OpenGL Notes [a]

Stu Pomerantz
smp@psc.edu
`http://www.psc.edu/~smp`

December 8, 2004

[a] Most material is adapted from: OpenGL ARB, et. al, "The OpenGL Programming Guide", Third Ed., Reading: Addison-Wesley, 1999

2D Parametric Curves

In two dimensions a parametric curve is defined by:

$$x = x(u)$$
$$y = y(u)$$

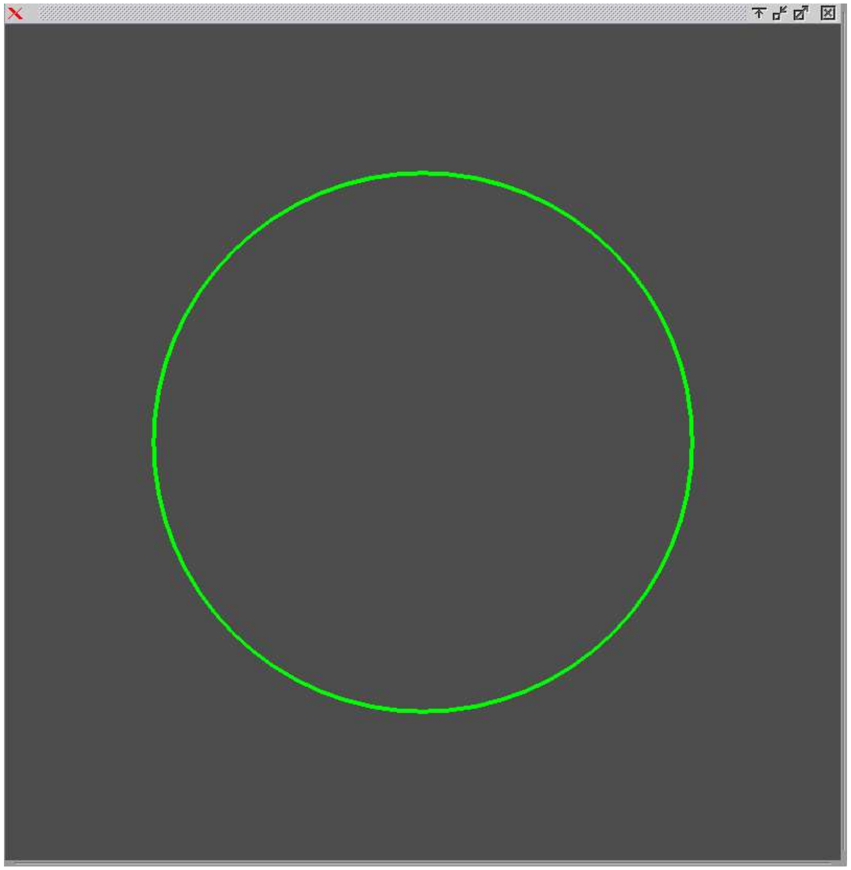
Where u is the parameter that is free to vary. For example:

$$x = \cos(\theta)$$
$$y = \sin(\theta)$$

x and y will trace out a circle as $0 \leq \theta \leq 2\pi$.

2D Parametric Curves

Drawing of the parametric circle:



2D Parametric Curves

Code for the parametric circle:

```
glBegin(GL_LINES) ;
du = PI/32 ;
for( u = 0 ; u < 8*PI ; u += du) {
  x = cos(u) ; y = sin(u) ; z = 0 ;
  glVertex3f(x,y,z) ;

  x = cos(u+du) ; y = sin(u+du) ; z = 0 ;
  glVertex3f(x,y,z) ;
}
glEnd() ;
```

