1. Three point Interpolation using Hermite Curves
We are given two endpoints p₀ and p₁ and an intermediate point pᵢ
We also have the unspecified parametric variable uᵢ
And unit tangent vectors t₀ and t₁.

How can you compute a cubic Hermite curve that interpolates the three points and has the specified unit tangent vectors at the boundary?

A Hermite curve is defined as:

\[ p(u) = (2u³ - 3u² + 1)p(0) + (-2u³ + 3u²)p(1) + (u³ - 2u² + u)p''(0) + (u³ - u²)p''(1) \]

This can also be rewritten as:

\[ p(u) = F_1(u)p(0) + F_2(u)p(1) + F_3(u)p''(0) + F_4(u)p''(1) \]

where \( F_1(u) = 2u³ - 3u² + 1 \), \( F_2(u) = -2u³ + 3u² \), \( F_3(u) = u³ - 2u² + u \), \( F_4(u) = u³ - u² \)

For our particular case, we know the beginning and ending point so we can substitute them into the equation:

\[ p(u_i) = F_1(u_i)p_0 + F_2(u_i)p_1 + F_3(u_i)p''_0 + F_4(u_i)p''_1 \]

However the \( p'' \) derivative is not the same at the unit tangent vector so we can also rewrite these as

\[ p''_0 = c_0 t_0 \quad \text{and} \quad p''_1 = c_1 t_1 \].

\[ p(u_i) = F_1(u_i)p_0 + F_2(u_i)p_1 + F_3(u_i)c_0 t_0 + F_4(u_i)c_1 t_1 \]

We need to find \( p''_0 \) and \( p''_1 \) which means we need to find the values of \( c_0 \) and \( c_1 \). We can expand the equations out to their components to get a better idea of the system of equations and variables that need to be solved. Doing this results in:

\[
\begin{align*}
    x_i &= F_1(u_i)x_0 + F_2(u_i)x_1 + F_3(u_i)c_0 t_{x_0} + F_4(u_i)c_1 t_{x_1} \\
    y_i &= F_1(u_i)y_0 + F_2(u_i)y_1 + F_3(u_i)c_0 t_{y_0} + F_4(u_i)c_1 t_{y_1} \\
    z_i &= F_1(u_i)z_0 + F_2(u_i)z_1 + F_3(u_i)c_0 t_{z_0} + F_4(u_i)c_1 t_{z_1}
\end{align*}
\]

We have three equations and 3 unknowns, namely \( c_0 \), \( c_1 \), and \( u_i \). In trying to eliminate the first two constants, we can rewrite the first two equations in matrix notation to get:

\[
\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} F_1(u_i) + \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} F_2(u_i) + \begin{bmatrix} c_0 t_{x_0} \\ c_0 t_{y_0} \end{bmatrix} F_3(u_i) + \begin{bmatrix} c_1 t_{x_1} \\ c_1 t_{y_1} \end{bmatrix} F_4(u_i)
\]

and

\[
\begin{bmatrix} z_i \\ y_i \end{bmatrix} = z_0 F_1(u_i) + z_1 F_2(u_i) + \begin{bmatrix} c_0 t_{z_0} \\ c_1 t_{z_1} \end{bmatrix} \begin{bmatrix} F_3(u_i) \\ F_4(u_i) \end{bmatrix}
\]

However, looking at this the x and y equation is not of the proper form for substitution. So rewriting it provides:

\[
\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} F_1(u_i) + \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} F_2(u_i) + \begin{bmatrix} c_0 t_{x_0} \\ c_0 t_{y_0} \end{bmatrix} F_3(u_i) + \begin{bmatrix} c_1 t_{x_1} \\ c_1 t_{y_1} \end{bmatrix} F_4(u_i)
\]

However, this is not the right form either as the z equation will result with all the variables in it instead of one. The \( c \)'s are common to both expressions so the can be factored out in both the x and y as well as z. Now we have a new set of equations as:
\[
\begin{bmatrix}
x_i \\
y_i \\
z_i
\end{bmatrix} = \begin{bmatrix} x_0 \\
y_0 \\
0
\end{bmatrix} F_1(u_i) + \begin{bmatrix} x_1 \\
y_1 \\
1
\end{bmatrix} F_2(u_i) + \begin{bmatrix} t_{x0} \\
t_{y0} \\
t_{t0}
\end{bmatrix} \begin{bmatrix} c_0 F_3(u_i) \\
c_1 F_4(u_i)
\end{bmatrix}
\]

\[z_i = z_0 F_1(u_i) + z_1 F_2(u_i) + \begin{bmatrix} t_{z0} \\
t_{z1}
\end{bmatrix} \begin{bmatrix} c_0 F_3(u_i) \\
c_1 F_4(u_i)
\end{bmatrix}\]

Now solving the first equation gives us:

\[
\begin{bmatrix} t_{x0} & t_{x1} \\
t_{y0} & t_{y1}
\end{bmatrix}^{-1} \begin{bmatrix} x_i \\
y_i \\
z_i
\end{bmatrix} = \begin{bmatrix} x_0 \\
y_0 \\
0
\end{bmatrix} F_1(u_i) - \begin{bmatrix} x_1 \\
y_1 \\
1
\end{bmatrix} F_2(u_i) + \begin{bmatrix} c_0 F_3(u_i) \\
c_1 F_4(u_i)
\end{bmatrix}
\]

