# Derivatives of B-spline curves

Let $\boldsymbol{C}^{(k)}(u)$ denote the $k$-th derivative of $\boldsymbol{C}(u)$. If $u$ is fixed, we can obtain $\boldsymbol{C}^{(k)}(u)$ by computing the $k$-th derivative of the basis functions,

$$\boldsymbol{C}^{(k)}(u) = \sum_{i=0}^{n} N_{i,p}^{(k)}(u)\,\boldsymbol{P}_i$$

An algorithm to compute the point on a B-spline curve and all derivatives up to and including the $d$-th, at a fixed $u$-value follows.

We allow $d > p$, although the derivatives are zero in this case (for nonrational curves); these derivatives are necessary for rational curves.

Input to the algorithm is $u$, $d$, and the B-spline curve, defined (throughout the course) by:

- $n$: the number of control points is $n + 1$
- $p$: the degree of the curve
- $U$: the knots
- $P$: the control points

Output is the array, **`CK[]`**, where **`CK[]`** is the $k$-th derivative, $0 \leq k \leq d$. A local array, **`nders[][]`**, is used to store the derivatives of the basis functions.

See algorithm A3.2

Now, instead of fixing $u$, we want to formally differentiate the $p$-th degree B-spline curve defined on the knot vector,

$$U = \{\underbrace{0, \ldots, 0}_{p+1}, u_{p+1}, \ldots u_{m-p-1}, \underbrace{1, \ldots, 1}_{p+1}\}$$

Thus,

$$\boldsymbol{C}'(u) = \sum_{i=0}^{n} N'_{i,p}(u)\boldsymbol{P}_i$$

$$= \sum_{i=0}^{n} \left( \frac{p}{u_{i+p} - u_i} N_{i,p-1}(u) - \frac{p}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \right) \boldsymbol{P}_i$$

$$\boldsymbol{C}'(u) =$$

$$\left(p \sum_{i=-1}^{n-1} N_{i+1,p-1}(u) \frac{\boldsymbol{P}_{i+1}}{u_{i+p+1} - u_{i+1}}\right)$$

$$-\left(p \sum_{i=0}^{n} N_{i+1,p-1}(u) \frac{\boldsymbol{P}_{i}}{u_{i+p+1} - u_{i+1}}\right)$$

or,

$$\boldsymbol{C}'(u) = p\frac{N_{0,p-1}(u)\boldsymbol{P}_0}{u_p - u_0}$$

$$+ p\sum_{i=0}^{n-1} N_{i+1,p-1}(u)\frac{(\boldsymbol{P}_{i+1} - \boldsymbol{P}_i)}{u_{i+p+1} - u_{i+1}}$$

$$-p\frac{N_{n+1,p-1}(u)\boldsymbol{P}_n}{u_{n+p+1} - u_{n+1}}$$

Note that the first and last terms evaluate to 0/0, which is 0 by definition. Thus:

$$\boldsymbol{C}'(u) = p\sum_{i=0}^{n-1} N_{i+1,p-1}(u)\frac{(\boldsymbol{P}_{i+1}-\boldsymbol{P}_i)}{u_{i+p+1}-u_{i+1}}$$

$$= \sum_{i=0}^{n-1} N_{i+1,p-1}(u)\,\boldsymbol{Q}_i$$

where,

$$\boldsymbol{Q}_i = p\frac{(\boldsymbol{P}_{i+1}-\boldsymbol{P}_i)}{u_{i+p+1}-u_{i+1}}$$

Now, let *U'* be the knot vector obtained by dropping the first and last knots from *U*, i.e.,

$$U' = \{\underbrace{0, \ldots, 0}_{p}, u_{p+1}, \ldots u_{m-p-1}, \underbrace{1, \ldots, 1}_{p}\}$$

(*U'* has *m* - 1 knots). Then it is easy to check that the function $N_{i+1,p-1}(u)$, computed on U, is equal to $N_{i,p-1}(u)$, computed on *U'*. Thus,

$$\boldsymbol{C}'(u) = \sum_{i=0}^{n-1} N_{i,p-1}(u)\,\boldsymbol{Q}_i$$

where the $N_{i,p\text{-}1}(u)$ are computed on $U'$. Hence, $\boldsymbol{C}'(u)$ is a ($p$ - 1)-th degree B-spline curve.

# Example:

Let $\boldsymbol{C}(u) = \sum_{i=0}^{4} N_{i,2}(u)\boldsymbol{P}_i$ , defined on

$U$ = {0, 0, 0, 2/5, 3/5, 1, 1, 1}.

