B-splines

So, it is useful to recall what we are seeking. We desire a parametric curve representation of the form:

$$\vec{C}(\xi) = \sum_{i=1}^{n} f_i(\xi)\,\vec{P}_i$$

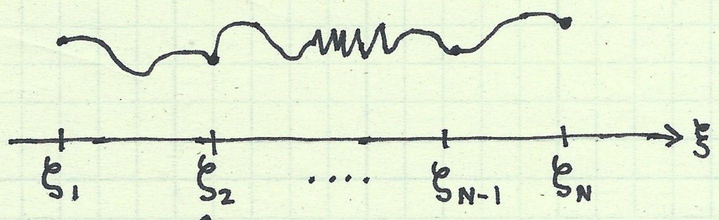
where:

ξ = parameter lying in $\mathbb{R}$

f_i = basis function

$\vec{P}_i$ = control point

Ideally, the basis functions $\{f_i\}_{i=1}^{n}$ should comprise a basis for the vector space of all piecewise polynomials of the desired degree and continuity with respect to a given mesh:



Continuity across mesh nodes is specified (e.g., C^0, C^1, C^2 continuity).

As continuity is determined by the basis itself, control points may be modified without altering the curve's continuity in contrast with composite Bézier curves.

In addition, we would like the curve to satisfy the same beneficial properties as a standard Bézier curve, such as the convex hull property and the variation diminishing property. Moreover, we would like to be able to locally control the curve. That is, we desire the following property: if a control point is moved, the resulting curve is modified only locally. These properties suggest that our basis functions are non-negative, locally supported, and form a partition of unity:

$$\sum_{i=1}^{n} f_i(\xi) \equiv 1$$

B-spline basis functions exhibit all of the aforementioned properties and more, and in fact they comprise the basis with minimal support with respect to a given degree, smoothness, and mesh partition that also is non-negative and forms a partition of unity.

B-splines were first introduced by mathematicians in the 19^{th} century, but the word "spline" was not introduced until 1946 by Issac Jacob Schoenberg. Schoenberg noticed that a twice-differentiable cubic spline approximates the shape of a draftsman's spline, hence the name. B-splines became particularly popular in 1970s when they found widespread use in the emerging field of computer aided geometric design.

The basic building blocks of a B-spline are as follows:

p = the polynomial degree
n = the number of basis functions
Ξ = the knot vector

The knot vector encodes both mesh and continuity information, and it takes the form:

$$\Xi := \{ \xi_1, \xi_2, \ldots, \xi_{n+p+1} \}$$

where $\xi_i \in \mathbb{R}$ is referred to as a knot and we require $\xi_i \leq \xi_{i+1}$. As we will later see, continuity is a function of knot repetition.

B-spline basis functions may be constructed in a number of ways, including divided differences, truncated formulas, blossoming, and recursion. We use the Cox-de Boor recursion formula as it is the most useful for computer implementation. We first present the formula for piecewise constants:

$$p = 0: \quad N_{i,0}(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi < \xi_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

and bootstrap recursively to higher polynomial degrees:

$$p \geq 0: \quad N_{i,p}(\xi) = \left(\frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} \right) N_{i,p-1}(\xi) + \left(\frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} \right) N_{i+1,p-1}(\xi)$$

where the fractions above are taken to be zero if the denominator is zero.

It is useful to first visualize B-spline basis functions in the setting of a uniform knot vector, that is, a knot vector with equispaced knots. Let us define:

$$\Xi := \{ 0, 1, 2, 3 \}$$

With respect to this knot vector, we have three B-spline basis functions of degree 0:

$$N_{1,0}(\xi) = \begin{cases} 1 & \text{if } 0 \leq \xi < 1 \\ 0 & \text{otherwise} \end{cases}$$

$$N_{2,0}(\xi) = \begin{cases} 1 & \text{if } 1 \leq \xi < 2 \\ 0 & \text{otherwise} \end{cases}$$

$$N_{3,0}(\xi) = \begin{cases} 1 & \text{if } 2 \leq \xi < 3 \\ 0 & \text{otherwise} \end{cases}$$

and from these $N_{i,0}$, we build two B-spline basis functions of degree 1:

$$N_{1,1}(\xi) = \left(\frac{\xi-\xi_1}{\xi_2-\xi_1}\right) N_{1,0}(\xi) + \left(\frac{\xi_3-\xi}{\xi_3-\xi_2}\right) N_{2,0}(\xi)$$

$$= \xi \, N_{1,0}(\xi) + (2-\xi)\, N_{2,0}(\xi)$$

$$= \begin{cases} \xi & \text{if } 0 \le \xi < 1 \\ (2-\xi) & \text{if } 1 \le \xi < 2 \\ 0 & \text{otherwise} \end{cases}$$

$$N_{2,1}(\xi) = \left(\frac{\xi-\xi_2}{\xi_3-\xi_2}\right) N_{2,0}(\xi) + \left(\frac{\xi_4-\xi}{\xi_4-\xi_3}\right) N_{3,0}(\xi)$$

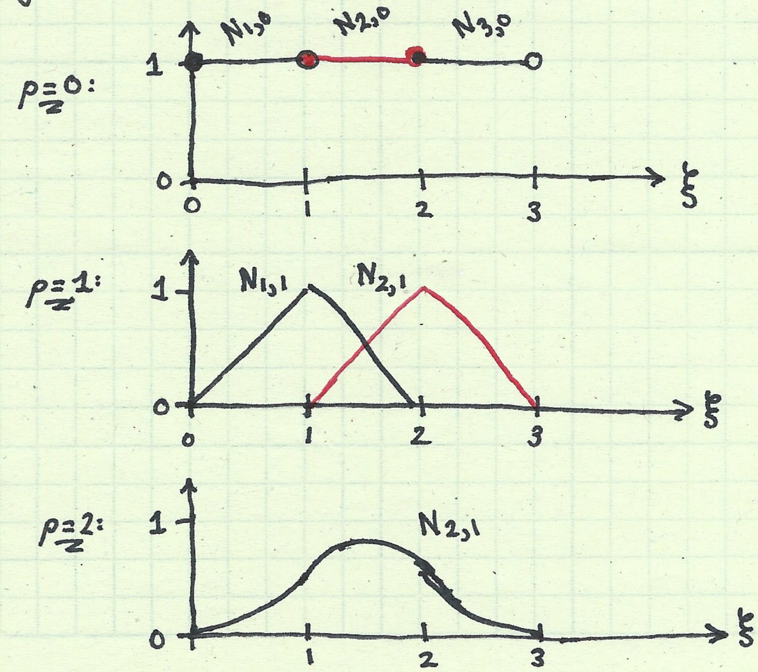
$$= \begin{cases} (\xi-1) & \text{if } 1 \le \xi < 2 \\ (3-\xi) & \text{if } 2 \le \xi < 3 \\ 0 & \text{otherwise} \end{cases}$$

Finally, from the linear B-spline basis functions, we build a quadratic B-spline basis function:

$$N_{1,2}(\xi) = \left(\frac{\xi-\xi_1}{\xi_3-\xi_1}\right) N_{1,1}(\xi) + \left(\frac{\xi_4-\xi}{\xi_4-\xi_2}\right) N_{2,1}(\xi)$$

$$= \begin{cases} \frac{1}{2}\xi^2 & \text{if } 0 \le \xi < 1 \\ \frac{1}{2}\xi(2-\xi) + \frac{1}{2}(3-\xi)(\xi-1) & \text{if } 1 \le \xi < 2 \\ \frac{1}{2}(3-\xi)^2 & \text{if } 2 \le \xi < 3 \\ 0 & \text{otherwise} \end{cases}$$

Visually:



We make the following observations:

- The basis functions are identical for each degree p but shifted relative to one another.
- The linear basis functions are standard linear finite elements.
- The quadratic basis function is <u>not</u> a standard quadratic finite element. It is C^1-continuous and has support over three knot spans or elements.
- The basis functions are pointwise non-negative and comprise a partition of unity for all $\xi \in [0,3)$.