Substituting this into the z equation results in:

\[z_i = z_0 F_1(u_i) + z_1 F_2(u_i) + \begin{bmatrix} t_{z0} \\
t_{z1}
\end{bmatrix} \begin{bmatrix} t_{x0} & t_{x1} \\
t_{y0} & t_{y1}
\end{bmatrix}^{-1} \begin{bmatrix} x_i \\
y_i \\
z_i
\end{bmatrix} - \begin{bmatrix} x_0 \\
y_0 \\
0
\end{bmatrix} F_1(u_i) - \begin{bmatrix} x_1 \\
y_1 \\
1
\end{bmatrix} F_2(u_i) + \begin{bmatrix} c_0 F_3(u_i) \\
c_1 F_4(u_i)
\end{bmatrix}\]

This can be solved for \( u \) since it is a cubic equation. However, there is a simpler form if we can do something with \( F_1 \) and \( F_2 \). We know \( F_1(u) = 2u^3 - 3u^2 + 1 \) and \( F_2(u) = -2u^3 + 3u^2 \) which tells us that we can simplify the z equation with \( F_1(u) = 1 - F_2(u) \). Substituting this results in:

\[z_i = z_0 (1 - F_2(u_i)) + z_1 F_2(u_i) + \begin{bmatrix} t_{z0} \\
t_{z1}
\end{bmatrix} \begin{bmatrix} t_{x0} & t_{x1} \\
t_{y0} & t_{y1}
\end{bmatrix}^{-1} \begin{bmatrix} x_i \\
y_i \\
z_i
\end{bmatrix} - \begin{bmatrix} x_0 \\
y_0 \\
0
\end{bmatrix} (1 - F_2(u_i)) - \begin{bmatrix} x_1 \\
y_1 \\
1
\end{bmatrix} F_2(u_i)\]

But we know that \( \begin{bmatrix} t_{x0} & t_{x1} \\
t_{y0} & t_{y1}
\end{bmatrix}^{-1} \) so this can be substituted and multiplied out and simplified to get:

\[0 = \left( \frac{t_{x0} t_{y1} - t_{x1} t_{y0}}{t_{x0} t_{y1} - t_{x1} t_{y0}} \right) \left( -2u^3 + 3u^2 \right) (x_i - x_0) + \left( -2u^3 + 3u^2 \right) (z_i - z_0) - (z_i + z_0)\]

or \( 0 = -(u - 1)^2 (1 + 2u) x_i + x_0 + z_0 + u^2 (2u - 3)(x_i + z_0 - z_i) - z_i\)

This can be solved for \( u \) where \( 0 \leq u_i \leq 1 \) for our curve. After that we can substitute into the x and y equations and get \( c_0, c_1 \) where \( c_0, c_1 > 0 \) for the tangents to respect the path along the curve. Once we have these then we use the formulas \( p_0' \) and \( p_1' \) to get the remaining two variables.
2. We are given a cubic Bezier curve $P(t)$ in $\mathbb{R}^3$ with control points $p_0, p_1, p_2, p_3$. A cusp is when $P'(t_0) = 0$. We assume that the four points are not collinear or coplanar. Can $P(t)$ have a cusp?

If the Bezier curve is cubic, then the derivative of the Bezier curve is quadratic and based on three points $q_0, q_1,$ and $q_2$. We know from lecture that these points are $n \Delta P_i = n(P_i - P_{i-1})$ where $n = 3$ for a cubic curve. The points are:

$$q_0 = 3(p_1 - p_0)$$
$$q_1 = 3(p_2 - p_1)$$
$$q_2 = 3(p_3 - p_2)$$

We also know that at the cusp $P'(t) = 0$. So the tangent curve must go through zero.

However, we also know that the tangent curve is bounded by the points $q_0, q_1,$ and $q_2$. Also these points form a triangle that has a point through the origin. Since we assume that the four points are not collinear and no coplanar then we can actually state the plane equation for the tangent triangle as:

$$q_2 = Aq_0 + Bq_1$$

where $A, B$ are constants

Now taking this we can show a relationship between $p_3$ and the other points.

$$p_3 = p_2 + 1/3 q_2$$
$$= p_1 + 1/3 q_1 + 1/3 q_2$$
$$= p_0 + 1/3 q_0 + 1/3 q_1 + 1/3 q_2$$
$$= p_0 + 1/3 q_0 + 1/3 q_1 + 1/3 (Aq_0 + Bq_1)$$
$$= p_0 + (1/3 + A/3) q_0 + (1/3 + B/3) q_1$$
$$= p_0 + (1/3 + A/3) (3(p_1 - p_0)) + (1/3 + B/3) (3(p_2 - p_1))$$
$$= p_2 + A(p_1 - p_0) + B (p_2 - p_1)$$

