Programming #5 (4190.410)

Due: November 28, 2012

A cubic Bézier curve $C(t) = \sum_{l=0}^{3} \mathbf{b}_l B_l^3(t)$, $0 \le t \le 1$, can be approximated by a polygonal curve $L^h(t)$ connecting a sequence of curve points $C(t_i^h) = C(i/2^h)$, for $i = 0, \cdots, 2^h$, within an approximation error bound (Filip et al., CAGD 1986):

$$\|C(t) - L^h(t)\| \le \frac{3}{4} \cdot \frac{1}{4^h} \cdot \max\left(\|\mathbf{b}_0 - 2\mathbf{b}_1 + \mathbf{b}_2\|, \|\mathbf{b}_1 - 2\mathbf{b}_2 + \mathbf{b}_3\|\right) = \epsilon_h.$$

More precisely, each line segment $L_i^h(t)$, $(t_{i-1}^h \le t \le t_i^h)$, approximates the corresponding curve segment $C_i^h(t) = C(t)$, $(t_{i-1}^h \le t \le t_i^h)$, within the error bound $\epsilon_h \ge 0$.

Design an interactive system that can edit an infinite line $ax + by + c = 0$ by controlling two points $\mathbf{p}_0$ and $\mathbf{p}_1$ on the line, and the cubic Bézier curve $C(t)$ by dragging the four control points $\mathbf{b}_l$. Moreover, implement a recursive algorithm for computing the intersection between the cubic Bézier curve $C(t)$ and the infinite line $ax + by + c = 0$.

1. Design a recursive algorithm for computing the signed distances d_{i-1}^h and d_i^h of the endpoints of $L_i^h(t)$ from the line $ax + by + c = 0$.

   (a) If the two signed distances are both less than $-\epsilon_h$ or both larger than ϵ_h (i.e., $[(d_{i-1}^h < -\epsilon_h) \wedge (d_i^h < -\epsilon_h)]$ or $[(d_{i-1}^h > \epsilon_h) \wedge (d_i^h > \epsilon_h)]$), there will be no intersection between $C_i^h(t)$ and the line $ax + by + c = 0$.

   (b) Otherwise, go down to the next level of the recursive approximation by evaluating the curve at the midpoint $C(\frac{1}{2}(\frac{i-1}{2^h} + \frac{i}{2^h})) = C_i^h(\frac{t_{i-1}^h + t_i^h}{2}) = C_{2i}^{h+1}(t_{2i-1}^{h+1})$.

2. Repeat the same procedure recursively until the maximum level $h = 10$ and report an appropriate point of the correspondig line segment $L_i^h(t)$ as an approximate intersection point.

3. Each time you modify the curve $C(t)$ or the line $ax + by + c = 0$, recompute the intersection points.

4. To speed up the curve evaluation $C(t_i)$, it is important to precompute the basic functions $B_l^3(t_i)$, for $l = 0, 1, 2, 3$, at all finest parameter values $t_i = i/2^{10}$, $i = 0, \cdots, 2^{10}$. Note that these function values are fixed