B-Splines

- Three useful references :

R. Bartels, J.C. Beatty, B. A. Barsky, An introduction to Splines for use in Computer Graphics and Geometric Modeling, Morgan Kaufmann Publications,1987

JC.Léon, Modélisation et construction de surfaces pour la CFAO, Hermes, 1991

L. Piegl, W. Tiller, The NURBS Book, Second Edition, Springer , 1996

B-Splines

- Isaac J. Schoenberg (1946)
- Carl De Boor (1972-76)
- Maurice G. Cox (1972)
- Richard Riesenfeld (1973)
- Wolfgang Boehm (1980)

B-Splines

- For Bézier curves, the polynomial degree is directly related to the number of control points.
  - The control of the continuity between Bézier curves is not trivial
- B-Splines are a generalization in the sense that the degree doesn't depend on the number of control points
  - One can impose every continuity at any point of the curve (we will see later how to do that)
  - They are polynomial curves, by pieces (Bézier curves have a unique polynomial representation along the interval of definition)
  - They may provide local control
  - The parametrization can be freely chosen (with Bézier, it is fixed , usually $0<u<1$.)

- Basis of Bézier curves :

$$P(u)=\sum_{i=0}^{d} P_i B_i^d(u)$$

  - The support of the basis functions is the interval $[0..1]$
  - Continuity is C_∞ , and between different Bézier curves it is enforced by a wise choice of the P_i's

- B-splines basis

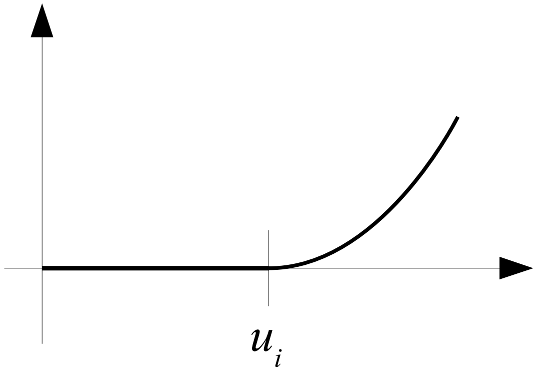
$$P(u)=\sum_{i=0}^{n} P_i N_i^d(u)$$

  - The basis functions N_i^d are piecewise polynomials
  - Have a *compact support* + satisfy partition of the unity
  - The continuity is defined at the basis function's level.

- B-spline basis functions
  - Defined by the nodal sequence and by the polynomials degree of the curve (d)
  - There are $n+1$ such functions, indexed from 0 to n .
- Nodal sequence:
  - It is a series of values u_i (knots) of the parameter u of the curve, not strictly increasing – there can be equal values.
  - There are $m+1$ such knots, indexed from 0 to m
  - e.g. $U=\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
    $U=\{0, 0, 0, 1, 2, 3, 4, 5, 5, 5\}$
    $U=\{0, 0, 0, 1, 2, 2, 3, 4, 4, 4\}$

- Construction of B-Spline basis functions
  - Truncated Power Function

$$(u-u_i)_+^d = \begin{cases} (u-u_i)^d & \text{if} \quad u \geq u_i \\ 0 & \text{otherwise} \end{cases}$$



  - It is a function of $C^{d\text{-}1}$ continuity

- Divided differences
  - order one (similar to a simple derivative)

$$[u_i, u_{i+1}]_U f(U) = \frac{f(u_{i+1}) - f(u_i)}{u_{i+1} - u_i}$$

U is a hidden parameter (like a variable used to differentiate)

  - order 2 : application of the above formula twice...

$$[u_i, u_{i+1}, u_{i+2}]_U f(U) = \frac{[u_{i+1}, u_{i+2}]_U f(U) - [u_i, u_{i+1}]_U f(U)}{u_{i+2} - u_i}$$

$$[u_i, u_{i+1}, u_{i+2}]_U f(U) = \frac{\dfrac{f(u_{i+2}) - f(u_{i+1})}{u_{i+2} - u_{i+1}} - \dfrac{f(u_{i+1}) - f(u_i)}{u_{i+1} - u_i}}{u_{i+2} - u_i}$$

B-Splines

- At the order k

$$[u_i,\cdots,u_{i+k}]f=\frac{[u_{i+1},\cdots,u_{i+k}]f-[u_i,\cdots,u_{i+k-1}]f}{u_{i+k}-u_i}$$

- One assumed that $u_i \neq u_{i+1} \neq u_{i+2} \cdots$

- Properties (see Bartels, 1987)

1- In the case where $u_i = u_{i+1} = u_{i+2} \cdots$

$$[u_i,\cdots,u_{i+k}]f=\frac{1}{k!}\left.\frac{d^k f}{du^k}\right|_{u=u_i}$$

2- if $u_i \neq u_{i+1} \neq u_{i+2} \cdots$ and $u_i < u_{i+1} < u_{i+2} \cdots$

$$[u_i,\cdots,u_{i+k}]f=\frac{1}{k!}\left.\frac{d^k f}{du^k}\right|_{u=u^*},\ u_i<u^*<u_{i+k}$$