So now we have shown that $p_3$ is really a combination of $p_0, p_1$ and $p_2$ which means that it must lie in the same plane at $p_0, p_1,$ and $p_2$. However, this contradicts our assumption that the four points are not coplanar. Therefore, a cubic Bezier curve cannot have a cusp.
3. We are given a Bezier curve $B(b_0, b_1, \ldots, b_n: t)$, specified in terms of control points. Given $0 \leq t_{\text{bar}} \leq 1$, show that the curve can be subdivided into two Bezier curves given as:

$$B(c_0, c_1, \ldots, c_n : t/t_{\text{bar}}), \quad 0 \leq t \leq t_{\text{bar}}$$

And

$$B(d_0, d_1, \ldots, d_n : (t-t_{\text{bar}})/(1-t_{\text{bar}})), \quad t_{\text{bar}} \leq t \leq 1$$

Where $c_i = b_0^{i}(t)$ and $d_i = b_0^{n-i}(t)$ and $0 \leq i \leq n$ and $b_i^{i}(t)$ is computed using de Casteljau’s algorithm.

Starting with the first curve:

$$B_c(t) = B\left(c_0, c_1, \ldots, c_n : \frac{t}{t_{\text{bar}}} \right)$$

Now we can rephrase this representation in terms of Bernstein polynomials

$$= \sum_{i=0}^{n} C_i B_i^n\left(\frac{t}{t_{\text{bar}}} \right)$$

At the beginning, we are told that $c_i = b_0^{i}(t)$ so substituting

$$= \sum_{i=0}^{n} b_0^{i}(t) B_i^n\left(\frac{t}{t} \right)$$

Now we have everything in terms of $b$’s and we can use the definition of $b_0^{i}(t) = \sum_{j=0}^{i} B_i^j(t) b_k$ from the Bernstein polynomials to get:

$$= \sum_{i=0}^{n} \left( \sum_{j=0}^{i} B_i^j(t) b_k \right) B_i^n\left(\frac{t}{t} \right)$$

It would be nice to use the subdivision formula: $B_i^n(st) = \sum_{j=0}^{n} B_i^j(s) B_j^n(t)$ and we are almost there except for the $b_k$ and the range of the inner summation. However, we also know that $B_i^j\left(\frac{t}{t} \right) = 0$ for all values $i < k$. This allows us to change the upper bound on the inner summation to match the $n$ of the outer summation. So we can say:

$$= \sum_{i=0}^{n} \sum_{k=0}^{n} B_i^j\left(\frac{t}{t} \right) b_k B_i^n\left(\frac{t}{t} \right)$$

Since the summations are over the same range now, we can swap the summations to get it more in the proper form. In fact, we can carry $b_k$ out of the summation since it does not depend on $i$. Then we get:

$$= \sum_{k=0}^{n} b_k \sum_{i=0}^{n} B_i^j\left(\frac{t}{t} \right) B_i^n\left(\frac{t}{t} \right)$$

Then by substituting the right hand side of our previous formula:

$$= \sum_{k=0}^{n} b_k B_i^n\left(\frac{t}{t} \right)$$

$$= \sum_{k=0}^{n} b_k B_i^n(t)$$

$$= B(b_0, b_1, \ldots, b_n : t) \quad \text{for } 0 \leq t \leq \tilde{t}$$
The same type of calculation applies to \( P_d(t) \)

\[
P_d(t) = B\left(d_0, d_1, \ldots, d_n : \frac{t-i}{1-t}\right)
\]

\[
= \sum_{i=0}^{n} d_i B^n_i\left(\frac{t-i}{1-t}\right)
\]

Now we can use the definition \( d_i = b_{n-i}^n(t) \)

\[
= \sum_{i=0}^{n} b_{n-i}^n(t) B^n_i\left(\frac{t-i}{1-t}\right)
\]

\[
= \sum_{i=0}^{n} \sum_{k=0}^{n-i} b_{n-k}^n(t) B^n_i\left(\frac{t-i}{1-t}\right)
\]

At this point we have too many variables: \( i \) and \( k \) on \( b \). This caused problems later when trying to use the subdivision formula in a previous attempt to solve this problem because \( b \) cannot just be moved outside of the summation due to its dependence on \( k \). The way around this is to introduce a new variable \( j = i + k \).

Now we can recast the above formula with the new \( j \) variable.

\[
= \sum_{i=0}^{n} \sum_{j=0}^{n-i} b_j B_{j-i}^n\left(\frac{t}{1-t}\right) B^n_i\left(\frac{t-i}{1-t}\right)
\]

But now we need to adjust the bound of the inner summation down to 0. However, \( B_{j-i}^n(t) = 0 \) if \( j < i \) so:

\[
= \sum_{i=0}^{n} \sum_{j=0}^{i} b_j B_{j-i}^n\left(\frac{t}{1-t}\right) B^n_i\left(\frac{t-i}{1-t}\right)
\]

Then we can use the relation \( B_{i,n}(u) = B_i^n(u) = B_{n-i}^n(1-u) \) to get the indices right for the outer \( B \).

\[
= \sum_{i=0}^{n} \sum_{j=0}^{i} b_j B_{j-i}^n\left(\frac{t}{1-t}\right) B^n_{n-i}\left(1 - \frac{t-i}{1-t}\right)
\]

