Today I will give an introduction to the topic of smooth curves and surfaces in $\Re^3$.

Cubic Curves (and splines)

Up to now, we have mostly considered simple linear shapes – lines, polygons — and second order shape, i.e. quadrics. (Last lecture was an obvious exception, where we considered fractals.) Today we consider smooth parametric curves and surfaces that are defined by third order polynomials. Such curves and surfaces are useful for two reasons. First, they can be used to define the paths of points in an animation. These paths can be the trajectory of an object, or they can be the paths of the camera. Second, smooth curves can be used to define shapes, not just the shapes of 1D objects (curves) but also shapes of surfaces as we'll see in the second part of this lecture.

Many times in this course we have defined a line (or ray) using a parametric curve. Suppose we have two points $\mathbf{p}(0), \mathbf{p}(1)$. Then we define a line that passes between them by

$$\begin{aligned}\mathbf{p}(t) &= \mathbf{p}(0) \ + \ t \ (\mathbf{p}(1) - \mathbf{p}(0)) \\ &= (1-t) \ \mathbf{p}(0) \ + \ t \ \mathbf{p}(1)\end{aligned}$$

Suppose we want to define a 3D curve that passes through three points $\mathbf{p}(0), \mathbf{p}(1), \mathbf{p}(2) \in \Re^3$. This is relatively easy. We can always fit a circle to any three points, as long as they are not collinear. We can define a parametric equation for a circle but this requires sines and cosines and they are expensive to compute. It turns out to be easier to go to a slightly more general model, the cubic curve.

Let's define a 3D parametric curve

$$\mathbf{p}(t) = (x(t), y(t), z(t))$$

using functions $x(t), y(t), z(t)$ that are polynomials of degree 3, namely

$$\begin{aligned}x(t) &= a_x t^3 + b_x t^2 + c_x t + d_x \\ y(t) &= a_y t^3 + b_y t^2 + c_y t + d_y \\ z(t) &= a_z t^3 + b_z t^2 + c_z t + d_z\end{aligned}$$

Since we have 12 coefficients in total, we might guess that we can specify a cubic curve by chosing four points in $\Re^3$ (since $4 \times 3 = 12$). This is indeed the case, as we see below.

The technique to fitting cubic curves boils down to the following idea. Consider the case $x(t)$ which we rewrite

$$x(t) \quad = \quad \begin{bmatrix} a_x & b_x & c_x & d_x \end{bmatrix} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix} . \tag{1}$$

We want to find coefficients a_x, b_x, c_x, d_x so that the function takes given values, say $x(0), x(1), x(2), x(3)$ at $t = 0, 1, 2, 3$. We can do so by substituting in for t and grouping into four columns.

$$\begin{bmatrix} x(0) & x(1) & x(2) & x(3) \end{bmatrix} \quad = \quad \begin{bmatrix} a_x & b_x & c_x & d_x \end{bmatrix} \begin{bmatrix} 0 & 1 & 8 & 27 \\ 0 & 1 & 4 & 9 \\ 0 & 1 & 2 & 3 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

We can then find the coefficients a_x, b_x, c_x, d_x by right multiplying by the inverse of the 4×4 matrix on the right side. We will use that inverse matrix again below. We call it $\mathbf{B}$.

$$\mathbf{B} \equiv \begin{bmatrix} 0 & 1 & 8 & 27 \\ 0 & 1 & 4 & 9 \\ 0 & 1 & 2 & 3 \\ 1 & 1 & 1 & 1 \end{bmatrix}^{-1}$$

We fit a curve $\mathbf{p}(t)$ to four points in $\Re^3$ using the same technique. Define $\mathbf{p}(t)$ to be a column vector:

$$\mathbf{p}(t) = \begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix} = \begin{bmatrix} a_x & b_x & c_x & d_x \\ a_y & b_y & c_y & d_y \\ a_z & b_z & c_z & d_z \end{bmatrix} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix} \tag{2}$$