3- $[u_i, \cdots, u_{i+k}]f$ is symmetric with respect to the knot vector

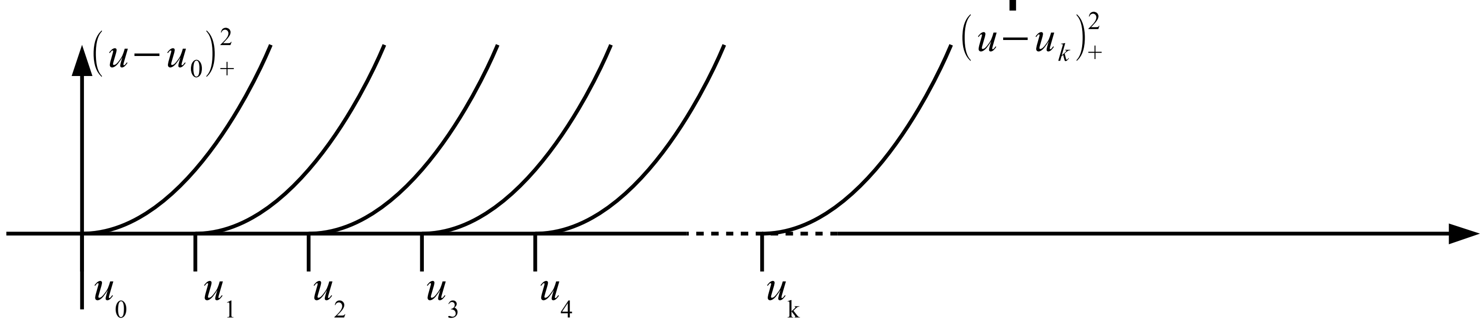
4- If $f(u)$ is a polynomial of degree at the most equal to k , then $[u_i, \cdots, u_{i+k}]f$

is a constant with respect to the u_i.

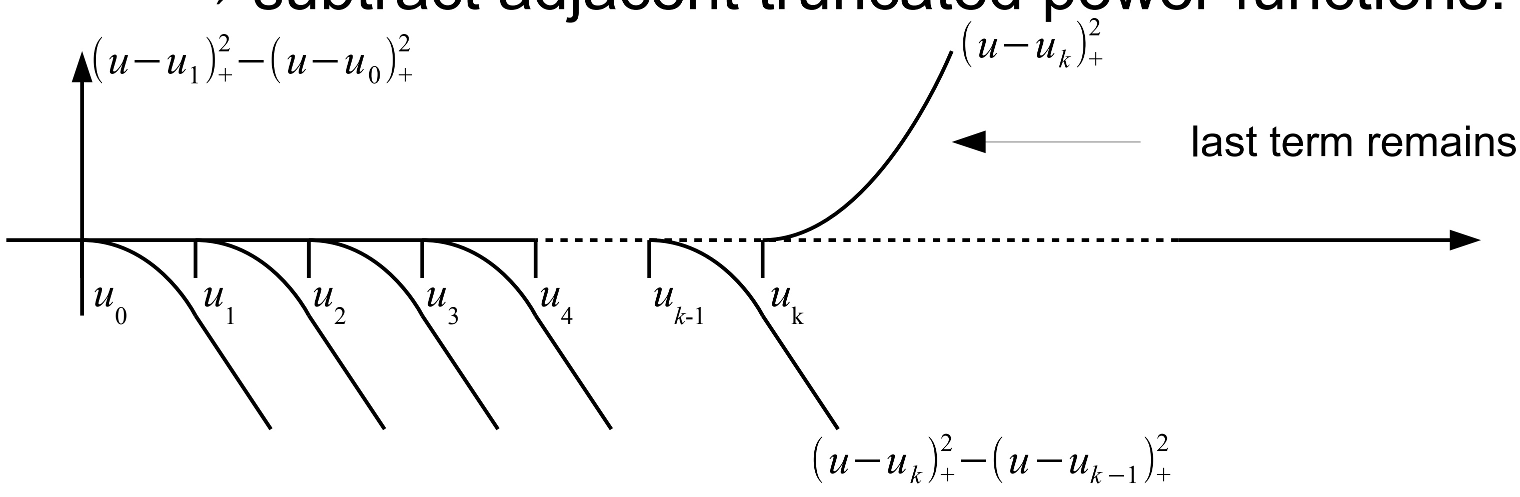
5- The divided difference of f=$g(u)$.$h(u)$ is :

$$[u_i, \cdots, u_{i+k}]f = \sum_{j=i}^{j=i+k} ([u_i, \cdots, u_j]g) \cdot ([u_j, \cdots, u_{i+k}]h)$$

- Divided differences and B-Splines



  - How to cancel quadratic terms ?
    → subtract adjacent truncated power functions.



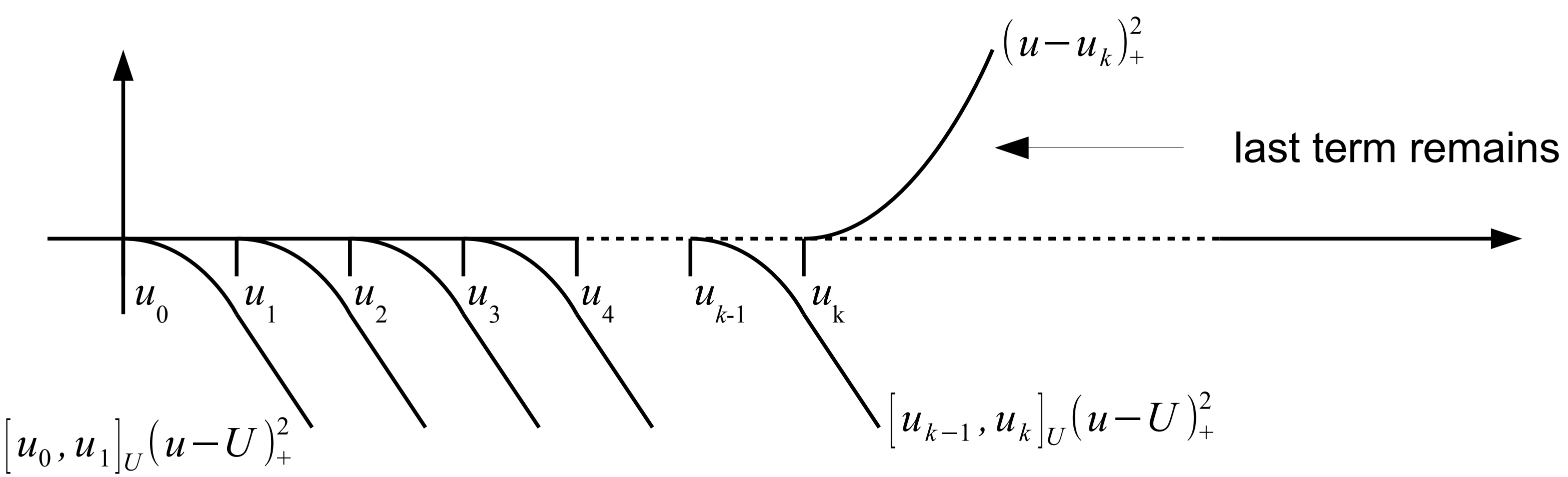
B-Splines

- Problem, lower order terms are dependent on k

$$\left.(u-u_k)_+^2-(u-u_{k-1})_+^2\right|_{u>u_k}=0\cdot u^2+(u_k-u_{k-1})\cdot u+(u_k-u_{k-1})(u_k+u_{k-1})\cdot 1$$

