Representing Geometry

* See Watt & Watt chapter 3

- CSG
- Polygons
- Parametric
- Implicit Surfaces
- Subdivision Surfaces

Each has its own strengths & weaknesses:

- ease of use for design
- ease/speed for rendering
- simplicity
- smoothness
- collision detection
- flexibility
- suitability for FEM
- etc...

* No one of these is best at everything.

Parametric Representations

Curve: \( x = x(u) \) \( x \in \mathbb{R}^n \), \( u \in \mathbb{R}^k \)

Surface: \( x = x(u,v) \) \( w \in \mathbb{R} \)

\( x = x(u) \) \( w \in \mathbb{R} \)

Volume: \( x(u,v,w) \) \( w \in \mathbb{R}^3 \)

and so on...

Parametric rep is not unique

\( x = [u, u] \)

or \( x = [2u, 2u] \)

or \( x = [u^2, u^3] \)

[Diff. Geo. Formulae assume normalize param]

or they include normalization

Surface normal: \( \hat{n} = \frac{\partial x \times \partial y x}{||\partial x|| \times ||\partial y||} \)
if \( x \) is any possible curve/surface

\[ \Rightarrow \text{hard to represent} \]

\[ \Rightarrow \text{How many parameters?} \]

\[ \Rightarrow \text{uncountable} \]

\underline{Being Practical:}

\textbf{Step 1 - Pick reasonable/useful subspace}

\textbf{Step 2 - Pick reasonable/useful basis functions}

\[ x(u) = \sum_{i=0}^{\infty} c_i \phi_i(u) \]

\[ \text{Still in finite number of parameters but countable} \]

\textbf{Step 3 - Truncate sum after finite number of terms}

\textbf{Note:} Could also pick something that is not linear in the \( c_i \) but that makes life hard so we won't do that. (Think about NURBS later on...)

\textbf{Examples}

- Fourier Series \( \Rightarrow \) the \( \phi_i \) are \( \cos/\sin \)
- Polynomials \( \Rightarrow \) the \( \phi_i \) are \( u^i \)

But \( x \in \mathbb{R}^n \mathbb{R}^3 \) (bc. we care about \( \mathbb{R}^3 \) and it's hard to stay generic.)

so let \( c_i \in \mathbb{R}^n \mathbb{R}^3 \)
A Closer look at Polynomials:
\[ x(u) = \sum_{i=0}^{d} c_i u^i = c \cdot P^u \]
where \[ c = [c_0, c_1, c_2, \ldots, c_d] \]
\[ P^u = [1, u, u^2, u^3, \ldots, u^d] \]
\[ \phi_i(u) = u^i \]

"Power Basis"
- Elements of \( P^u \) are linearly independent.

\[ \Rightarrow \text{no good approx of } u^k \text{ w/ } \sum_{j \neq k} u^j \]

Why use \( u^0, \ldots, u^d \)?
Why not \( u^j \) with \( j = 0, 2, 4, 8 \ldots 2^k \)
or \( u^j \) if \( j \) is first \( k \) primes?

Task: Pick \( c_i \) to generate some useful curve.

Imagine we know \( x(0), x(1), x'(0) \) & \( x'(1) \)
\[ \text{\& } d = 3 \text{ (cubic polynomials)} \]

Note:
\[ x(0) = c_0 \]
\[ x'(0) = c_1 \]
\[ x(1) = \sum c_i \]
\[ x'(1) = \sum i c_i \]
\[
\begin{bmatrix}
X(0) \\
X(1) \\
X'(0) \\
X'(1)
\end{bmatrix} =
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 \\
0 & 0 & 1 & 0 \\
0 & 1 & 2 & 3
\end{bmatrix}
\begin{bmatrix}
C_0 \\
C_1 \\
C_2 \\
C_3
\end{bmatrix}
\]

\[C_{call} \Rightarrow C = B_H^{-1}p\]

\[p = B_H C \Rightarrow C = B_H^{-1}p\]

\[
P_H = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & -1 & 2 \\
0 & 0 & -2 & 1
\end{bmatrix}
\]

\[\therefore x(u) = P^3 C = \underbrace{P^3 P_H p}_{p}\]

\[
= \begin{bmatrix}
1 + 5u - 3u^2 + 2u^3 \\
0 + 0u + 2u^2 - 2u^3 \\
0 + 1u - 2u^2 + 1u^3 \\
0 + 0u - 1u^2 + 1u^3
\end{bmatrix}
\]

\[
= \sum_{i=0}^{3} p_i b_i(u)
\]

\[\Rightarrow \text{Look Familiar?} \]