Let $\mathbf{p}(0)$, $\mathbf{p}(1)$, $\mathbf{p}(2)$, $\mathbf{p}(3)$ be the four given values specified on the desired curve at $t = 0, 1, 2, 3$. We write these four points as a 3×4 matrix which is called a *geometry matrix*, or matrix of *control points.*

$$\mathbf{G} \equiv \begin{bmatrix} \mathbf{p}(0) & \mathbf{p}(1) & \mathbf{p}(2) & \mathbf{p}(3) \end{bmatrix}$$

Now substitute the four values $t = 0, 1, 2, 3$ into Eq. (2) and group the four substituted vectors into a four column matrix on both the left and right side:

$$\begin{bmatrix} \mathbf{p}(0) & \mathbf{p}(1) & \mathbf{p}(2) & \mathbf{p}(3) \end{bmatrix} = \begin{bmatrix} a_x & b_x & c_x & d_x \\ a_y & b_y & c_y & d_y \\ a_z & b_z & c_z & d_z \end{bmatrix} \begin{bmatrix} 0 & 1 & 8 & 27 \\ 0 & 1 & 4 & 9 \\ 0 & 1 & 2 & 3 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

We then solve for the a, b, c, d coefficients by right-multiplying by matrix $\mathbf{B}$ just as we did on the previous page. i.e.

$$\mathbf{G}\,\mathbf{B} = \begin{bmatrix} a_x & b_x & c_x & d_x \\ a_y & b_y & c_y & d_y \\ a_z & b_z & c_z & d_z \end{bmatrix}$$

We rewrite Eq. (2) as:

$$\mathbf{p}(t) = \begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix} = \mathbf{G}\,\mathbf{B} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}.$$

The product

$$\mathbf{B} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$

is a 4×1 column vector whose rows (single elements) are *blending functions.* These blending functions are polynomials of degree 3 in t which define the weights of each of the four control points

$\mathbf{p}(0), \mathbf{p}(1), \mathbf{p}(2), \mathbf{p}(3)$ to the curve point $\mathbf{p}(t)$. That is, for any t, $\mathbf{p}(t)$ is a linear combination of the four control points, and the blending functions specify the weights of the combinations.

Suppose we would like to make a curve that passes through n points instead of 4 points. How could we do this? One way would be to take points 1-4 and fit a cubic curve to these points. We could then take points 4-7 and fit a cubic curve to them and piece the two curves together, etc for points 7-10, 10-13, etc. The problem is that the curve might have a kink in it at points 4, 7, 10, etc that is, it might change direction suddenly. To avoid this problem, we need to model explicitly the derivative of the curve and make sure the derive is continuous at the points where we join the curve pieces together.

Hermite Curves (and splines)

Instead of using four points, we use two points $\mathbf{p}(0)$ and $\mathbf{p}(1)$ and we specify the tangent vector at each of these two points. By *tangent vector*, I mean the derivative with respect to the parameter t:

$$\mathbf{p}'(t) \equiv \frac{d\mathbf{p}(t)}{dt} \equiv lim_{\ \delta t \rightarrow 0} \frac{\mathbf{p}(t+\delta t) - \mathbf{p}(t)}{\delta t}$$

If t were interpreted as time, then the tangent vector would be the 3D velocity of a point following the curve. This interpretation is relevent, for example, in animations if we are defining the path of a camera or the path of a vertex.

Taking the derivative of Eq. (2) gives

$$\mathbf{p}'(t) = \begin{bmatrix} x'(t) \\ y'(t) \\ z'(t) \end{bmatrix} = \begin{bmatrix} a_x & b_x & c_x & d_x \\ a_y & b_y & c_y & d_y \\ a_z & b_z & c_z & d_z \end{bmatrix} \begin{bmatrix} 3t^2 \\ 2t \\ 1 \\ 0 \end{bmatrix}$$

Substituting for $t = 0$ and $t = 1$ for $\mathbf{p}(t)$ and $\mathbf{p}'(t)$ gives us a 3×4 Hermite geometry matrix:

$$\begin{aligned} \mathbf{G}_{Hermite} &\equiv \begin{bmatrix} \mathbf{p}(0) & \mathbf{p}(1) & \mathbf{p}'(0) & \mathbf{p}'(1) \end{bmatrix} \\ &= \begin{bmatrix} a_x & b_x & c_x & d_x \\ a_y & b_y & c_y & d_y \\ a_z & b_z & c_z & d_z \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}. \end{aligned}$$

