Interpolation and Approximation of functions
Part 1

Dinesh K. Pai

Some parts in Textbook Chapter 9

Today

- Announcements
  - Quiz 2 results will be available on Monday
- Assignment 2 spotlights
- Interpolation, continued
Today:

- Generalize

(1) Higher Dimensions: easy

Interpolate each coordinate separately.

If \( \overline{C}_0 = \begin{pmatrix} C_{0x} \\ C_{0y} \\ C_{0z} \end{pmatrix} \) and \( \overline{C}_1 = \begin{pmatrix} C_{1x} \\ C_{1y} \\ C_{1z} \end{pmatrix} \)

\( C_x = C_{0x} (1-t) + C_{1x} t \)

\( \vdots \)

or write as a vector

\( \overline{C} = \overline{C}_0 (1-t) + \overline{C}_1 t \)

(2) Higher degree polynomials: quadratic, cubic, ...

- quartic, quintic, ...

- deg=2

- deg=3

- deg=4

- deg=5

Rarely use deg > 3
Higher degree polynomials are too wiggly.

Can set "overfitting"

(3) Splines, i.e., piecewise polynomials

Let's look at linear interpolation as an:

\[ C(t) = C_0 (1-t) + C_1 t \]

\[ b_0(t) \]

\[ b_1(t) \text{ left deficit} \]

\[ = \sum_{i=0}^{1} C_i b_i(t) \]

\[ \text{This is really just a weighted average of pins with weights parametrized by } t \]

\[ b_i \text{ are called blending weights or blending functions} \]

\[ b_0(0) = 1, b_0(t) = 1-t \]

\[ b_1(0) = 0, b_1(t) = t \]

\[ b_0 + b_1 = (1-t) + t = 1 \]

So \( b_i \) form a "partition of unity"
How can we get higher degree interpolation, with the same properties?

\[ C(t) = C_0 b_0(t) + C_1 b_1(t) + C_2 b_2(t) \]

\[ b_0 = (1-t)^3 \quad b_1 = 3t(1-t)^2 \quad b_2 = 3t^2(1-t) \quad b_3 = t^3 \]

It turns out these correspond Bernstein polynomials.

\[ \begin{array}{c|cccc}
\text{deg} & b_0 & b_1 & b_2 & b_3 \\
\hline
0 & 1 & & & \\
1 & (1-t) & t & & \\
2 & (1-t)^2 & 2t(1-t) & t^2 & \\
3 & (1-t)^3 & 3(1-t)^2 & 3(1-t) & t^3 \\
\end{array} \]

§ Bézier Curves

Curves specified by control points

- Catmull-Rom, similar, used in animation.
- Can be generalized to produce smooth surfaces

"Begin patch"

"NURBS"

CISE 424 for more details