However, this form causes problems when applying the subdivision formula because

\[
\left(\frac{t}{1-t}\right)\left(1 - \frac{t-i}{1-t}\right) = \frac{t(t-1)}{t-1}
\]

which does not help up. But if we also reverse the inner \( B \) we get:

\[
= \sum_{i=0}^{n} \sum_{j=0}^{i} b_j B_{n-j}^n\left(1-t\right) B^n_{n-i}\left(1 - \frac{t-i}{1-t}\right)
\]

Now we can swap the order of the summation and get \( b \) out of the summation (since we changed variables earlier).

\[
= \sum_{j=0}^{n} b_j \sum_{i=0}^{j} B_{n-j}^n\left(1-t\right) B^n_{n-i}\left(1 - \frac{t-i}{1-t}\right)
\]

Now the subdivision formula \( B_i^n(st) = \sum_{j=0}^{n} B_i^j(s) B_j^n(t) \) can be applied directly resulting in:

\[
= \sum_{j=0}^{n} b_j B_{n-j}^n\left(1-t\right)\left(1 - \frac{t-i}{1-t}\right)
\]

\[
= \sum_{j=0}^{n} b_j B_{n-j}^n\left(1-t\right)
\]
Apply $B_{i,n}(u) = B^n_i(u) = B^n_{n-i}(1-u)$

$$= \sum_{j=0}^{n} b_j B^n_j(t)$$

$$= B(b_0, b_1, ..., b_n : t) \quad \tilde{t} < t < 1$$

So now we have shown that both curves can be rephrased in terms of the original curve.

---

**THE WRONG WAY TO SOLVE IT** (for posterity reasons)

We can adjust the range of the inner summation because we know that $B^{n-i}_k(\tilde{t}) = 0$ if $n-i < k$

$$= \sum_{i=0}^{n} \sum_{k=0}^{n} b_{i+k} B^{n-i}_k(\tilde{t}) B^n_i \left( \frac{t - \tilde{t}}{1 - \tilde{t}} \right)$$

We still need to get to the form: $B^n_i(st) = \sum_{j=0}^{n} B^j_i(s) B^n_j(t)$ which means we should use the relation $B^i_n(st) = B^n_i(u) = B^n_{n-i}(1-u)$ to get the outermost B with the same subscripts. So we have:

$$= \sum_{i=0}^{n} \sum_{k=0}^{n} b_{i+k} B^{n-i}_k(\tilde{t}) B^n_i \left( \frac{1 - t - \tilde{t}}{1 - \tilde{t}} \right)$$

But unfortunately in this form, the parameters do not multiply into something that will get us a simpler expression. We must also reverse the inner B to get:

$$= \sum_{i=0}^{n} \sum_{k=0}^{n} b_{i+k} B^{n-i}_k(1 - \tilde{t}) B^n_{n-i} \left( \frac{1 - t - \tilde{t}}{1 - \tilde{t}} \right)$$

Now we are stuck because b uses both i and k and using a new variable changes the inner summation bounds. The variable must be introduced earlier and carried through the calculations.
4. We are given a rational Bezier curve as 
\[ P(t) = \frac{\sum_{i=0}^{n} b_i \frac{w_i B_i^n(t)}{\sum_{j=0}^{n} w_j B_j^n(t)}}{w_0}. \]

What happens to the curve at \( t=0 \) as the weight \( w_0 \to 0 \)?

Then we get a function like: 
\[ P(0) = \sum_{i=0}^{n} b_i \frac{w_i B_i^n(0)}{\sum_{j=0}^{n} w_j B_j^n(0)} \]

However, if \( i \) or \( j = 0 \) then \( B_i^n(0) = 1 \) and if \( i \) or \( j \neq 0 \) then they equal 0.

We can use this to simplify the denominator first to get:
\[ P(0) = \sum_{i=0}^{n} b_i \frac{w_i B_i^n(0)}{w_0 * 1} \]

Then applying that to the other summation we can further simplify to say:
\[ P(0) = b_0 \frac{w_0 * 1}{w_0 * 1} = b_0 \]

So \( w \) goes away! \( P(0) = b_0 \) it does not depend on the weight at all. Farin mentions in his paper, “Rational Curves and Surfaces” that the derivative is expressed as: 
\[ \dot{x}(0) = \frac{nw_i}{w_0} \Delta b_0 \]

which in our case means:
\[ P'(0) = \frac{nw_i}{w_0} (b_1 - b_0) \]

So this presents a problem in that as \( w_0 \to 0 \) then \( P'(0) \to \infty \).

Can you derive a lower degree Bezier representation of the curve if \( w_0 = 0 \)?