Of course, you should realize here that we are just taking the 1D problem,

$$\begin{bmatrix} x(0) & x(1) & x'(0) & x'(1) \end{bmatrix} = \begin{bmatrix} a_x & b_x & c_x & d_x \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

and similarly for $y(t)$ and $z(t)$ and then stacking the three solutions. That is, there is no *interaction* of x, y, z in the solution.

As before, we can compute the coefficient matrix by right multiplying by the inverse of the 4×4 matrix at the right end of the above equation. Define this inverse to be

$$\mathbf{B}_{Hermite} \equiv \begin{bmatrix} 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}^{-1}$$

and define the 3×4 *geometry matrix* (or *control points*)

$$\mathbf{G}_{Hermite} \equiv \left[\begin{array}{cccc} \mathbf{p}(0) & \mathbf{p}(1) & \mathbf{p}'(0) & \mathbf{p}'(1) \end{array} \right].$$

Then we can write

$$\mathbf{p}(t) \;=\; \mathbf{G}_{Hermite}\,\mathbf{B}_{Hermite} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}. \tag{3}$$

The 4×1 column vector

$$\mathbf{B}_{Hermite} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$

contains four *Hermite blending functions*. These four functions are each third order polynomials in t, and define the contribution of the four control points $\mathbf{p}(0), \mathbf{p}(1), \mathbf{p}'(0), \mathbf{p}'(1)$, respectively, to the curve $\mathbf{p}(t)$. That is, $\mathbf{p}(t)$ is a linear combination of $\mathbf{p}(0), \mathbf{p}(1), \mathbf{p}'(0), \mathbf{p}'(1)$ and the weights of this combination are the blending coefficients.

Now suppose we want to draw a more complicated curve in which we have n points $\mathbf{p}(0)$ to $\mathbf{p}(n-1)$ and we have tangents $\mathbf{p}'(0)$ to $\mathbf{p}'(n-1)$ defined at these points as well. We would like to fit a curve (say Hermite) between each pair of points. We can be sure that the curve passes through the points and that the tangents are continuous, simply by making the endpoint and tangent of one curve segment (say, from $t = k-1$ to $t = k$) be the same as the beginning point and tangent of the next curve segment (from $t = k$ to $t = k+1$). This defines a *Hermite (cubic) spline*.

Bézier Curves

A slightly different method for defining a curve is to specify the tangent vectors in terms of points that are not on the curve. Given $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$, define a curve $\mathbf{q}(\mathrm{t})$ where $t \in [0, 1]$. Let respectively, and define the control points $\mathbf{p}_1$ and $\mathbf{p}_2$ such that

$$\begin{aligned} \mathbf{q}(0) &= \mathbf{p}_0 \\ \mathbf{q}(1) &= \mathbf{p}_3 \\ \mathbf{q}'(0) &= \beta(\mathbf{p}_1 - \mathbf{p}_0) \\ \mathbf{q}'(1) &= \beta(\mathbf{p}_3 - \mathbf{p}_2) \end{aligned}$$

It is convenient to use $\beta = 3$ for reasons we will mention below, which gives:

$$\begin{aligned}\mathbf{G}_{Hermite} &= \begin{bmatrix} \mathbf{q}(0) & \mathbf{q}(1) & \mathbf{q}'(0) & \mathbf{q}'(1) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{p}_0 & \mathbf{p}_1 & \mathbf{p}_2 & \mathbf{p}_3 \end{bmatrix} \begin{bmatrix} 1 & 0 & -3 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & -3 \\ 0 & 1 & 0 & 3 \end{bmatrix}.\end{aligned}$$