We now visualize B-spline basis functions for a non-<u>uni</u>form knot vector:

$$\Xi := \{0, 0, 0, 1, 2, 2, 2\}$$

The $p = 0$ basis functions for this knot vector are then:

$$N_{1,0} \equiv 0$$
$$N_{2,0} \equiv 0$$

$$N_{3,0} = \begin{cases} 1 & \text{if } \xi \in [0,1) \\ 0 & \text{otherwise} \end{cases}$$

$$N_{4,0} = \begin{cases} 1 & \text{if } \xi \in [1,2) \\ 0 & \text{otherwise} \end{cases}$$

$$N_{5,0} \equiv 0$$
$$N_{6,0} \equiv 0$$

If we bootstrap to $p=1$, we find:

$$N_{1,1} \equiv 0$$

$$N_{2,1} = \begin{cases} 1 - \xi & \text{if } \xi \in [0,1) \\ 0 & \text{otherwise} \end{cases}$$

$$N_{3,1} = \begin{cases} \xi & \text{if } \xi \in [0,1) \\ 2 - \xi & \text{if } \xi \in [1,2) \\ 0 & \text{otherwise} \end{cases}$$

$$N_{4,1} = \begin{cases} \xi - 1 & \text{if } \xi \in [1,2) \\ 0 & \text{otherwise} \end{cases}$$

$$N_{5,1} \equiv 0$$

Finally, bootstrapping to $p = 2$, we find:

$$N_{1,2} = \begin{cases} (1-\xi)^2 & \text{if } \xi \in [0,1) \\ 0 & \text{otherwise} \end{cases}$$

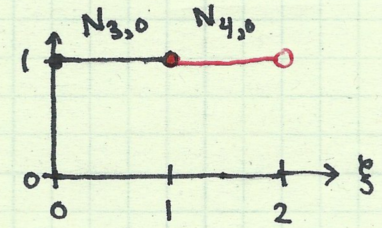
$$N_{2,2} = \begin{cases} \text{[redacted]}\ \xi(1-\xi) + \frac{1}{2}(2-\xi)\xi & \text{if } \xi \in [0,1) \\ \frac{1}{2}(2-\xi)^2 & \text{if } \xi \in [1,2) \\ 0 & \text{otherwise} \end{cases}$$

$$N_{3,2} = \begin{cases} \frac{1}{2}\xi^2 & \text{if } \xi \in [0,1) \\ \frac{1}{2}\xi(2-\xi) + (2-\xi)(\xi-1) & \text{if } \xi \in [1,2) \\ 0 & \text{otherwise} \end{cases}$$

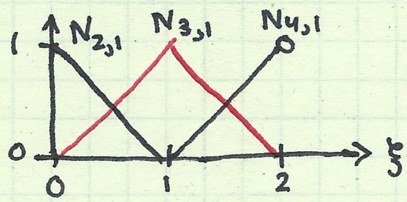
$$N_{4,2} = \begin{cases} (\xi-1)^2 & \text{if } \xi \in [1,2) \\ 0 & \text{otherwise} \end{cases}$$

Visually:

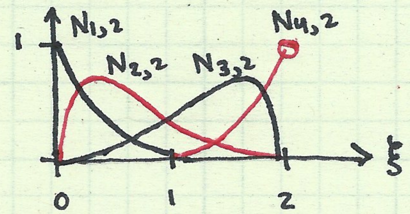
$\underline{p=0}$:



$\underline{p=1}$:



$\underline{p=2}$:



We make the following observations:

- With the exception of $p=0$, the basis functions are no longer homogeneous.
- The basis functions are interpolatory $\underline{\text{and}}$ discontinuous at $\xi=0$ and $\xi=2$.

This last property is a result of knot repetition. Namely, the knots at $\xi=0$ and $\xi=2$ were repeated 3 times, resulting in a discontinuity for polynomial degrees $p=0,1,2$.

From the Cox-de Boor formula, we may show that the B-spline basis exhibits the following properties:

- <u>Positivity</u>: $N_{i,p}(\xi) \geq 0$
- <u>Local Support</u>: $N_{i,p}(\xi)$ is nonzero only over the interval $[\xi_i, \xi_{i+p+1})$.

- "Flipside" of Local Support: On any knot span $[\xi_i, \xi_{i+1})$, only $p+1$ degree p basis functions are nonzero:
$$N_{i-p,p}, \ldots, N_{i,p}$$

- Each basis function is a composite Bézier curve.

- At a knot of multiplicity k (i.e., $\xi_{i+1} = \cdots = \xi_{i+k}$), basis functions are C^{p-k}-continuous.