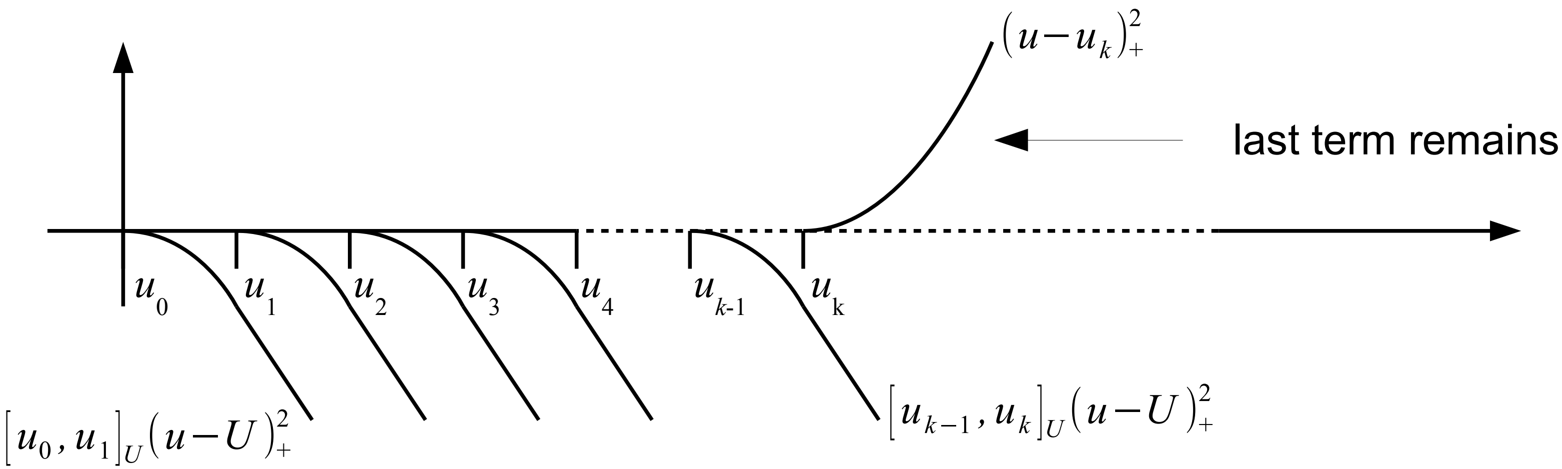
But, dividing by (u_k-u_{k-1}) yields a divided difference :

$$\frac{(u-u_k)_+^2-(u-u_{k-1})_+^2}{u_k-u_{k-1}}=[u_{k-1},u_k]_U(u-U)_+^2$$

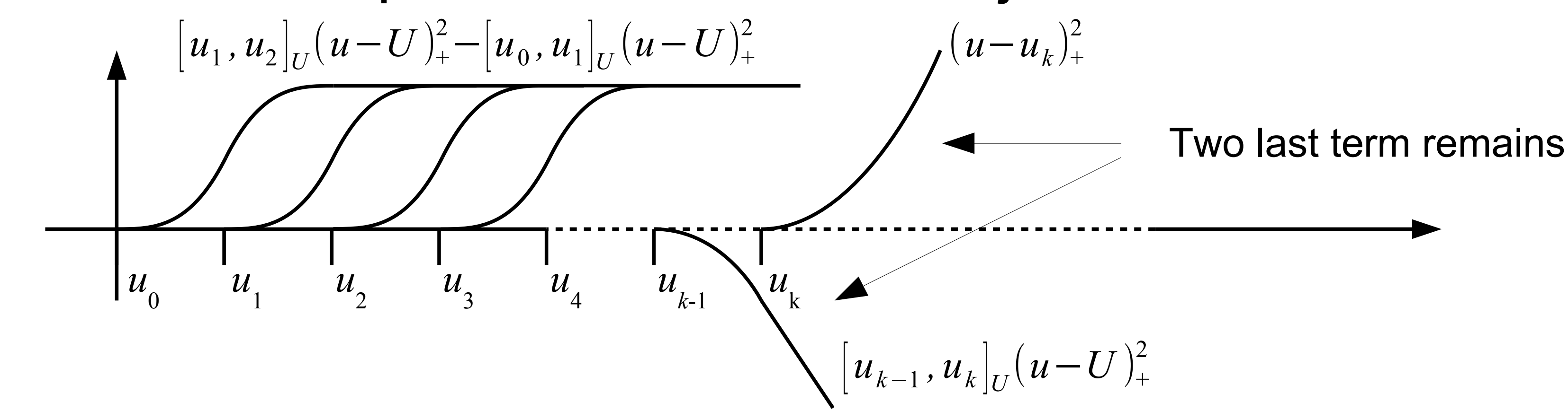


B-Splines

- Now, cancel linear terms …



- Same procedure : subtract adjacent terms.



