**2.** The purpose of this exercise is to prove the assertion in the notes about the way in which a computer draws Bezier curves. Fix $t_0$ and $t_1$. Define

$$B(t) = (t_1 - t)^3 P_0 + 3(t - t_0)(t_1 - t)^2 P_1 + 3(t - t_0)^2 (t_1 - t) P_2 + (t - t_0)^3 P_3 / (t_1 - t_0)^3$$

The point is to verify that each half of the curve from $t_0$ to $t_1$ is also a Bezier curve, with known control points. We'll do one half here. Let $t_{1/2} = (1/2)(t_1 - t_0)$, the parameter value half way from $t_0$ to $t_1$, $\Delta t = t_{1/2} - t_0$. Let $P_i^\bullet$ be as in the notes. What must be shown is that

$$\begin{aligned} P(t_0) &= P_0^\bullet \qquad \text{(trivial)} \\ P'(t_0) &= 3\Delta t(P_1^\bullet - P_0^\bullet) \qquad \text{( almost trivial)} \\ P(t_{1/2}) &= P_3^\bullet \\ P'(t_{1/2}) &= 3\Delta t(P_3^\bullet - P_2^\bullet) \end{aligned}$$

A result in the notes then implies that the curve $P(t)$ from $t_0$ to $t_{1/2}$ is the Bezier curve with control points $P_i^\bullet$.

**3.** Write a **`PostScript`** procedure **`polynomial`** that you can fit into **`mkpath`** (the version that accepts an array of parameters) that will graph a polynomial between $x_0$ and $x_1$ with $N$ Bezier segments. You will use it like this:

```
[2 3 1] /polynomial -2 2 4 mkpath
```

to draw the graph of $2x^2 + 3x + 1$ between $x = -2$ and $x = 2$, using $4$ Bezier segments. As a beginning, you should start by drawing, say, quartic polynomials, then move on to the more difficult problem of variable degree.

There are a couple of of things you might think about: (1) For evaluating a polynomial in a program it is easiest to use an expression like $5x^3 + 2x + 3x + 4 = ((5x + 2)x + 3)x + 4$. This is called Horner's method for evaluation of polynomials. (2) You will have to add an argument to this procedure to pass the polynomial coefficients as an array. You will have to know that the PostScript command **`length`** returns the size of an array.

Include sample usage.

**4.** Write a **`PostScript`** program to draw the **Lissajous figure** with parametrization $t \mapsto (\cos 3t, \cos 4t)$, $t = 0$ to where it repeats. Generalize this to make a procedure to do this with $3$ and $4$ replaced by $m$ and $n$. Examples.