We start with the Bezier formulation: 
\[ P(t) = \sum_{i=0}^{n} b_i \frac{w_i B_i^n(t)}{\sum_{j=0}^{n} w_j B_j^n(t)} \]

And then take the lowest term out (the \( w_0 \) term) from both the b and the denominator:
\[ P(t) = b_0 \frac{w_0 B_0^n(t)}{\sum_{j=0}^{n} w_j B_j^n(t)} + \sum_{i=1}^{n} b_i \frac{w_i B_i^n(t)}{w_0 B_0^n(t) + \sum_{j=0}^{n} w_j B_j^n(t)} \]

But since \( w_0 = 0 \) we get:
\[ P(t) = \sum_{i=1}^{n} b_i \frac{w_i B_i^n(t)}{\sum_{j=1}^{n} w_j B_j^n(t)} \]
Now to derive how the new points relate to the old points, it helps to expand the definition of the basis functions as: 

\[ B^n_i(u) = \binom{n}{i} u^i (1-u)^{n-i} = \frac{n!}{(n-i)!i!} u^i (1-u)^{n-i} \]

Substituting the first part gives us:

\[
P(t) = \sum_{i=1}^{n} b_i \sum_{j=1}^{n} w_j \binom{n}{j} t^j (1-t)^{n-1-i}
\]

Substituting the second part results in:

\[
P(t) = \sum_{i=1}^{n} b_i \sum_{j=1}^{n} w_j \frac{n!}{(n-i)!j!} t^j (1-t)^{n-1-i}
\]

Now to keep the proper form, we must reindex the sum to start at 0 for both i and j.

\[
P(t) = \sum_{i=0}^{n-1} b_{i+1} \sum_{j=0}^{n-1} w_{j+1} \frac{n!}{(n-i-1)!(i+1)!} t^{i+1} (1-t)^{n-(i+1)}
\]

Finally we take a term out of the fractions in the denominator and numerator to provide coefficients which relate the old and new parameters.

\[
P(t) = \sum_{i=0}^{n-1} b_{i+1} \sum_{j=0}^{n-1} w_{j+1} \frac{n!}{i+1 (n-i-1)!j!} t^i (1-t)^{n-(i+1)}
\]

So this has the form of the Bezier with one degree less, degree n-1 and the \( b_i \) is now \( b_{i+1} \). So the control points \( b_i^{new} = b_{i+1} \) and from the coefficient terms taken out in the last step, the weights have also been changed. \( w_i^{new} = \frac{n}{i+1} w_{i+1} \). The range is \( 0 \leq i \leq n-1 \) as shown in the new sum ranges.

Now for an even more difficult derivation.
Similarly, a rational tensor product patch is defined as

\[ P(s, t) = \frac{\sum_{i=0}^{n} \sum_{j=0}^{m} b_{i,j} \frac{w_{i,j} B_i^n(s) B_j^m(t)}{\sum_{k=0}^{n} \sum_{l=0}^{m} w_{k,l} B_k^n(s) B_l^m(t)}}}{\sum_{k=0}^{n} \sum_{l=0}^{m} w_{k,l} B_k^n(s) B_l^m(t)} \]

What happens to the image of \((s, t) = (0, 0)\) as \(w_{0,0} \to 0\)?

This is similar to the previous part except with an extra set of parameters.

If \((s, t)\) goes to \((0, 0)\) then we get a function like:

\[ P(0,0) = \frac{\sum_{i=0}^{n} \sum_{j=0}^{m} b_{i,j} \frac{w_{i,j} B_i^n(0) B_j^m(0)}{\sum_{k=0}^{n} \sum_{l=0}^{m} w_{k,l} B_k^n(0) B_l^m(0)}}}{\sum_{k=0}^{n} \sum_{l=0}^{m} w_{k,l} B_k^n(0) B_l^m(0)} \]

However, if \(i \) or \(j = 0\) then \(B_i^n(0) = 1\) and if \(i \) or \(j \neq 0\) then they equal 0.

We can use this to simplify the denominator first to get:

\[ P(0,0) = \frac{\sum_{i=0}^{n} \sum_{j=0}^{m} b_{i,j} \frac{w_{i,j} B_i^n(0) B_j^m(0)}{w_{0,0} \ast 1 \ast 1}}{w_{0,0} \ast 1 \ast 1} = b_{0,0} \]

So \(w\) goes away! \(P(0) = b_0\) it does not depend on the weight at all. Farin mentions in his paper, “Rational Curves and Surfaces” that the derivative is expressed as:

\[ \dot{x}(0) = \frac{n w_1}{w_0} \Delta b_0 \]

which is similar to our case.

The velocity will be very high as \(w_{0,0} \to 0\).

****

What is the image of \(P(s, t)\) at \((s, t) = (0, 0)\), given \(w_{0,0} = 0\) ?

\[ P(s, t) = \frac{\sum_{i=0}^{n} \sum_{j=0}^{m} b_{i,j} \frac{w_{i,j} B_i^n(s) B_j^m(t)}{\sum_{k=0}^{n} \sum_{l=0}^{m} w_{k,l} B_k^n(s) B_l^m(t)}}}{\sum_{k=0}^{n} \sum_{l=0}^{m} w_{k,l} B_k^n(s) B_l^m(t)} \]

Now like last time we can substitute use the definition \(B_i^n(u) = \binom{n}{i} u^i (1-u)^{n-i} \) in the numerator and denominator (but not in the denominator quite yet).

\[ P(s, t) = \frac{\sum_{i=0}^{n} \sum_{j=0}^{m} b_{i,j} \frac{w_{i,j} \binom{n}{i} s^i (1-s)^{n-i} \binom{m}{j} t^j (1-t)^{m-j}}{\sum_{k=0}^{n} \sum_{l=0}^{m} w_{k,l} B_k^n(s) B_l^m(t)}}}{\sum_{k=0}^{n} \sum_{l=0}^{m} w_{k,l} B_k^n(s) B_l^m(t)} \]

Now taking out the first term of the summation we get:
As we approach the surface, we have a choice of direction. As in calculus, we approach from one direction
and see the result and then another. In this case, the first direction will be t=α.