- At a knot of multiplicity p, $\xi = \xi_{i+1} = \cdots = \xi_{i+p}$, $N_{j,p}(\xi) = \delta_{i,j}$.

- At a knot of multiplicity $p+1$, $\xi = \xi_i = \cdots = \xi_{i+p}$, $N_{j,p}(\xi) = \delta_{i,j}$.

- At a knot of multiplicity $p+1$, $\xi = \xi_i = \cdots = \xi_{i+p}$, $N_{i-1,p}(\xi^-) = 1$.

Limit taken from left.

- Partition of Unity: $\displaystyle\sum_{i=1}^{n} N_{i,p}(\xi) = 1$ for $\xi \in [\xi_1, \xi_{n+p+1})$

- Derivative as Recursion:
$$\frac{d}{d\xi} N_{i,p}(\xi) = \frac{p}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) - \frac{p}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi)$$

These properties will be very useful in constructing B-spline curves and also later in the context of finite element analysis.

Throughout, we will be solely interested in B-splines associated with open knot vectors. A knot vector is open if its first and last knots appear $p+1$ times. Then, from above, we have:

Endpoint Interpolation: $N_{i,p}(\xi_1) = \delta_{i,1}$ $\quad N_{i,p}(\xi^-_{n+p+1}) = \delta_{i,n}$

Note that the Bernstein basis is obtained from an open knot vector with no interior knots!

We are now in a position to define a B-spline curve. Given a set of n control points, a B-spline curve is:

$$\vec{C}(\xi) = \sum_{i=1}^{n} N_{i,p}(\xi) \vec{P}_i$$

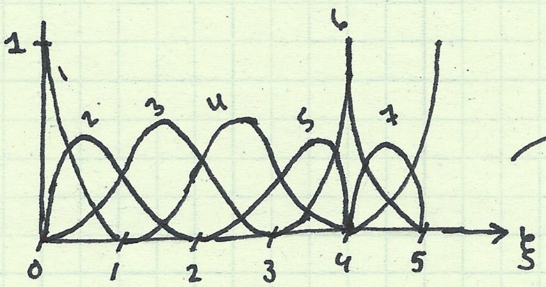
B-spline Basis Functions

Control Points

An app (interactive) demonstrating the relationship of B-spline basis functions and curves is available at:

research.engr.utexas.edu/cagd/B-Spline-Interaction

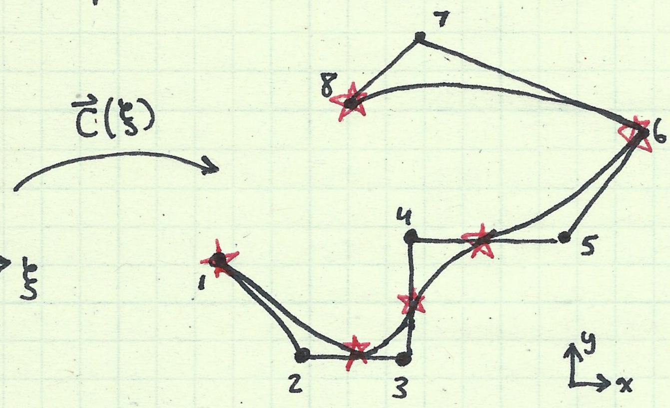
An example of a piecewise quadratic B-spline curve in 2D is illustrated below:



Basis Functions

$p = 2$

$\Xi = \{0, 0, 0, 1, 2, 3, 4, 4, 5, 5, 5\}$



B-spline Curve and Control Mesh

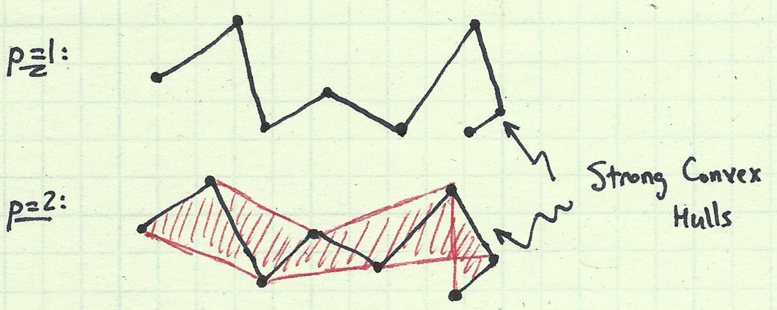
★ = Images of Knots

Note that the curve is interpolatory at the first and last control points, a general feature of a curve built from an open knot vector. It is also interpolatory at the sixth control point. This is because the knot at $\xi = 4$ is repeated twice. Hence, the curve reflects the basis. The curve is also tangent to the first and last segments at the beginning and end of the curve of the control polygon and also to the control polygon at the sixth control point, which is equivalently the image of knot $\xi = 4$.