3D Parametric Curves

In two dimensions a parametric curve is defined by:

$$x = x(u)$$
$$y = y(u)$$
$$z = z(u)$$

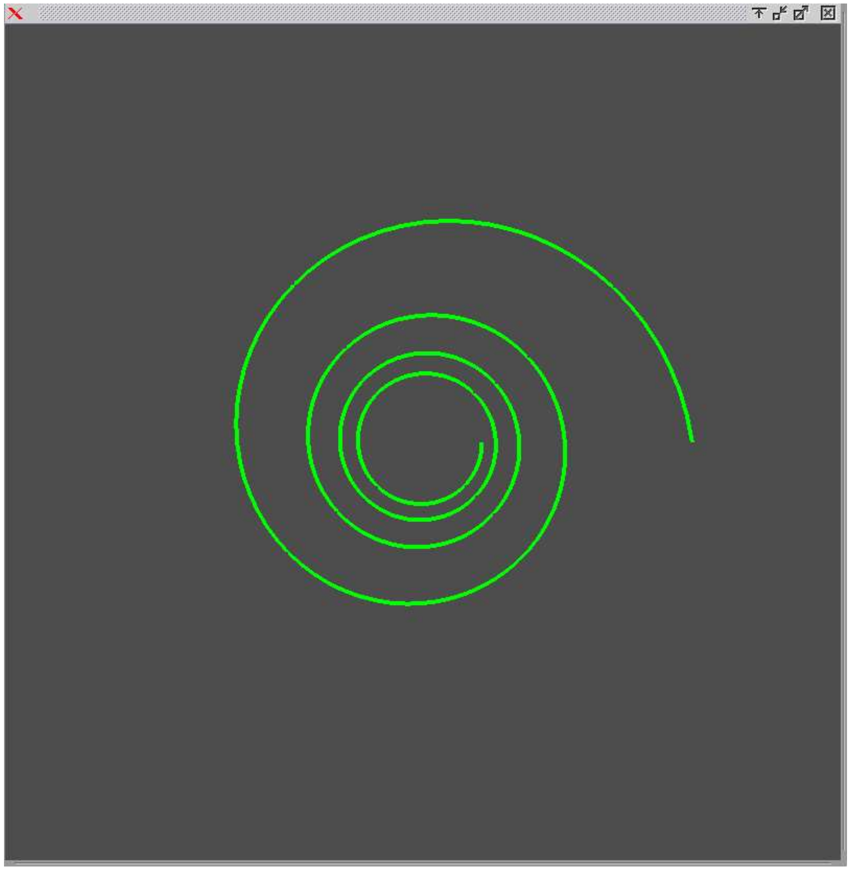
Where u is the parameter that is free to vary. For example:

$$x = \cos(\theta)$$
$$y = \sin(\theta)$$
$$z = \theta$$

x, y and z will trace out a circlular helix as $0 \leq \theta \leq 2\pi$.

3D Parametric Curves

Drawing of the parametric helix:



3D Parametric Curves

Code for the parametric helix:

```
glBegin(GL_LINES) ;
du = PI/32 ;
for( u = 0 ; u < 8*PI ; u += du) {
  x = cos(u) ; y = sin(u) ; z = u ;
  glVertex3f(x,y,z) ;

  x = cos(u+du) ; y = sin(u+du) ; z = u+du ;
  glVertex3f(x,y,z) ;
}
glEnd() ;
```

3D Parametric Surfaces

In three dimensions a parametric surface is defined by:

$$x = x(u, v)$$
$$y = y(u, v)$$
$$z = z(u, v)$$

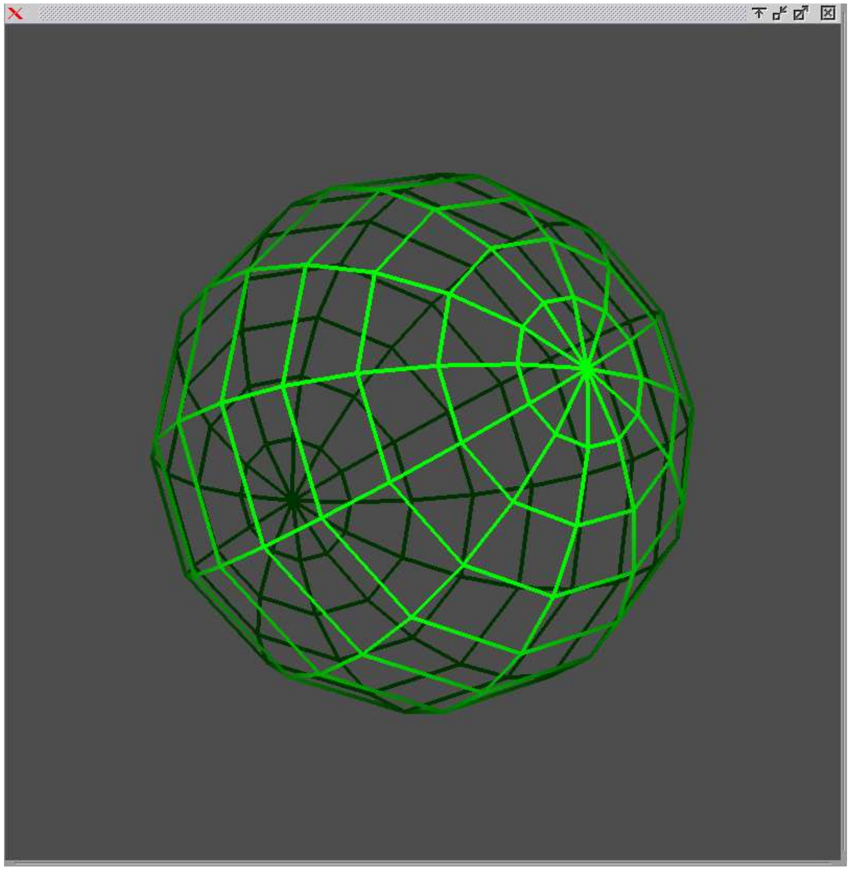
Where u and v are the parameters that are free to vary. For example:

$$x = \cos(\theta)\sin(\phi)$$
$$y = \sin(\theta)\sin(\phi)$$
$$z = \cos(\phi)$$

x, y and z will trace out a sphere as $0 \leq \theta \leq 2\pi$ and $0 \leq \phi \leq 2\pi$.

3D Parametric Surfaces

Drawing of the parametric sphere:



Polynomial Parametric Curves

$$\vec{p}(u) = \sum_{k=0}^{n} u^k \vec{c}^{\,k}$$

where

$$\vec{c}^{\,k} = \begin{pmatrix} c_{xk} \\ c_{yk} \\ c_{zk} \end{pmatrix}$$

Polynomial Parametric Curves

For example if $n = 3$,

$$\vec{p}(u) = \sum_{k=0}^{3} u^k \vec{c}^{\,k}$$

expands to:

$$\vec{p}(u) = u^3 \begin{pmatrix} c_{x3} \\ c_{y3} \\ c_{z3} \end{pmatrix} + u^2 \begin{pmatrix} c_{x2} \\ c_{y2} \\ c_{z2} \end{pmatrix} + u \begin{pmatrix} c_{x1} \\ c_{y1} \\ c_{z1} \end{pmatrix} + \begin{pmatrix} c_{x0} \\ c_{y0} \\ c_{z0} \end{pmatrix}$$

Polynomial Parametric Curves

Continuing,

$$\vec{p}(u) = u^3 \begin{pmatrix} c_{x3} \\ c_{y3} \\ c_{z3} \end{pmatrix} + u^2 \begin{pmatrix} c_{x2} \\ c_{y2} \\ c_{z2} \end{pmatrix} + u \begin{pmatrix} c_{x1} \\ c_{y1} \\ c_{z1} \end{pmatrix} + \begin{pmatrix} c_{x0} \\ c_{y0} \\ c_{z0} \end{pmatrix}$$

expands to

$$\vec{p}(u) = \begin{array}{r} x(u) = c_{x3}u^3 + c_{x2}u^2 + c_{x1}u + c_{x0} \\ y(u) = c_{y3}u^3 + c_{y2}u^2 + c_{y1}u + c_{y0} \\ z(u) = c_{z3}u^3 + c_{z2}u^2 + c_{z1}u + c_{z0} \end{array}$$

Notice that 4 coefficient vectors which must be specified.