Computer Aided Design

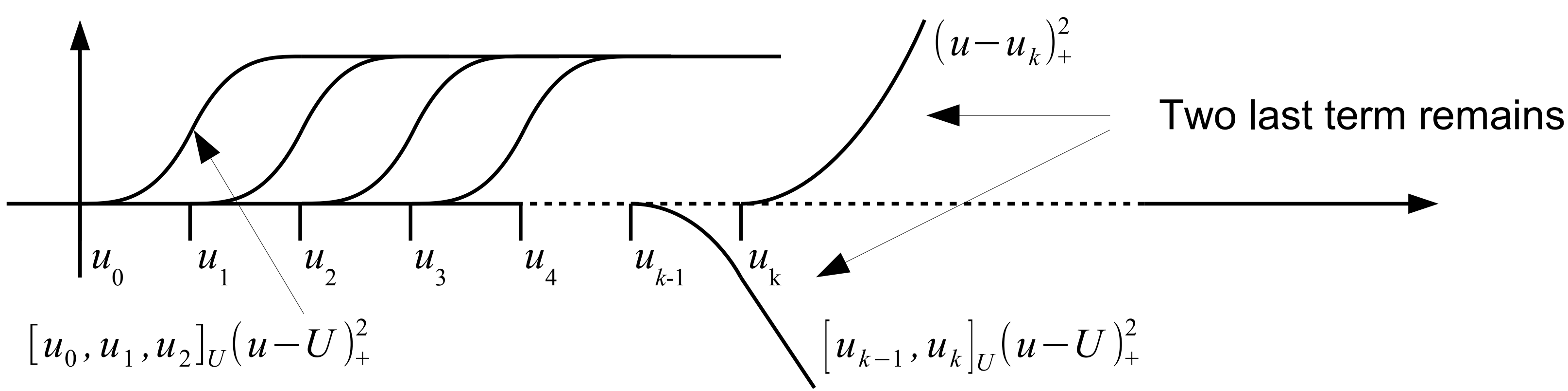
B-Splines

- Again, lower order terms are dependent on k

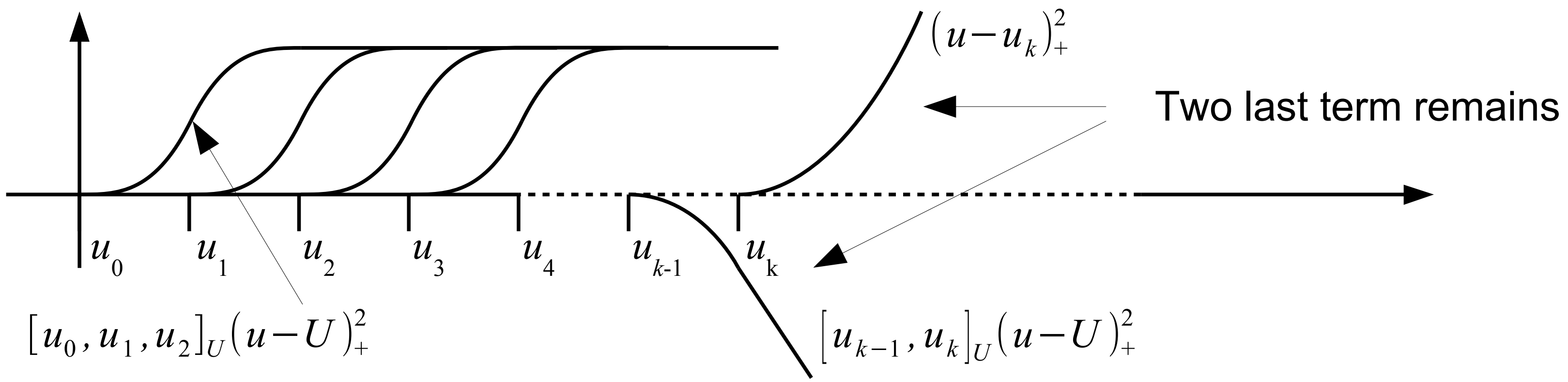
$$\left[u_{k-1}, u_{k-2}\right]_U (u-U)_+^2 - \left[u_k, u_{k-1}\right]_U (u-U)_+^2 \Big|_{u>u_k}$$
$$= 0\cdot u + \left((u_k + u_{k-1}) - (u_{k-1} + u_{k-2})\right)\cdot 1 = (u_k - u_{k-2})\cdot 1$$

Dividing by $(u_k - u_{k-2})$ yields a divided difference again :

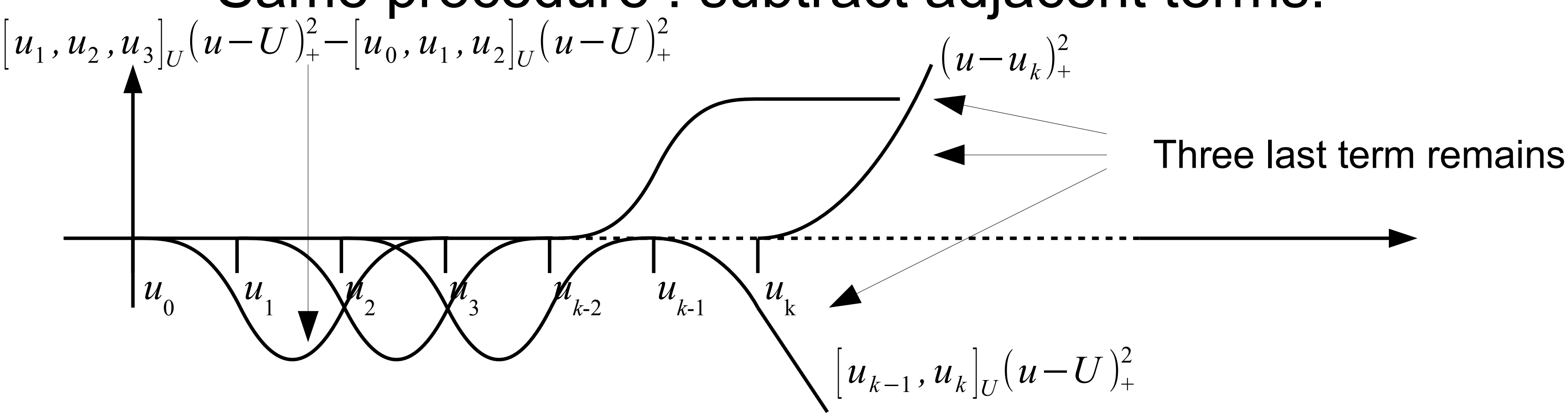
$$\frac{\left[u_{k-1}, u_{k-2}\right]_U (u-U)_+^2 - \left[u_k, u_{k-1}\right]_U (u-U)_+^2}{u_k - u_{k-2}}$$
$$= \left[u_{k-2}, u_{k-1}, u_k\right]_U (u-U)_+^2$$



- Now, cancel constant terms …



- Same procedure : subtract adjacent terms.

