Drawing Bezier Curves

Bezier curve is an important modeling tool in curve/surface design. It is defined by a sequence of connected line segments called control polyline, as shown in Figure 1(a). To draw a Bezier curve, a subdivision algorithm can be used to recursively refine the control polyline to generate progressive linear approximations to the curve.

Let $P_0^i$ ($i = 0 \cdots n - 1$) be the original vertices of the control polyline. The following procedure generates one subdivision:

1. Define $P_1^i$ ($i = 1 \cdots n - 1$) as the mid-points of all line segments in the control polyline, i.e. $P_1^i = (P_0^i + P_0^{i+1})/2$, ($i = 1 \cdots n - 1$).

2. Similarly, define $P_2^i$ ($i = 2 \cdots n - 1$) as the mid-points of all new line segments formed by $P_1^i$, i.e. $P_2^i = (P_1^{i-1} + P_1^i)/2$, ($i = 2 \cdots n - 1$).

3. Continue doing the above for each newly formed polyline, i.e. $P_k^i = (P_{i-1}^{k-1} + P_k^i)/2$, where $k = 1 \cdots n - 1$, and $i = k \cdots n - 1$. When $k = n - 1$, there is only one point left, the process is then complete.

After this subdivision, the original control polyline is divided into two separate control polylines: $P_i^n$ ($i = 0 \cdots n - 1$), and $P_{n-1}^i$ ($i = n - 1 \cdots 0$), as shown in Figure 1(b). Each of these two polylines represents half of the curve. But they are now closer to the curve than the original polyline. We call the original polyline the level-0 approximation, and the two refined polylines the level-1 approximation. Naturally, the above procedure can be applied to each of the level-1 polylines to generate four new (more refined) polylines for level-2 approximation to the curve. The process can continue for the level-2 polylines and all subsequent higher level polylines to generate level-3, level-4, ... polylines.
In this project, you will write an interactive program for the definition and drawing of Bezier curve.

1. You will modify the program of your first project to interactively define the initial control polyline (note: it is a line strip, not a closed polygon).

2. When drawing the curve, the following menu and keyboard events/functions need to be defined:

   - **a number** $i$, ($i = 1 \cdots 9$): draw the level-$i$ polylines generated from the above procedure.
   - **“+”** or **“-”**: increase or decrease the current level of approximation by 1.
   - **“p”** or **“P”**: toggle the drawing of the original (level-0) control polyline. By default, the level-0 polyline is drawn along with the curve (i.e. the current approximation of the curve). When the original polyline is drawn, vertex editing and deletion are also enabled.

3. Your program should allow the user to resize the window, and the aspect ratio of your graphics needs to be maintained.

Figure 1: The Bezier curve subdivision algorithm