Polynomial Parametric Surfaces

$$\vec{p}(u,v) = \begin{pmatrix} x(u,v) \\ y(u,v) \\ z(u,v) \end{pmatrix} = \sum_{i=0}^{n} \sum_{j=0}^{m} \vec{c}_{ij} u^i v^j$$

So, if $n = m = 3$ then 16 coefficient vectors will be needed to evaluate the formula and determine the value of $\vec{p}$

- u for curves, and u and v for surfaces have an infinite domain.
- Generally, though, only an interval of values for the domain of u and v is desired.
- So, without lost of generality, restrict the range of u and v to $[0, 1]$.

Interpolation Revisited

Two points, $\vec{p}_0$ and $\vec{p}_1$, can be linearly interpolated like this:

$$\vec{p} = a\ \vec{p}_0 + (1 - a)\ \vec{p}_1$$

where $a \in [0, 1]$

In this context, interpolation is also called *blending*. The points are mixed together by the a and $1 - a$ which are *weights*.

The points $\vec{p}_0$ and $\vec{p}_1$ can be thought of as *control points* since they influence (constrain) the final interpolated value.

This is similar, but not identical to, OpenGL's blending functions. And, in this case the purpose is *not* to produce transparent surfaces but rather to produce a new interpolated point.

Interpolation using Cubic Polynomials

Recall the cubic polynomial

$$\vec{p}(u) = \sum_{k=0}^{3} u^k \vec{c}^{\,k}$$

Requires 4 vectors of coefficients in order for it to be completely determined.

Rewrite this equation...

Interpolation using Cubic Polynomials

..using a matrix representation of the coefficient vectors.

$$\vec{p}(u) = [x(u)\ y(u)\ z(u)] =$$

$$\left[u^3\ u^2\ u\ 1\right] \cdot \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} \cdot \begin{bmatrix} g_{1x} & g_{1y} & g_{1z} \\ g_{2x} & g_{2y} & g_{2z} \\ g_{3x} & g_{3y} & g_{3z} \\ g_{4x} & g_{4y} & g_{4z} \end{bmatrix}$$

Or more briefly,

$$\vec{p}(u) = U \cdot M \cdot G$$

Call M the basis matrix. Call G the Geometry matrix.

Interpolation using Cubic Polynomials

For example,

$$\begin{aligned} x(u) = \quad & (u^3 m_{11} + u^2 m_{21} + u m_{31} + m_{41}) g_{1x} + \\ & (u^3 m_{12} + u^2 m_{22} + u m_{32} + m_{42}) g_{2x} + \\ & (u^3 m_{13} + u^2 m_{23} + u m_{33} + m_{43}) g_{3x} + \\ & (u^3 m_{14} + u^2 m_{24} + u m_{34} + m_{44}) g_{4x} + \end{aligned}$$

This form emphasizes that the curve is a weighted sum of the elements of G (geometry) matrix.

Interpolation using Cubic Polynomials

- The geometry matrix, G, contains 4 points. They are called *control points*.
- Analogous to the 2 control points in the previous linear interpolation example.
- The meaning of these 4 points is dependant on the basis matrix M.
- The basis matrix M affects
  - The continuity of the curve.
  - Which and whether control points are interpolated.

The Hermite Basis

The Hermite basis is named after the Mathematician. The construction the Hermite basis matrix is as follows:

Given 4 control points: $\vec{p}_1, \vec{p}_2, \vec{p}_3, \vec{p}_4$

- Constrain the curve so that it interpolates $\vec{p}_1$ at $u = 0$ and $\vec{p}_4$ at $u = 1$
- Let $\vec{p}_2$ and $\vec{p}_3$ be the tangent vectors for $\vec{p}_1$ and $\vec{p}_4$ respectively.