Thus,

$$\mathbf{G}_{Hermite}\, \mathbf{B}_{Hermite} = \mathbf{G} \begin{bmatrix} 1 & 0 & -3 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & -3 \\ 0 & 1 & 0 & 3 \end{bmatrix} \mathbf{B}_{Hermite}$$

and so if we define

$$\mathbf{B}_{Bezier} \equiv \begin{bmatrix} 1 & 0 & -3 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & -3 \\ 0 & 1 & 0 & 3 \end{bmatrix} \mathbf{B}_{Hermite}$$

we get

$$\mathbf{q}(t) = \begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix} = \mathbf{G}\, \mathbf{B}_{Bezier} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}.$$

It turns out that

$$\mathbf{B}_{Bezier} = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

The blending functions[1] for Bézier curve are particularly important:

$$\mathbf{B}_{Bezier} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix} = \begin{bmatrix} (1-t)^3 \\ 3t(1-t)^2 \\ 3t^2(1-t) \\ t^3 \end{bmatrix}$$

The key property of these functions is each is bounded by $[0, 1]$ over the interval $t \in [0, 1]$, and the four functions add up to 1 for each t. Thus, for all t, the point $\mathbf{q}(t)$ is a "convex combination"[2] of the points $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_n$. It is therefore contained within the tetrahedron defined by these four

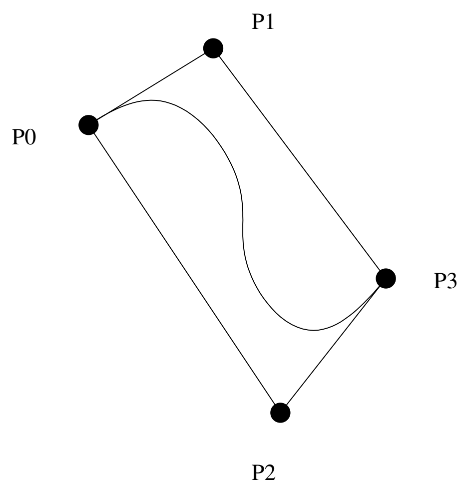
[1] They are called Bernstein polynomials of degree 3. The Bernstein polynomials of degree n are

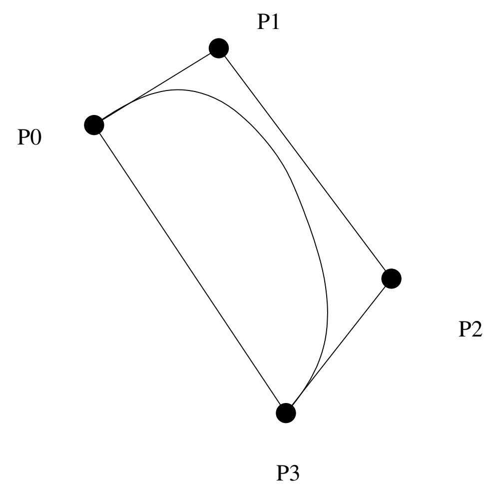
$$B_{i,n} = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i}.$$

[2] Given some points $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n \in \Re^3$ a convex combination of these points is a point of the form $\alpha_1 \mathbf{p}_1 + \alpha_2 \mathbf{p}_2 + \cdots + \alpha_n \mathbf{p}_n$ where $1 \geq \alpha_i \geq 0$ and $\alpha_1 + \alpha_2 + \cdots + \alpha_n = 1$.

points. This is useful for clipping: when we project the curve into the image plane, the projection of the four P points defines a convex polygon in the image plane that must contain the curve!

The following are two examples of Bézier curves. Note the positions of $\mathbf{p}_2$ and $\mathbf{p}_3$ are swapped.





Catmull-Rom splines

With Bezier splines, half the points serve as serve as anchors through which the spline passes, and half serve as controls which essentially define the tangent vectors. A slightly different approach is to make the spline pass through all the points and to use the points themselves to define the tangents. One example for doing this is the Catmull-Rom spline.

