CSE410/CIS510 – Curves and Surfaces: Theory and Applications
Fall 2005, 303 Towne, TR 10:30AM-12PM

Homework Assignment 2

Due: Wednesday, October 20, 2005, by 10.30 AM (IN CLASS)

1. [**10 points**]

   Write a procedure to compute the barycentric coordinates of a given point $p \in \mathbb{A}^2$ with respect to a given affine frame (p_0, p_1, p_2). If you are using Matlab, your procedure can be written as a functions as follows

   ```
   function out=barcoords(af,pc)
   ```

   The parameter `af` is a 2×3 matrix whose columns are the coordinates of p_0, p_1, and p_2. The parameter `pc` is a vector with the coordinates of p. Your procedure must verify if (p_0, p_1, p_2) is an affine frame. If so, the procedure computes the barycentric coordinates of p and return them in `out`. Otherwise, it returns $(0, 0, 0)$ in `out` to indicate that (p_0, p_1, p_2) is not an affine frame.

2. [**30 points**]

   This problem has a close relation with *ray tracing*. If you are taking CSE460/CIS560 or if you intend to, you will find this problem very useful. Your goal here is to write a *robust* procedure to compute the intersection point of a ray and a triangle in $\mathbb{A}^3$. A *ray* r is defined by a point $o \in \mathbb{A}^3$ and a vector $\vec{d} \in \mathbb{A}^3$ as follows:

   $$r = \{r(t) \in \mathbb{A}^3 \mid r(t) = o + t \cdot \vec{d},\ t \in \mathbb{R} \text{ and } t \geq 0\}.$$

   The point o is the *ray origin* and the vector $\vec{d}$ is the *direction vector*. Given a triangle Δ in $\mathbb{A}^3$ with vertices p_0, p_1, and p_2, a ray r intersects Δ if, and only if, there exists $t \in \mathbb{R}$, with $t \geq 0$, such that $r(t) = o + t \cdot \vec{d}$ is a point of Δ.

   You are now asked to write a procedure to compute the intersection of a given ray r and a given triangle Δ. Your procedure takes as input the ray origin o, the ray direction vector $\vec{d}$, and the vertices p_0, p_1, and p_2 of Δ. The output is a real number t such that $r(t)$ is the intersection of r and Δ if such intersection exists, and $t = -1$ if r does not intersect Δ. To check if r intersects Δ, you first computes the intersection point of r and the plane defined by Δ. How do you do this?

   Recall that a plane can be represented by a point n_0 in the plane and a (normal) vector $\vec{n}$. So, you just have to choose one of p_0, p_1, and p_2 as the point n_0, and let $\vec{n}$ be the cross product of the vectors $\overrightarrow{p_0p_1}$ and $\overrightarrow{p_0,p_2}$, $\vec{n} = \overrightarrow{p_0,p_2} \times \overrightarrow{p_0p_1}$. A point p is a point in the plane defined by

n_0 and $\vec{n}$ if, and only if, $\vec{n_0 p} \cdot \vec{n} = 0$. So, to find out if r intersects this plane, we just have the value of t that satisfies the equation $\overrightarrow{n_0 r(t)} \cdot \vec{n} = 0$. So,

$$t = -\frac{\vec{on}_o \cdot \vec{n}}{\vec{d} \cdot \vec{n}}.$$

Note that the above equation can only be solved if $\vec{d} \cdot \vec{n} \neq 0$. If $\vec{d} \cdot \vec{n} = 0$, then the plane and the ray are *parallel*, and in this case you can consider that there is no intersection between the ray and the plane.

Suppose that the ray and the plane are not parallel. So, $r(t)$ is a point in the plane defined by Δ. Now, all we have to do is to find out whether $r(t)$ is inside Δ. Here is where you will use your knowledge of barycentric coordinates and affine maps. The idea is to *project* Δ and $r(t)$ onto one of the XY, XZ and YZ plane, and then compute the barycentric coordinates of the projection of $r(t)$ with respect to the affine frame resulting from the projection of Δ. Why do we do this? Well, first note that Δ is an affine frame for the plane defined by it. So, every point in this plane (including $r(t)$) can be uniquely written as an affine combination of the vertices of Δ. If we know the barycentric coordinates of $r(t)$ with respect to Δ, all we have to do is to examine the signs of such coordinates to find out whether $r(t)$ is inside or outside Δ. However, since $r(t)$ and Δ are in $\mathbb{A}^3$, we must choose two of the three coordinates of $r(t)$, p_0, p_1, and p_2 to assemble our familiar 3×3 linear system to compute the barycentric coordinates. This is the same thing as saying that we "project" $r(t)$ and Δ onto one of the XY, XZ and YZ plane. But, which one do we pick? Does it matter? Yes, it really does. You must choose a plane such that the projection of Δ is also a triangle. I will leave for you the problem of choosing an appropriate plane. You can do that very efficiently. Just think hard about it. Finally, why are the barycentric coordinates of $r(t)$ with respect to Δ the same as the barycentric coordinates of the projection of $r(t)$ with respect to the projection of Δ? The answer is simple: such a projection is an affine map and affine combinations are preserved by affine maps.

3. **[20 points]**

   Write a procedure to compute the values of all Bernstein polynomials of degree n at a given $t \in \mathbb{R}$. Your procedure takes as input t and n, where $t \in [0, 1]$ and n must be equal to or greater than 1, and then returns a vector of $n + 1$ elements such that the i-th element of this vector is the value of $B_i^n(t)$, for $0 \leq i \leq n$. If you are using Matlab, your procedure can be written as a functions as follows:

   ```
   function out=bernpolys(n,t)
   ```

   where `out` will contain the vector of values of Bernstein polynomials. Recall that you can compute $B_i^n(t)$ using the definition of $B_i^n(t)$ given in class, but you can also use the property $B_i^n(t) = (1 - t)B_i^{n-1}(t) + tB_{i-1}^{n-1}(t)$ and come up with a procedure that looks like the de Casteljau algorithm.

4. **[20 points]**

   Using the procedure you developed for the previous problem, write another procedure for computing a point at a Bézier curve. Your procedure takes as input a control polygon with $n+1$ vertices, where $n \geq 1$, an affine frame $[r, s]$ of $\mathbb{A}$ and a value $t \in [r, s]$. The output is the point of the Bézier curve defined by the given control polygon and the parameter value t. If you are using Matlab, your procedure can be written as a functions as follows:

```
function out=bezier(cp,r,s,t)
```

where `out` will contain the point corresponding to `t` on the curve defined by `cp`.

5. [**20 points**]

   Using the procedure you developed for the previous problem, write procedures for *plotting* 2D and 3D Bézier curves. To make your life easier, use some software with graphics capabilities, such as Matlab, Mapple or Mathematica. Finally, use your procedures to plot the two curves given by the following control points:

   a) $b_0 = (-\frac{9}{4}, 0)$, $b_1 = (-1, -\frac{3}{4})$, $b_2 = (-3, -2)$, $b_3 = (-3, -3)$, and $[r, s] = [0, 1]$.
   b) $b_0 = (0, 0, 0)$, $b_1 = (\frac{1}{3}, 0, 0)$, $b_2 = (\frac{2}{3}, \frac{1}{3}, 0)$, $b_3 = (1, 1, 1)$, and $[r, s] = [0, 1]$.