Matrix Form of Parametric Curves

Polynomial parametric curves can be written as a function of the parameter \( t \). For example, any cubic curve can be expressed as \( p(t) = at^3 + bt^2 + ct + d \) for some coefficients \( a, b, c, d \). The matrix form isolates the parameterization from the coefficients such that \( p(t) = tA \) where \( t = [t^3 \ t^2 \ t \ 1] \) and \( A \) is the basis matrix of the coefficients: \([a \ b \ c \ d]^T\). Directly specifying each entry of \( A \) is nonintuitive however, so we replace it with a convenient basis (e.g. user specifies points in space) and a change of basis matrix for the desired form (Bezier, Hermite, etc.). We now have \( p(t) = tMP \). \( M \) is the change of basis matrix, specific for each form, and \( P \) contains the basis information of the particular form.

Now consider a cubic Bezier curve: \( p(t) = \sum_{i=0}^{3} B^3_i(t)p_i \). We want to express this in the form \( p(t) = tM_{Bez}P_{Bez} \). \( P_{Bez} = [p_0 \ p_1 \ p_2 \ p_3]^T \) contains the user-provided control points. We want to find \( M_{Bez} \). Evaluate \( p(t) \) and rearrange a little and you get

\[
M_{Bez} = \begin{bmatrix}
-1 & 3 & -3 & 1 \\
3 & -6 & 3 & 0 \\
-3 & 3 & 0 & 0 \\
1 & 0 & 0 & 0
\end{bmatrix}.
\]

This can be done for any cubic form.

What dimension are the coefficients \( a, b, c, d \)?

How does this extend to higher or lower order curves?

Note: This representation unifies many representations of a curve of a given degree by expressing them as polynomials of \( u \), but is not the most numerically stable representation (Why?). And many textbook authors don’t like it.

Switching Among Cubic Forms

Let some general form have change of basis matrix \( M_F \). The change-of-basis matrices provide an easy way to switch among forms. For example, to get the basis information for some form \( F \) from the Bezier control points, \( P_F = M_F^{-1}M_{Bez}P_{Bez} \).

Tangents and Curvature

With the above form, computing parametric derivatives is easy. \( M \) and \( P \) are constant coefficients with respect to the parameter \( t \), so \( p'(t) = t'MP \) where \( t' = [3t^2 \ 2t \ 1 \ 0] \). Similarly for higher order derivatives.

For a Bezier curve, derive the expression for the tangent (first derivative). At the endpoints, the tangents will come to \( p'(0) = 3(p_1 - p_0) \) and \( p'(0) = \)
So this means that the endpoint tangent direction and magnitude can be specified by manipulating the two closest points only. This also turns out to be true for Bezier curves of any order.

For curvature, substitute in $t'' = [6t 2 0 0]$ instead of $t$. Note that curvature also has a direction and magnitude (consult a differential geometry text for an intuitive understanding).

4 Continuity

For polynomial parametric curves such as Bezier curves and similar forms, parametric derivatives are continuous for any order, that is $p^k(t)$ is continuous for any $k$. If we want to build bigger curves (surfaces) out of cubic curves (surfaces), we can enforce continuity constraints. If the endpoints of two adjacent curves are coincident, they are said to have $C^0$ parametric continuity. Note that this does not mean they join smoothly; their tangents or higher order derivatives will generally not be continuous where they meet. We enforce higher order parametric continuity by ensuring the higher order derivatives at the join are equal. For example, when joining two cubic Bezier curves, we can enforce $C^1$ continuity by manipulating the adjacent points of the two curves.

This restricts the possible shapes that can be modeled with the two adjacent curves, as the join now specifies internal control points. We can define a looser form of continuity, where only derivative direction is continuous, but magnitude is not. We call this geometric ($G^k$) continuity. For first order geometric ($G^1$) continuity, the tangent directions are now constrained at join points, but the user is still free to set their magnitudes.