Suppose again we have a sequence of points $\mathbf{p}_0, \cdots \mathbf{p}_{n-1}$ and we want to fit a spline that passes through all of them. For any $\mathbf{p}_i$ and $\mathbf{p}_{i+1}$, define the tangent vectors $\mathbf{p}'_i$ and $\mathbf{p}'_{i+1}$ to be

$$\mathbf{p}'_i = \frac{1}{2}(\mathbf{p}_{i+1} - \mathbf{p}_{i-1})$$

$$\mathbf{p}'_{i+1} = \frac{1}{2}(\mathbf{p}_{i+2} - \mathbf{p}_i) \quad .$$

As an exercise, derive the geometry and blending matrices.

Bicubic Surfaces

We next turn from curves to surfaces. We wish to define a two parameter surface:

$$\mathbf{p}(s,t) = (x(s,t), y(s,t), z(s,t))$$

which maps

$$\mathbf{p} : \Re^2 \rightarrow \Re^3$$

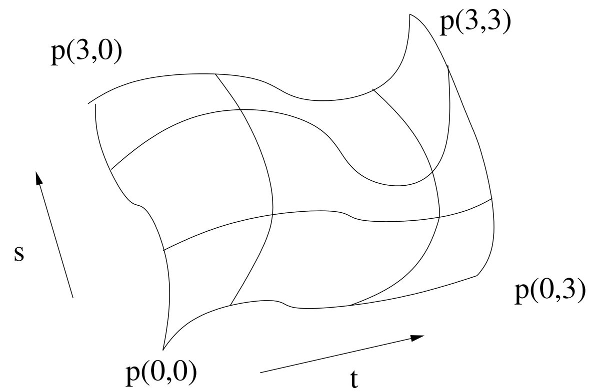
and is such that each $x(s,t), y(s,t)$, and $z(s,t)$ is a polynomial in s,t of (maximum) degree 3, and $\mathbf{p}(s,t)$ passes through a given set of 3D data points (see below). The surface $\mathbf{p}$(s,t) is a *parametric* *bicubic* surface with parameters s and t, in the following sense. For any fixed s, the curve $\mathbf{p}(s,t)$ is a cubic function of t and, for any fixed t, the curve $\mathbf{p}(s,t)$ is a cubic function of s.

The construction of this surface builds on the construction for cubic curves. For any four distinct points in $\Re^3$, we can define a cubic curve that passes through these four points. We use four sets of four points, giving us four cubic curves.

We fit a *bicubic surface* such that it passes through a 4×4 array of arbitrary 3D points

$$\{\mathbf{p}(s,t) : s,t \in \{0,1,2,3\}.$$

These sixteen points are the intersections of the eight curves shown in the following sketch. The eight curves correspond to four values $(0,1,2,3)$ for each of the s,t variables that define the surface. Only the four corner points of the surface patch are labelled in the figure.



To define the surface, we need to specify four functions, $x(s,t)$, $y(s,t)$, and $z(s,t)$. I'll present the construction for $x(s,t)$ only.

For each of the four values of $s \in \{0,1,2,3\}$, we define a cubic curve $x(s,t)$ over $t \in \Re$ by applying the solution from the beginning of the lecture. This gives:

$$x(s,t) = \begin{bmatrix} x(s,0) & x(s,1) & x(s,2) & x(s,3) \end{bmatrix} \mathbf{B} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}. \tag{4}$$

Next, take these x component of the four curves $s = 0,1,2,3$ and stack them. This gives the four horizontally oriented curves in the above sketch.

$$\begin{bmatrix} x(0,t) \\ x(1,t) \\ x(2,t) \\ x(3,t) \end{bmatrix} = \begin{bmatrix} x(0,0) & x(0,1) & x(0,2) & x(0,3) \\ x(1,0) & x(1,1) & x(1,2) & x(1,3) \\ x(2,0) & x(2,1) & x(2,2) & x(2,3) \\ x(3,0) & x(3,1) & x(3,2) & x(3,3) \end{bmatrix} \mathbf{B} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix} \tag{5}$$

The left hand side of Eq. (5) is interesting. For any t, it is a four-tuple. To fit a cubic to this four-tuple (for fixed t), we apply the solution again, but now the curve parameter is s. The illustration above shows the s curves for the particular case of $t \in \{0,1,2,3\}$. These are the four vertically oriented curves in the above sketch.