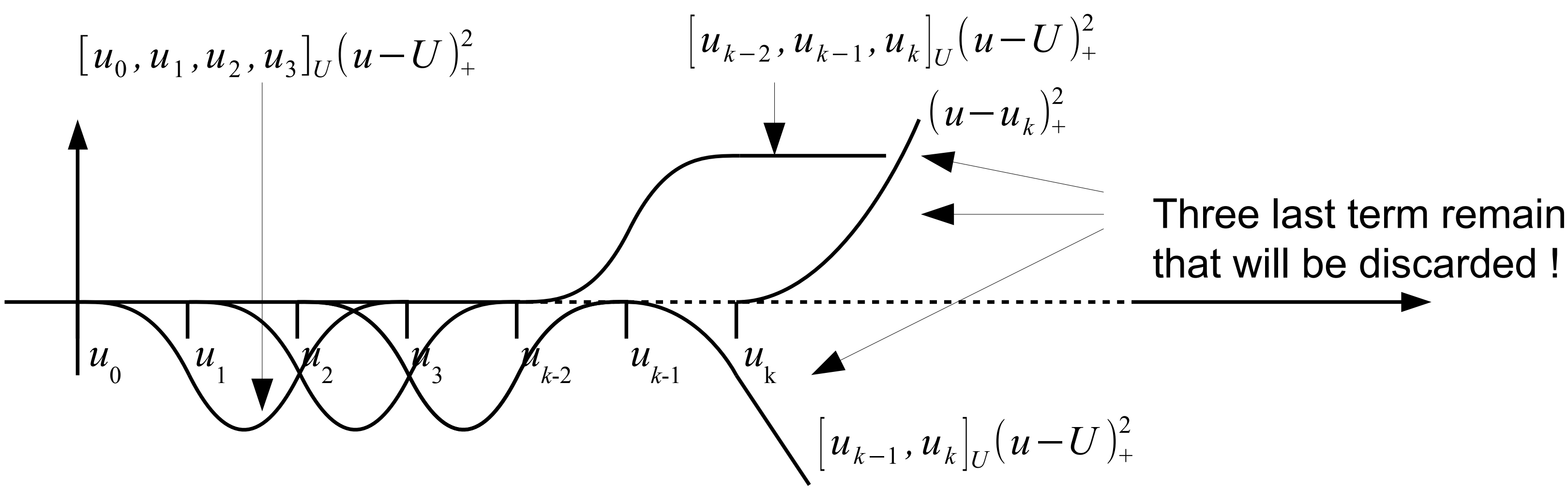


B-Splines

- There are no lower order terms. However we might divide anyway by $(u_k - u_{k-3})$ to remain consistent and get an expression as a divided difference again...

$$\frac{[u_{k-2}, u_{k-1}, u_k]_U (u-U)_+^2 - [u_{k-3}, u_{k-2}, u_{k-1}]_U (u-U)_+^2}{u_k - u_{k-3}}$$

$$= [u_{k-3}, u_{k-2}, u_{k-1}, u_k]_U (u-U)_+^2$$



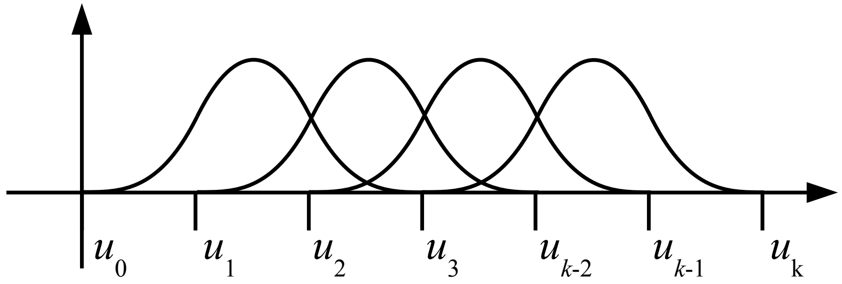
B-Splines

- The sign is alternating with the degree. Shape functions of even degree are negative, while SFs of uneven degree are positive.
- Multiplying by $(-1)^{d+1}$ makes every SF positive.
- To ensure that the SF form a partition of unity , we have to multiply again by $(u_{i+d+1}-u_i)$
- The compact representation of the B-Splines basis functions of degree d with the use of divided differences is therefore :