The Hermite Basis

Recall,

$$U = \quad \left[u^3 \ u^2 \ u \ 1\right]$$

$$so,$$

$$U' = \quad \left[3u^2 \ 2u \ 1 \ 0\right]$$

There are 4 equations and 4 unknowns:

$$\vec{p}_1 = [0 \ 0 \ 0 \ 1] \quad (u = 0)$$

$$\vec{p}_4 = [1 \ 1 \ 1 \ 1] \quad (u = 1)$$

$$\vec{p}_2 = [0 \ 0 \ 1 \ 0] \quad (u' = 0)$$

$$\vec{p}_3 = [3 \ 2 \ 1 \ 0] \quad (u' = 1)$$

The Hermite Basis

Rewriting the 4 equations above as a matrix:

$$\begin{bmatrix} \vec{p}_1 \\ \vec{p}_4 \\ \vec{p}_2 \\ \vec{p}_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix}$$

For these equations to be satisfied M, the Hermitian basis matrix must be the inverse of this matrix.

The Hermite Basis

The Hermitian basis matrix:

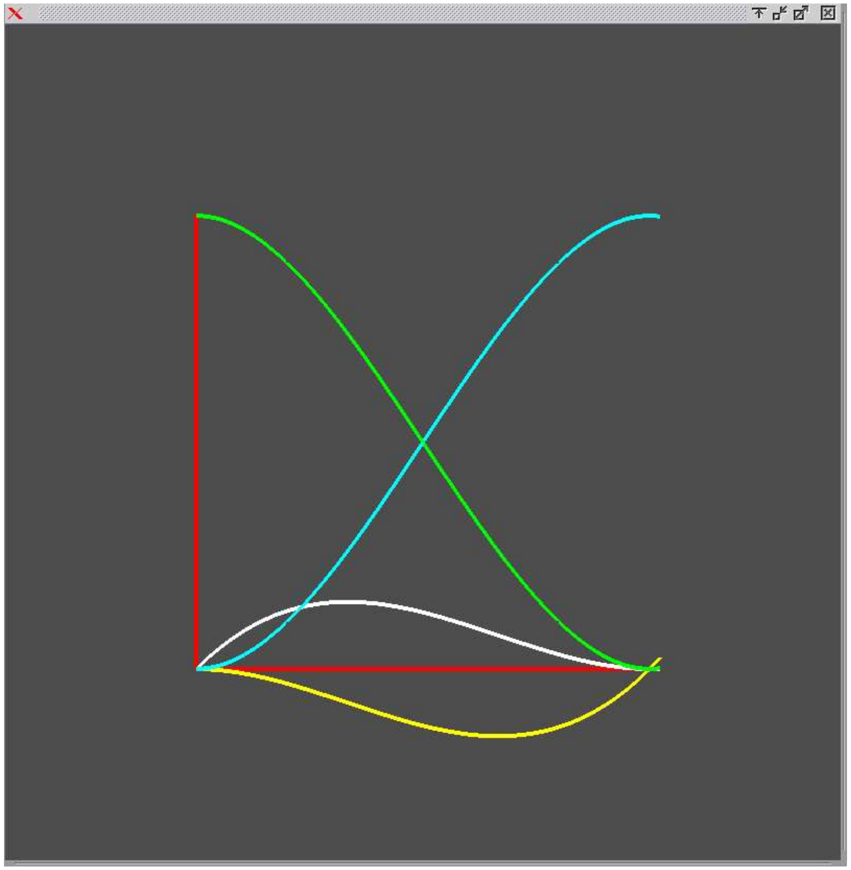
$$M = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