Taking out one more term from the first summation:

\[
P(s,t) = \sum_{j=0}^{m} b_{0j} \frac{n}{0} (1-s)^n \alpha \left( m \right) j \left( 1-t \right)^{m-j} + \sum_{i=1}^{n} \sum_{j=0}^{m} b_{i,j} \frac{n}{i} \alpha \left( m \right) j \left( 1-t \right)^{m-j}
\]

Now taking off the first term of first summation:

\[
P(s,t) = b_{00} w_{00} \left( n \right) (1-s)^n \alpha \left( m \right) j \left( 1-t \right)^{m}
\]

Perhaps at this point it is easier to deal with the numerator and denominator separately.

Numerator(s,t) =

\[
\sum_{j=0}^{m} b_{0j} w_{0j} \left( n \right) (1-s)^n \alpha \left( m \right) j \left( 1-t \right)^{m-j} + \sum_{i=1}^{n} \sum_{j=0}^{m} b_{i,j} w_{i,j} \left( n \right) \alpha \left( m \right) j \left( 1-t \right)^{m-j}
\]

Denominator(s,t) =

\[
\sum_{k=0}^{n} \sum_{l=0}^{m} w_{k,l} B_k^n(s) B_l^m(t)
\]

But the first term disappears because \( w_{00} = 0 \).

Perhaps at this point it is easier to deal with the numerator and denominator separately.

Numerator(s,t) =

\[
\sum_{j=0}^{m} b_{0j} w_{0j} \left( n \right) (1-s)^n \alpha \left( m \right) j \left( 1-t \right)^{m-j} + \sum_{i=1}^{n} \sum_{j=0}^{m} b_{i,j} w_{i,j} \left( n \right) \alpha \left( m \right) j \left( 1-t \right)^{m-j}
\]

Denominator(s,t) =

\[
\sum_{k=0}^{n} \sum_{l=0}^{m} w_{k,l} B_k^n(s) B_l^m(t)
\]

As we approach the surface, we have a choice of direction. As in calculus, we approach from one direction
and see the result and then another. In this case, the first direction will be t=α. Substituting this for the numerator results in:

Numerator(s,αs) =

\[
\sum_{j=0}^{m} b_{0j} w_{0j} \left( n \right) (1-s)^n \alpha \left( m \right) j \left( 1-\alpha s \right)^{m-j} + \sum_{i=1}^{n} \sum_{j=0}^{m} b_{i,j} w_{i,j} \left( n \right) \alpha \left( m \right) j \left( 1-\alpha s \right)^{m-j}
\]

if we take an \( \alpha s \) and \( s \) out of the summation then the terms will result in 0 for the later summations.

Numerator(s,αs) =

\[
\alpha s \sum_{j=0}^{m} b_{0j} w_{0j} \left( n \right) (1-s)^n \alpha \left( m \right) j \left( 1-\alpha s \right)^{m-j} + \sum_{i=1}^{n} \sum_{j=0}^{m} b_{i,j} w_{i,j} \left( n \right) \alpha \left( m \right) j \left( 1-\alpha s \right)^{m-j}
\]

Now we can use \( s=0 \) on the sums (before applying it to the whole expression) and get:

Numerator(s,αs) =

\[
\alpha s b_{01} w_{01} \left( n \right) \left( m \right) + s b_{10} w_{10} \left( n \right) \left( m \right)
\]

with \( s \to 0 \)

If we expand the denominator as:

\[
D(s,t) = \sum_{k=0}^{n} \sum_{l=0}^{m} w_{k,l} B_k^n(s) B_l^m(t) = \sum_{k=0}^{n} \sum_{l=0}^{m} w_{k,l} \left( n \right) \alpha \left( m \right) k \left( l \right) \left( 1-t \right)^{m-l}
\]

Now like for the numerator we can get:

\[
D(s,t) = \sum_{l=0}^{m} w_{0l} \left( n \right) (1-s)^n \alpha \left( m \right) l \left( 1-t \right)^{m-l} + \sum_{k=1}^{n} \sum_{l=0}^{m} w_{k,l} \left( n \right) \alpha \left( m \right) k \left( l \right) \left( 1-t \right)^{m-l}
\]

Taking out one more term from the first summation:
\[ D(s, t) = w_{0,0} \binom{n}{0} s^0 (1-s)^m \binom{m}{0} t^0 (1-t)^m + \sum_{l=1}^{m} w_{0,l} \binom{n}{0} s^0 (1-s)^m \binom{m}{l} t^l (1-t)^{m-l} + \sum_{k=1}^{n} \sum_{l=0}^{m} w_{k,l} \binom{n}{k} s^k (1-s)^n-k \binom{m}{l} t^l (1-t)^{n-l} \]