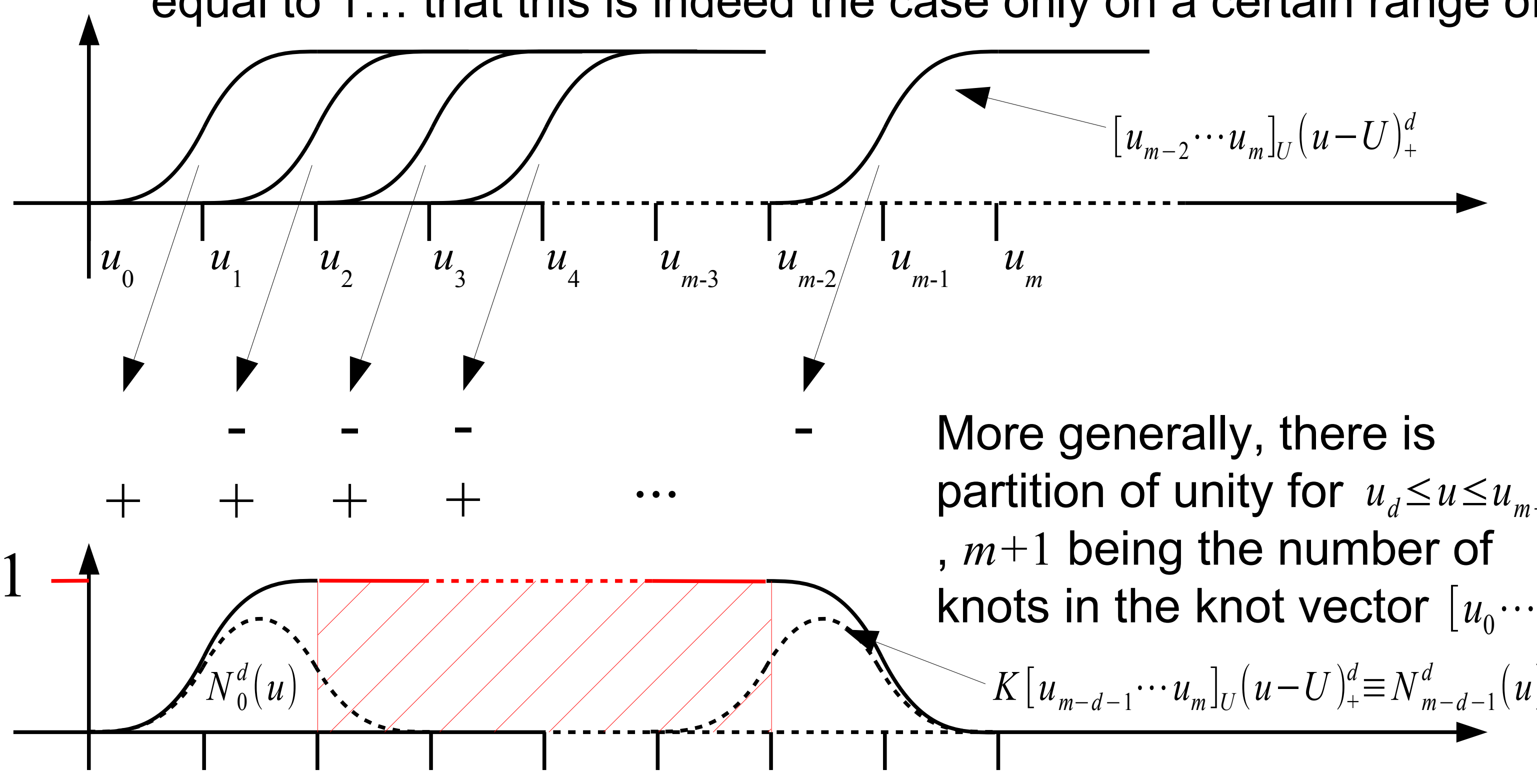
$$N_i^d = (-1)^{d+1}(u_{i+d+1}-u_i)[u_i, \cdots, u_{i+d+1}]_U (u-U)_+^d$$

B-Splines

$$N_i^d = (-1)^{d+1}(u_{i+d+1} - u_i)[u_i, \cdots, u_{i+d+1}]_U (u - U)_+^d$$



- Proof of the partition of unity : consider the before last operation (the cancellation of constant terms)
  - We subtract consecutive terms to form the final shape functions
  - Partition of unity means the sum of all the final shape functions is equal to 1… that this is indeed the case only on a certain range of u.



More generally, there is partition of unity for $u_d \le u \le u_{m-d}$, $m+1$ being the number of knots in the knot vector $[u_0 \cdots u_m]$

- Recurrence relations
  - It is easy to see that $(u-U)_+^d=(u-U)(u-U)_+^{d-1}$
  - Replacing in $N_i^d=(-1)^{d+1}(u_{i+d+1}-u_i)[u_i,\cdots,u_{i+d+1}]_U(u-U)_+^d$ and using the expression of divided differences of a product, $[u_i,\cdots,u_{i+k}]g\cdot h=\sum_{j=i}^{j=i+k}([u_i,\cdots,u_j]g)\cdot([u_j,\cdots,u_{i+k}]h)$ it yields :

$$N_i^d=(-1)^{d+1}(u_{i+d+1}-u_i)\Big([u_i]_U(u-U)[u_i,\cdots,u_{i+d+1}]_U(u-U)_+^{d-1}+[u_i,u_{i+1}]_U(u-U)[u_{i+1},\cdots,u_{i+d+1}]_U(u-U)_+^{d-1}\Big)$$

because DD of order >1 of $(U-u)=0$. It simplifies to :

$$N_i^d=(-1)^{(d+1)}(u_{i+d+1}-u_i)\Big((u-u_i)[u_i,\cdots,u_{i+d+1}]_U(u-U)_+^{d-1}+(-1)[u_{i+1},\cdots,u_{i+d+1}]_U(u-U)_+^{d-1}\Big)$$

B-Splines

$$N_i^d = (-1)^{(d+1)}(u_{i+d+1}-u_i)\Big((u-u_i)[u_i,\cdots,u_{i+d+1}]_U(u-U)_+^{d-1} + (-1)[u_{i+1},\cdots,u_{i+d+1}]_U(u-U)_+^{d-1}\Big)$$

- Using the recursive definition of the divided differences

$$[u_i,\cdots,u_{i+d+1}]f = \frac{[u_{i+1},\cdots,u_{i+d+1}]f - [u_i,\cdots,u_{i+d}]f}{u_{i+d+1}-u_i}$$

one gets :

$$N_i^d = (-1)^d(u-u_i)[u_i,\cdots,u_{i+d}]_U(u-U)_+^{d-1} + (-1)^d(u_{i+d+1}-u)[u_{i+1},\cdots,u_{i+d+1}]_U(u-U)_+^{d-1}$$

which is, to a term depending on the knot vector and u, equal to $N_i^d = \dfrac{u-u_i}{u_{i+d}-u_i}N_i^{d-1} + \dfrac{u_{i+d+1}-u}{u_{i+d+1}-u_{i+1}}N_{i+1}^{d-1}$.