OK, I drew these poorly but you can plot them yourself to see what they look like... .

\[p^5 \text{ These } b_i(u) \text{ are known as the } \text{Hermite Basis}\]
Cubic Bézier

*Note: Bézier are related to Bernstein polys, but well talk about that later*

Constraints:

\[ x(0) = p_0 \quad x'(0) = 3(p_1 - p_0) \]
\[ x(1) = p_3 \quad x'(1) = 3(p_3 - p_2) \]

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
-3 & 3 & 0 & 0 \\
0 & 1 & 2 & 3
\end{bmatrix}
\begin{bmatrix}
x(0) \\
x'(0) \\
x(1) \\
x'(1)
\end{bmatrix}
= 
\begin{bmatrix}
1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 \\
0 & 1 & 0 & 0 \\
0 & 1 & 2 & 3
\end{bmatrix}
\begin{bmatrix}
c_0 \\
c_1 \\
c_2 \\
c_3
\end{bmatrix}
\]

\[ \Rightarrow \quad c = \begin{bmatrix}
1 & 0 & 0 & 0 \\
-3 & 3 & 0 & 0 \\
3 & -3 & 0 & 0 \\
-1 & 3 & -3 & 1
\end{bmatrix}
\begin{bmatrix}
p
\end{bmatrix}
\]

\[ \Rightarrow \quad \beta_H \beta_H = \beta_Z \beta_Z \quad \Rightarrow \quad \beta_Z = \beta_Z^{-1} \beta_H \beta_H
\]

* Bézier, Power Basis, & Hermite all span the same space

* Think of FF axes in \( R^3 \) ...
Useful Properties of a Basis

* Convex Hull

\[ \sum b_i(u) = 1 \quad \text{and} \quad b_i(u) \geq 0, \quad u \in \mathbb{R} \]

Bézier has this property:

Hermite & Power donut.

(?) Why is this a nice property?

* Invariance under some class of transformation

\[ x(u) = \sum p_i b_i(u) \quad \Rightarrow \quad XF(x(u)) = \sum XF(p_i) b_i(u) \]

Bézier invariant under affine \( XF \), but not \( proj \).

Hermite not inv. under either affine or \( proj \).

NURBS are inv. under \( Proj \).

(?) Why nice property?

* Others

- Local support
- Nice subdivision rules - we'll see in few minutes
- Orthogonality is Fourier
- Fast evaluation scheme
- Interpolate vs. approximate
Example of "nice" evaluation scheme for Bézier

De Casteljau Eval.

* Spend 10 minutes & write a program that does this.

Joining

For

\[ c' \equiv b - a = c - b \]
\[ c' \equiv b - a = c - b \]
\[ \| b - a \| \quad \| c - b \| \]

* If you change a, b, or c need to change one of the others as well

* But if you change a, b, or c you don't need to change anything that is not a, b, or c

3 Local support
Tensor Product Surfaces

Surface is the result of sweeping a curve through space

* replace control points \( p_i \) w curves in \( U \)

\[
x(u, v) = \sum_i g_i(u) b_i(u)
\]

* So

\[
x(u, v) = \sum_{ij} p_{ij} b_i(u) b_j(v)
\]

But \( b_{ij}(u, v) = b_i(u) b_j(v) \)

* No different

\[
\text{done } v \text{ then } u \text{ or } u \text{ then } v
\]

**

\[
x(u, v) = \sum_{ij} p_{ij} b_{ij}(u, v)
\]

Tangent vectors

\[
t_u = \frac{\partial x(u, v)}{\partial u} \quad t_v = \frac{\partial x(u, v)}{\partial v}
\]

\[
\mathbf{n} = t_u \times t_v
\]

But things can happen:

\[
x(u, v) = \sum_{ij} U_i U_j v_i v_j
\]

\[
|t_v| = 0 \text{ at } v = 0
\]