Now noting that \( w_{0,0} = 0 \),

\[ D(s, t) = \sum_{l=1}^{m} w_{0,l} \binom{n}{0} s^0 (1-s)^m \binom{m}{l} t^l (1-t)^{m-l} + \sum_{k=1}^{n} \sum_{l=0}^{m} w_{k,l} \binom{n}{k} s^k (1-s)^n-k \binom{m}{l} t^l (1-t)^{n-l} \]

Now we can also substitute \( t = \alpha s \) in this case to get the denominator of the expression.

\[ D(s, \alpha s) = \sum_{l=1}^{m} w_{0,l} \binom{n}{0} s^0 (1-s)^m \binom{m}{l} (\alpha s)^l (1-\alpha s)^{m-l} + \sum_{k=1}^{n} \sum_{l=0}^{m} w_{k,l} \binom{n}{k} s^k (1-s)^n-k \binom{m}{l} (\alpha s)^l (1-\alpha s)^{n-l} \]

Now performing the same trick as in the numerator by taking out a multiplier and then putting \( s \to 0 \).

\[ D(s, \alpha s) = \alpha s w_{0,1} \binom{m}{1} + sw_{1,0} \binom{n}{1} \] where \( s \to 0 \)

Finally, the Numerator over the denominator is

\[ P(0,0) = \frac{\alpha s b_{0,1} w_{0,1} \binom{m}{1} + sb_{1,0} w_{1,0} \binom{n}{1}}{\alpha s w_{0,1} \binom{m}{1} + sw_{1,0} \binom{n}{1}} = \frac{\alpha b_{0,1} w_{0,1} m + b_{1,0} w_{1,0} n}{\alpha s w_{0,1} m + w_{1,0} n} \] with \( s \to 0 \) from s direction

BUT WE ARE NOT DONE YET, we must now try \( s = \alpha t \).

This time for the numerator

\[ N(\alpha t, t) = \sum_{j=1}^{m} b_{0,j} w_{0,j} \binom{n}{0} (1-\alpha t)^v \binom{m}{j} t^j (1-t)^{m-j} + \sum_{i=1}^{n} \sum_{j=0}^{m} b_{i,j} w_{i,j} \binom{n}{i} \alpha t^i (1-\alpha t)^{n-i} \binom{m}{j} t^j (1-t)^{n-j} \]

Now doing the same trick again of taking a \( t \) out while letting the summation’s \( t \to 0 \) we get:

\[ N(\alpha t, t) = \alpha t b_{0,1} w_{0,1} \binom{m}{1} + \alpha b_{1,0} w_{1,0} \binom{n}{1} \binom{m}{0} \]

Now for the denominator:

\[ D(\alpha t, t) = \sum_{l=1}^{m} w_{0,l} \binom{n}{0} (1-\alpha t)^v \binom{m}{l} t^l (1-t)^{m-l} + \sum_{k=1}^{n} \sum_{l=0}^{m} w_{k,l} \binom{n}{k} \alpha t^k (1-\alpha t)^{n-k} \binom{m}{l} t^l (1-t)^{n-l} \]

Same trick again

\[ D(\alpha t, t) = \alpha t \sum_{l=1}^{m} w_{0,l} \binom{n}{0} (1-\alpha t)^v \binom{m}{l} t^l (1-t)^{m-l} + \alpha \sum_{k=1}^{n} \sum_{l=0}^{m} w_{k,l} \binom{n}{k} \alpha t^{k-l} (1-\alpha t)^{n-k} \binom{m}{l} t^l (1-t)^{n-l} \]

We let the \( t \) term go to 0 in the summation and get:
\[ D(\alpha t, t) = tw_{0,1}\begin{pmatrix}n \\ 0\end{pmatrix}\begin{pmatrix}m \\ 1\end{pmatrix} + \alpha tw_{1,0}\begin{pmatrix}n \\ 1\end{pmatrix}\begin{pmatrix}m \\ 0\end{pmatrix} \]

Combining the numerator and denominator results in:

\[ P(0,0) = \frac{tb_{01}w_{01}\begin{pmatrix}n \\ 0\end{pmatrix}\begin{pmatrix}m \\ 1\end{pmatrix} + \alpha tb_{1,0}w_{1,0}\begin{pmatrix}n \\ 1\end{pmatrix}\begin{pmatrix}m \\ 0\end{pmatrix}}{tw_{0,1}\begin{pmatrix}n \\ 0\end{pmatrix}\begin{pmatrix}m \\ 1\end{pmatrix} + \alpha tw_{1,0}\begin{pmatrix}n \\ 1\end{pmatrix}\begin{pmatrix}m \\ 0\end{pmatrix}} = \frac{b_{01}w_{01}m + \alpha b_{1,0}w_{1,0}n}{w_{0,1}m + \alpha w_{1,0}n} \quad \text{for } t \to 0 \text{ from the } t \text{ direction} \]

Compare this to the s direction:

\[ P(0,0) = \frac{\alpha sb_{01}w_{01}\begin{pmatrix}m \\ 1\end{pmatrix} + sb_{1,0}w_{1,0}\begin{pmatrix}n \\ 1\end{pmatrix}}{\alpha sw_{0,1}\begin{pmatrix}m \\ 1\end{pmatrix} + sw_{1,0}\begin{pmatrix}n \\ 1\end{pmatrix}} = \frac{\alpha b_{01}w_{01}m + b_{1,0}w_{1,0}n}{\alpha w_{0,1}m + w_{1,0}n} \quad \text{for } s \to 0 \text{ from the } s \text{ direction} \]