- Recursive definition of basis functions
  - Setting $U=\{u_0,\cdots,u_m\}$, $u_i \le u_{i+1}$, $i=0\cdots m-1$ (nodal sequence)
  - The functions are such as : (recurrence formula of Cox – de Boor)

$$N_i^0(u)=\begin{cases}1 & \text{if} \quad u_i \le u < u_{i+1} \\ 0 & \text{otherwise}\end{cases}$$

$$N_i^d(u)=\frac{u-u_i}{u_{i+d}-u_i}N_i^{d-1}(u)+\frac{u_{i+d+1}-u}{u_{i+d+1}-u_{i+1}}N_{i+1}^{d-1}(u)$$

  - Where $u_{i+d}-u_i=0$, necessarily $N_i^{d-1}(u)\equiv 0$

    By convention, we set in this case $\frac{0}{0}=0$ when the limit is undefined.

- Example : computation of basis functions of degree $d \leq 2$ for $U = \{u_0 = 0, u_1 = 0, u_2 = 0, u_3 = 1, u_4 = 1, u_5 = 1\}$

$$N_0^1(u) = \frac{u - u_0}{u_1 - u_0} N_0^0(u) + \frac{u_2 - u}{u_2 - u_1} N_1^0(u) \longrightarrow \frac{0}{0} = 0$$

by convention

$$N_0^0 = 0 \qquad N_1^0 = 0 \qquad N_2^0 = \begin{cases} 1 & 0 \leq u < 1 \\ 0 & \text{otherwise} \end{cases} \qquad N_3^0 = 0 \qquad N_4^0 = 0$$

$$N_0^1 = 0 \qquad N_1^1 = \begin{cases} 1 - u & 0 \leq u < 1 \\ 0 & \text{otherwise} \end{cases} \qquad N_2^1 = \begin{cases} u & 0 \leq u < 1 \\ 0 & \text{otherwise} \end{cases} \qquad N_3^1 = 0$$

$$N_0^2 = \begin{cases} (1-u)^2 & 0 \leq u < 1 \\ 0 & \text{otherwise} \end{cases}$$

$$N_1^2 = \begin{cases} 2u(1-u) & 0 \leq u < 1 \\ 0 & \text{otherwise} \end{cases}$$

$$N_2^2 = \begin{cases} u^2 & 0 \leq u < 1 \\ 0 & \text{otherwise} \end{cases}$$

Bernstein polynomials of degree 2

- The Bernstein polynomials of degree d are a particular case of the B-splines basis
  - They correspond to a nodal sequence

$$U_B = \{u_0 = 0, \cdots, u_d = 0, u_{d+1} = 1, \cdots, u_{2d+1} = 1\}$$

  - Bézier curves are therefore a particular case of B-splines.
  - It is also possible to transform any B-spline into a sequence of Bézier curves – because the Bernstein polynomials form a complete basis of polynomials of degree d.

- Basis functions and control points
  - In contrary to Bézier curves, the number of control points is not imposed by the degree d
  - Let $m+1$ the number of knots. We have $n+1$ independent basis functions at our hands
    - For every basis function, we associate a control point

$$P(u)=\sum_{i=0}^{n} P_i N_i^d(u)$$

  - The number of control points is fixed by the relation $n+1=m-d$

- Types of nodal sequence...
  - Uniform – The gap between two successive knots is constant

$$U=\{u_0, u_1, \cdots, u_{m-d-1}\} \quad , \quad u_{i+1}-u_i=k$$

  - Periodic - The gap between the knots at the start of a nodal sequence is identical to the one at the end of the nodal sequence

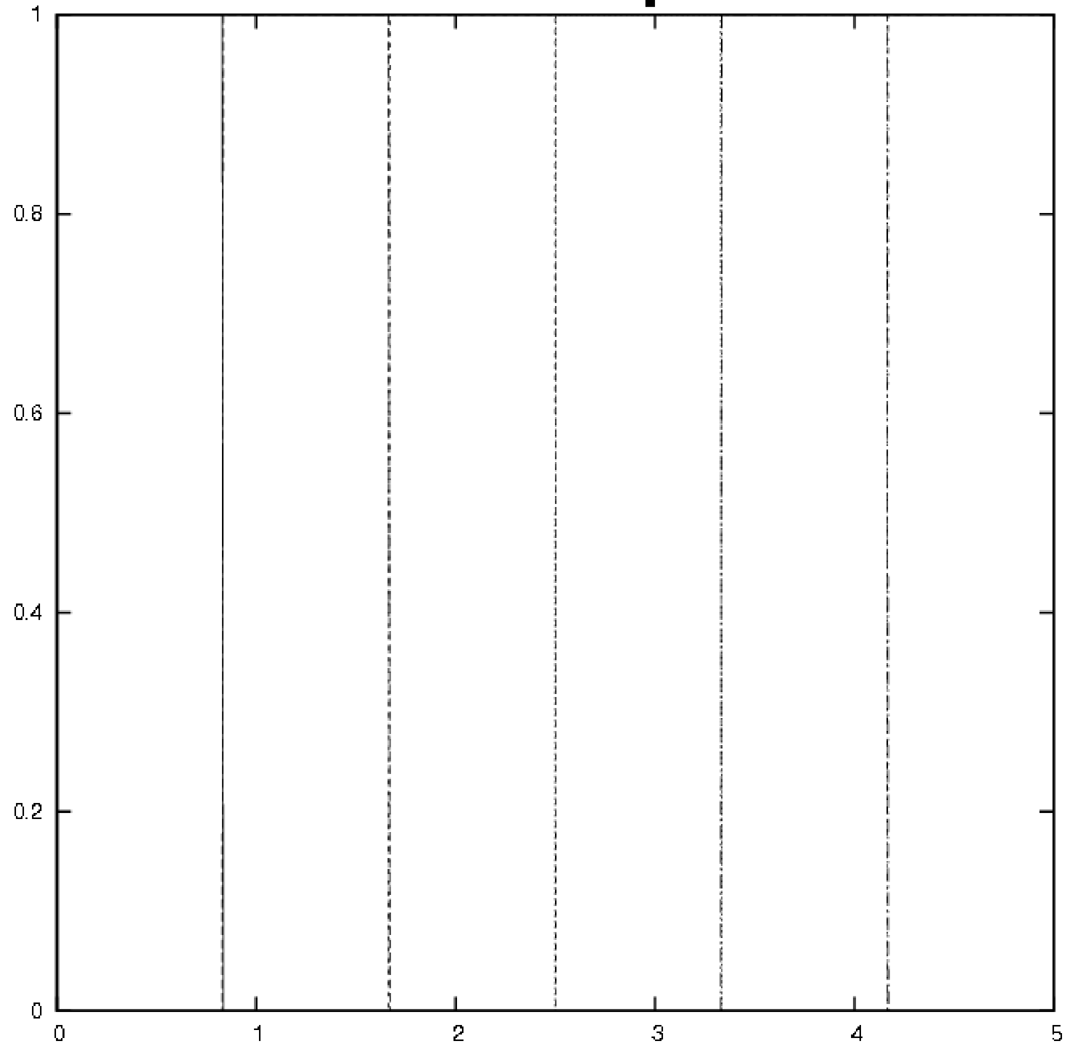
$$U=\{\underbrace{u_{0,}\cdots, u_d}_{d+1}, u_{d+1}, \cdots, u_{m-d-1}, \underbrace{u'_{0,}\cdots, u'_d}_{d+1}\} \quad , \quad u'_i-u_i=k$$