Then, $U'$ = {0, 0, 2/5, 3/5, 1, 1}, and

$$\boldsymbol{C}'(u) = \sum_{i=0}^{3} N_{i,1}(u)\boldsymbol{Q}_i$$

where,

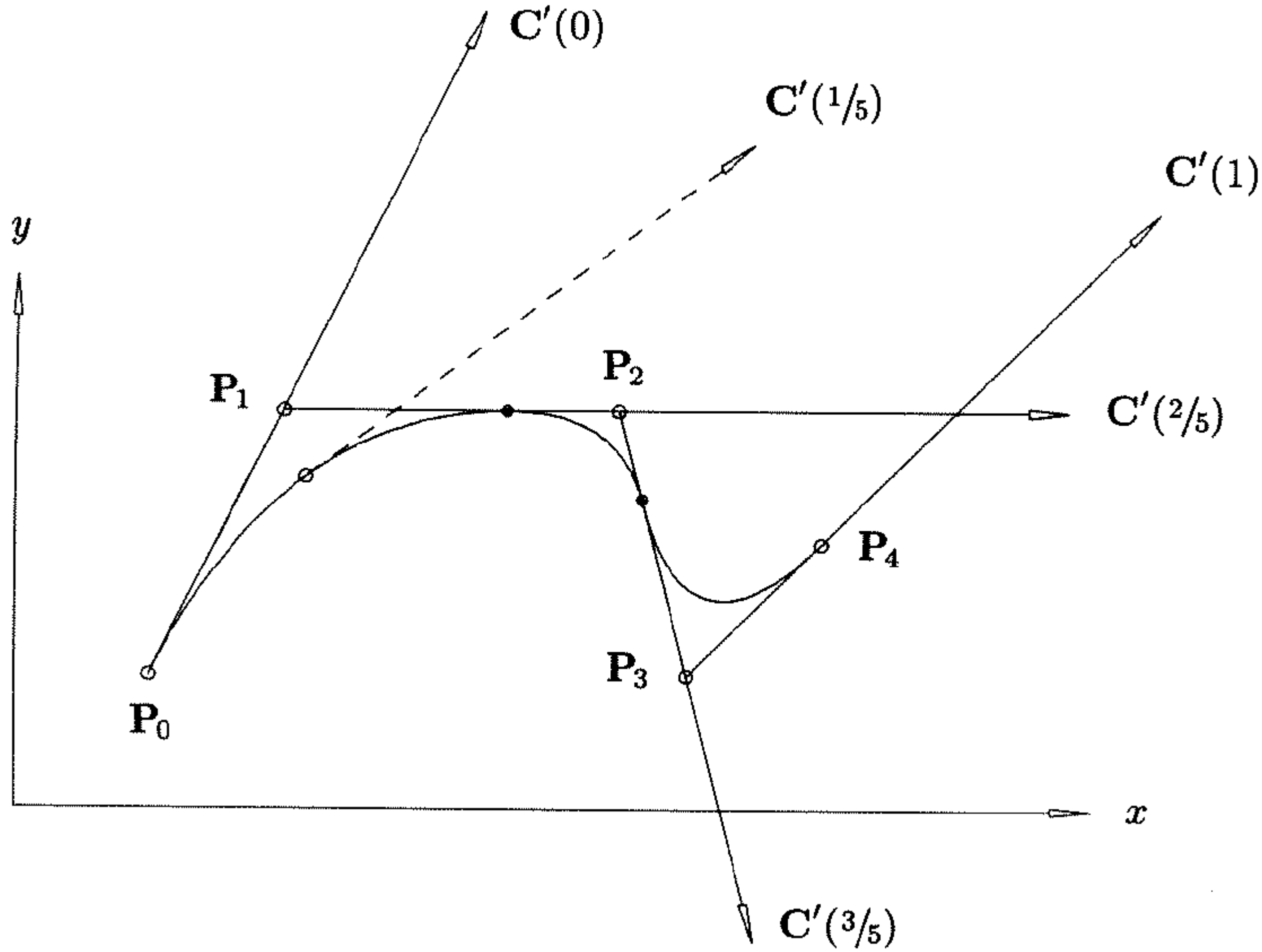
$$\boldsymbol{Q}_0 = \frac{2\,(\boldsymbol{P}_1 - \boldsymbol{P}_0)}{\frac{1}{3} - 0} = 6\,(\boldsymbol{P}_1 - \boldsymbol{P}_0)$$