Above, we have also portrayed the image of the knots under the B-spline mapping. These points naturally partition the curve into elements we will later use in finite element analysis.

B-spline curves have several nice features, much like Bézier curves. These are listed below:

- Convex Hull Property: This is an immediate consequence of the positivity and partition of unity properties of the B-spline basis.
- Variation Diminishing Property
- Strong Convex Hull Property: The B-spline curve lies within the union of all convex hulls formed by $p+1$ successive control points.



This is a stronger version of the standard convex hull property.

- <u>Local Control Property</u>: Every curve segment $\vec{C}([\xi_i, \xi_{i+1}))$ is only influenced by $p+1$ control points:

$$\vec{P}_{i-p}, \ldots, \vec{P}_i$$

- Affine <u>Covariance</u>: An affine transformation of the B-spline curve may be obtained by applying the transformation to the control points. This includes translations, rotations, scalings, and uniform stretchings and shearings.

Moreover, if a B-spline curve is built from an <u>open</u> knot vector, it satisfies the end<u>point</u> interpo<u>lation</u> and tan<u>gency</u> properties.

We conclude this section by discussing evaluation of a B-spline curve. Repeated substitution of the recursive definition of the B-spline basis function and re-indexing leads to the following:

$$\vec{C}(\xi) = \sum_{i=1}^{n+k-1} \vec{P}_{k,i} \, N_{i,p+1-k}(\xi) \qquad k = 1, \ldots, p+1$$

where:

$$\vec{P}_{k,i} = (1-\alpha_{k,i}) \vec{P}_{k-1,i-1} + \alpha_{k,i} \vec{P}_{k-1,i}$$

$$k = 2, \ldots, p+1$$
$$i = 1, \ldots, n+k-1$$

and

$$\vec{P}_{1,i} = \vec{P}_i \qquad i = 1, \ldots, n$$

with:

$$\alpha_{k,i} = \frac{\xi - \xi_i}{\xi_{i+p+2-k} - \xi_i}$$

Now, we make a crucial observation. Supposing that $\xi \in [\xi_l, \xi_{l+1})$, only basis functions $N_{l-p+(k-1),\, p-(k-1)}, \ldots, N_{l,\, p-(k-1)}$ are nonzero at ξ where $k = 1, \ldots, p+1$. Thus:

$$\vec{C}(\xi) = \sum_{i=l-p+(k-1)}^{l} \vec{P}_{k,i} \, N_{i,p+1-k}(\xi) \qquad k = 1, \ldots, p+1$$

and we compute the necessary control points via the triangular scheme:

$$\begin{tikzcd}
 & \vec{P}_{p+1,l} \arrow[dl] \arrow[d] & \vec{C}(\xi) \arrow[l, squiggly] \\
\vec{P}_{p,l-1} \arrow[d] \arrow[dl] & \vec{P}_{p,l} \arrow[d] \arrow[dl] & \\
\cdots\cdots & \cdots\cdots &
\end{tikzcd}$$

This greatly reduces computational complexity and simultaneously allows us to avoid division by zero in computing the alpha coefficients. This inspires the following pseudocode:

```
function   Spline Curve (ξ)
begin
           find l s.t. ξ ∈ [ξ_l, ξ_{l+1}) via binary search

           return deBoor (p+1, l)
end
```

```
function   deBoor (k, i, ξ)
begin
           if k=1 then
                return P_i
           else
                compute α_{k,i} = (ξ - ξ_i)/(ξ_{i+p+2-k} - ξ_i)

                return (1 - α_{k,i})* deBoor (k-1, i-1, ξ)
                       + α_{k,i} * deBoor (k-1, i, ξ)
end
```

Like, de Casteljau, the above algorithm is not efficient for large p as it has $O(p^2)$ complexity, but it is very stable as it involves convex combinations.