To obtain the s curves, we need to deal with a minor subtlety in notation. The four-tuple on the left side of (5) is written a column vector whereas previously we wrote our four-tuple control points as row vectors. To use the same form as Eq. 4 and fit a cubic to the four points on the left side of (5), we use the transpose of the left side of (5), namely

$$\begin{bmatrix} x(0,t) & x(1,t) & x(2,t) & x(3,t) \end{bmatrix}$$

and fit a cubic of parameter $s \in \Re$

$$x(s,t) = \begin{bmatrix} x(0,t) & x(1,t) & x(2,t) & x(3,t) \end{bmatrix} \mathbf{B} \begin{bmatrix} s^3 \\ s^2 \\ s \\ 1 \end{bmatrix}. \tag{6}$$

We then take the transpose again (of both sides now) which of course this does nothing to the left side since it is a scalar or 1×1 matrix.

$$x(s,t) = \begin{bmatrix} s^3 & s^2 & s & 1 \end{bmatrix} \mathbf{B}^T \begin{bmatrix} x(0,t) \\ x(1,t) \\ x(2,t) \\ x(3,t) \end{bmatrix} \tag{7}$$

Finally, substitute (5) into the right side of (7), we get:

$$x(s,t) = \begin{bmatrix} s^3 & s^2 & s & 1 \end{bmatrix} \mathbf{B}^T \begin{bmatrix} x(0,0) & x(0,1) & x(0,2) & x(0,3) \\ x(1,0) & x(1,1) & x(1,2) & x(1,3) \\ x(2,0) & x(2,1) & x(2,2) & x(2,3) \\ x(3,0) & x(3,1) & x(3,2) & x(3,3) \end{bmatrix} \mathbf{B} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}.$$

This is our parametric bicubic function. The 4×4 $x(*,*)$ matrix is a geometrix matrix which we can call $\mathbf{G}_x$ and so

$$x(s,t) = \begin{bmatrix} s^3 & s^2 & s & 1 \end{bmatrix} \mathbf{B}^T \mathbf{G}_x \mathbf{B} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}. \tag{8}$$

We use the same formula to define $y(s,t)$ and $z(s,t)$ via geometry matrices $\mathbf{G}_y$ and $\mathbf{G}_z$.

Surface Normals

As we will see later in the course, it is often very useful to know the surface normal for any (s,t). How could we define this surface normal? Once we have the geometry matrice $\mathbf{G}_x, \mathbf{G}_y, \mathbf{G}_z$, we can compute tangent vectors to the surfaces for any (s,t), namely

$$\frac{\partial}{\partial t} x(s,t) = \begin{bmatrix} s^3 & s^2 & s & 1 \end{bmatrix} \mathbf{B}^T \mathbf{G}_x \mathbf{B} \begin{bmatrix} 3t^2 \\ 2t \\ 1 \\ 0 \end{bmatrix}.$$

and

$$\frac{\partial}{\partial s}x(s,t) = \begin{bmatrix} 3s^2 & 2s & 1 & 0 \end{bmatrix} \mathbf{B}^T \, \mathbf{G}_x \, \mathbf{B} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$

The same form is used for the partial derivatives of $y(s,t)$ and $z(s,t)$. Thus, for any (s,t) we have a way to exactly compute:

$$\frac{\partial}{\partial t} \begin{bmatrix} x(s,t) \\ y(s,t) \\ z(s,t) \end{bmatrix} \quad \text{and} \quad \frac{\partial}{\partial s} \begin{bmatrix} x(s,t) \\ y(s,t) \\ z(s,t) \end{bmatrix}.$$

From Calculus III (and intuition ?), these two partial derivatives are both tangent to the surface (and are not identical). SInce the surface normal must be perpendicular to both of these tangent vectors, the surface normal must be parallel to the cross product of these two partial derivatives.

$$\frac{\partial}{\partial s}\mathbf{p}(s,t) \times \frac{\partial}{\partial t}\mathbf{p}(s,t)$$

We will use this cross product method for defining the surface normal in a later lecture, when we discuss bump mapping, although there we won't be assuming necessarily that $\mathbf{p}(s,t)$ is a bicubic.