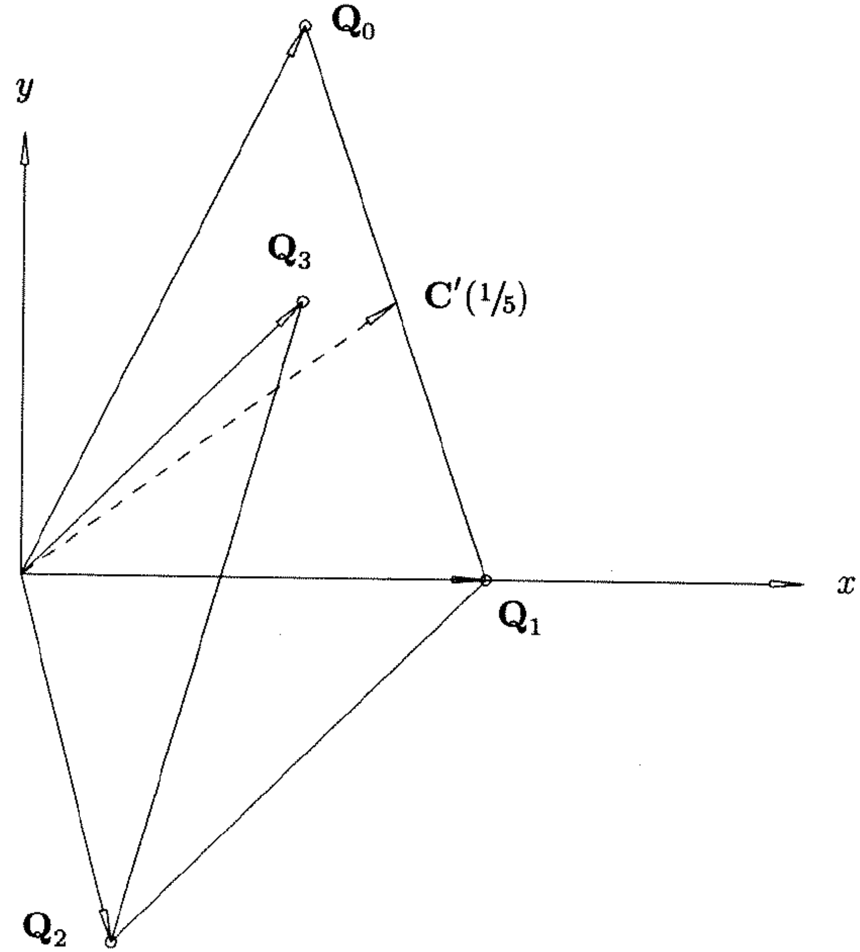
$$\boldsymbol{Q}_1 = \frac{2\,(\boldsymbol{P}_2 - \boldsymbol{P}_1)}{\frac{3}{4} - 0} = \frac{8}{3}\,(\boldsymbol{P}_2 - \boldsymbol{P}_1)$$

and,

$$\boldsymbol{Q}_2 = \frac{2(\boldsymbol{P}_3 - \boldsymbol{P}_2)}{1 - \frac{1}{3}} = 3(\boldsymbol{P}_3 - \boldsymbol{P}_2)$$

$$\boldsymbol{Q}_3 = \frac{2(\boldsymbol{P}_4 - \boldsymbol{P}_3)}{1 - \frac{3}{4}} = 8(\boldsymbol{P}_4 - \boldsymbol{P}_3)$$


C′(0)
C′(1/5)
C′(1)
y
P1
P2
C′(2/5)
P4
P3
P0
x
C′(3/5)



$\mathbf{Q}_0$
$y$
$\mathbf{Q}_3$
$\mathbf{C}'(1/5)$
$x$
$\mathbf{Q}_1$
$\mathbf{Q}_2$

Since $\boldsymbol{C}'(u)$ is a B-spline curve, we can apply this formulation recursively to obtain higher order derivative. Letting $\boldsymbol{P}_i^{(0)} = \boldsymbol{P}_i$, we write:

$$\boldsymbol{C}(u) = \boldsymbol{C}^{(0)}(u) = \sum_{i=0}^{n} N_{i,p}(u)\boldsymbol{P}_i^{(0)}$$

Then,

$$\boldsymbol{C}^{(k)}(u) = \sum_{i=0}^{n-k} N_{i,p-k}(u)\,\boldsymbol{P}_i^{(k)}$$

with,

$$\boldsymbol{P}_i^{(k)} = \begin{cases} \boldsymbol{P}_i\,, & if\ k = 0 \\ \dfrac{p-k+1}{u_{i+p+1}-u_{i+k}}\left(\boldsymbol{P}_{i+1}^{k-1}+\boldsymbol{P}_i^{k-1}\right), & if\ k>0 \end{cases}$$

and,

$$U^{(k)} = \{\underbrace{0, \ldots, 0}_{p-k+1}, u_{p+1}, \ldots u_{m-p-1}, \underbrace{1, \ldots, 1}_{p-k+1}\}$$

An algorithm based on this formulation computes the control points of all derivative curves up to and including the *d*-th derivative.

On output, **`PK[k][j]`** is the j-th control point of the *k*-th derivative curve, where $0 \le k \le d$ and $r_1 \le j \le r_2 - k$. If $r_1 = 0$ and $r_2 = n$, all control points are computed.

See algorithm A3.3

This formulation can be used to develop another algorithm to compute the point on a B-spline curve and all of its derivatives up to and including the *d*-th at a fixed *u*-value.

This algorithm is based on Algorithm A3.3, and assumes a simple modification of Algorithm A2.2 (**`BasisFuns`**) to return all nonzero basis functions of all degrees (from 0 up to *p*).

In particular, **`N[j][i]`** is the value of the *i*-th degree basis function $N_{span-i+j,i}(u)$, where $0 \le i \le p$ and $0 \le j \le i$

See algorithm A3.4