So,

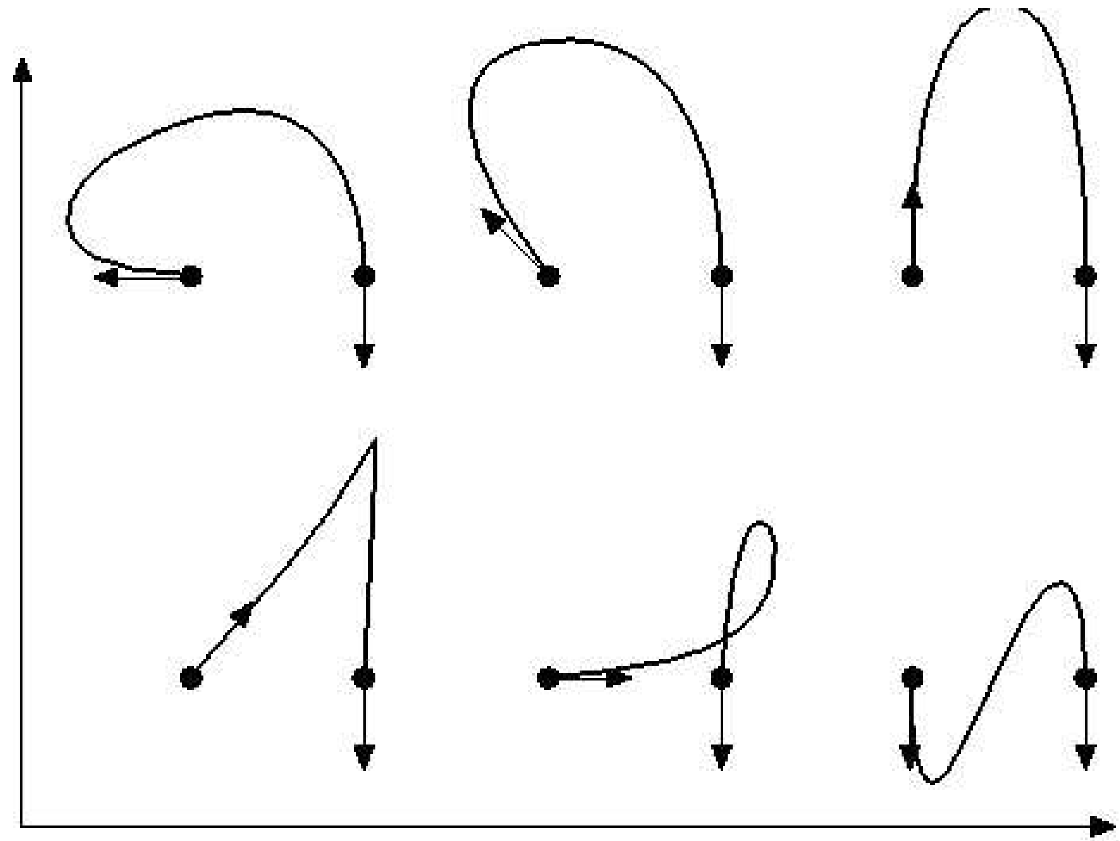
$$\begin{aligned} \vec{p}(u) = \quad & (2u^3 - 3u^2 + 1)\vec{p}_1 \ + (-2u^3 + 3u^2)\vec{p}_4 \ + \\ & (u^3 - 2u^2 + u)\vec{p}_3 \ + (u^3 - u^2)\vec{p}_4 \end{aligned}$$

The Hermite Basis

A picture of the Hermitian basis functions:



Example Hermite Curves



Comparison of Curve Types

Curves differ by basis matrix.

	Hermite	Bezier	Non-Uniform B-Spline	Catmull-Rom
A	Y	Y	N	Y
B	Y	N	N	Y
C	C^0	C^0	C^2	C^1

A = Interpolates some control points
B = Interpolates all control points
C = Inherent Continuity

If the direction and magnitude of the nth derivative are equal at the joint point, the curve is called C^n continuous.

Example NURBS Surface