So they are not the same. \(P(0,0)\) depends on the direction you approach the origin.
5. Define a piecewise rational curve \( p(t) \) as
\[
p(t) = \begin{cases} 
\left( \frac{x_1(t)}{w_1(t)}, \frac{y_1(t)}{w_1(t)} \right) & 0 \leq t \leq t_1 \\
\left( \frac{x_2(t)}{w_2(t)}, \frac{y_2(t)}{w_2(t)} \right) & t_1 \leq t \leq 1
\end{cases}
\]

Consider the polynomial curves in the higher dimensional space:
\[
\begin{align*}
P_1(t) &= (x_1(t), y_1(t), w_1(t)) & 0 \leq t \leq t_1 \\
P_2(t) &= (x_2(t), y_2(t), w_2(t)) & t_1 \leq t \leq 1
\end{align*}
\]

If the polynomial curves have \( C^k \) continuity at \( t = t_1 \), than the rational curve \( p(t) \) has \( C^k \) continuity at \( t = t_1 \) as well. However, this condition is sufficient and not necessary for the \( C^k \) continuity of \( p(t) \). Derive the necessary and sufficient conditions between the polynomials \( x_i(t) \), \( y_i(t) \), \( w_i(t) \) such that \( p(t) \) has \( C^k \) continuity at \( t = t_1 \).

Intuitively, the polynomials should be \( C^k \) differentiable and have matching derivatives at each level of differentiation in order to have \( C^k \) continuity. There must be equal tangent vectors in the limit. This should also apply to all the variables separately in the rational curve form because of the way the limit is defined (hold all variables constant except the one you are testing the continuity of). However in the polynomial form, the continuity depends on both \( x \) and \( w \) since the \( w \) will be divided to get the curve.

Following Farin’s discussion of rational curves, in his article “Rational Curves and Surfaces”, let’s start with:

Say that \( P_{x1} \) is the \( x \) part of polynomial 1 and \( P_{x2} \) is the \( x \) part of polynomial 2.

We know that \( P_{x1}(t) = \frac{x_1(t)}{w_1(t)} \) so \( P_{x1}(t)w_1(t) = x_1(t) \) and \( x_1^{(r)}(t) = \sum_{j=0}^{r} \binom{r}{j} w_1^{(j)}(t) P_{x1}^{(r-j)}(t) \)

Now we can take this further and try and relate all the derivatives simultaneously. We can create a matrix out of the values for \( w \) using the summation above. For a 3x3 case, we get:

\[
\begin{bmatrix}
x_1(t) \\
x_1'\, (t) \\
x_1''\, (t)
\end{bmatrix} = \begin{bmatrix}
w_1(t) & 0 & 0 \\
w_1'(t) & w_1(t) & 0 \\
w_1''\, (t) & 2w_1'(t) & w_1(t)
\end{bmatrix} \begin{bmatrix}
P_{x1}(t) \\
P_{x1}'(t) \\
P_{x1}''\, (t)
\end{bmatrix}
\]

And so on. The next row would be \( 1w''' \ 3w'' \ 3w' \ 1w \).

In general, \( w_{ij} = \begin{cases} 
\binom{j-1}{i-1} w_1^{(j-i)}(t) & \text{if } j > i \\
0 & \text{if } j = i \\
\end{cases} \) the minus 1 is because matrices are from (1,1) to (n,n)

Using this term for an entry in the matrix, we can create a \( n+1 \) by \( n+1 \) matrix called \( M \). Also we need vector \( \overline{x_1}(t) = (x_1(t), x_1'(t), x_1''(t)...) \) and \( \overline{P_x}(t) = (P_{x1}(t), P_{x1}'(t), P_{x1}''(t)...P_{x1}^{(n)}(t)) \) and write succinctly that \( M_1(t) \overline{P_x}(t) = \overline{x_1}(t) \). We can also say for the 2nd curve \( M_2(t) \overline{P_x}(t) = \overline{x_2}(t) \).

Finally the condition for continuity can be given as:
\[
\overline{P_{x1}}(t_1) = \overline{P_{x2}}(t_1) \Rightarrow M_1(t_1)^{-1} \overline{x_1}(t_1) = M_2(t_1)^{-1} \overline{x_2}(t_1)
\]

Similarly for \( y \):
\[
\overline{P_{y1}}(t_1) = \overline{P_{y2}}(t_1) \Rightarrow M_1(t_1)^{-1} \overline{y_1}(t_1) = M_2(t_1)^{-1} \overline{y_2}(t_1)
\]

Or simultaneously:
\[
\begin{bmatrix}
M_1(t_1)^{-1} \overline{x_2}(t_1) & -\overline{y_1}(t_1)
\end{bmatrix} = \begin{bmatrix}
M_2(t_1)^{-1} \overline{x_1}(t_1) & \overline{y_2}(t_1)
\end{bmatrix}
\]