  - Non uniform, interpolating – first and last control point are interpolated

$$U=\{\underbrace{a, \cdots, a}_{d+1}, u_{d+1}, \cdots, u_{m-d-1}, \underbrace{b, \cdots, b}_{d+1}\}$$

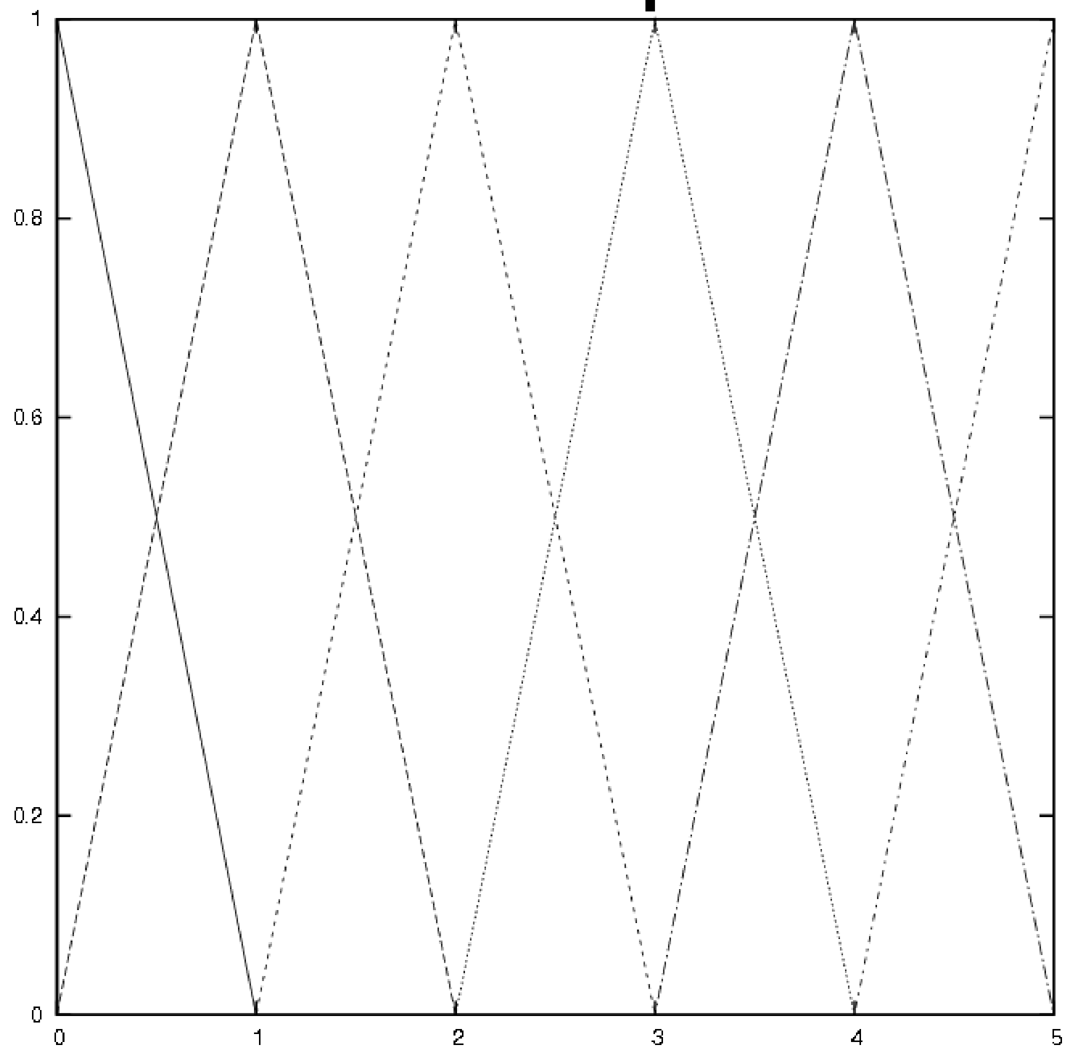
In the sequel, except where indicated, we consider non uniform nodal sequences interpolating the first and last control points.

B-Splines



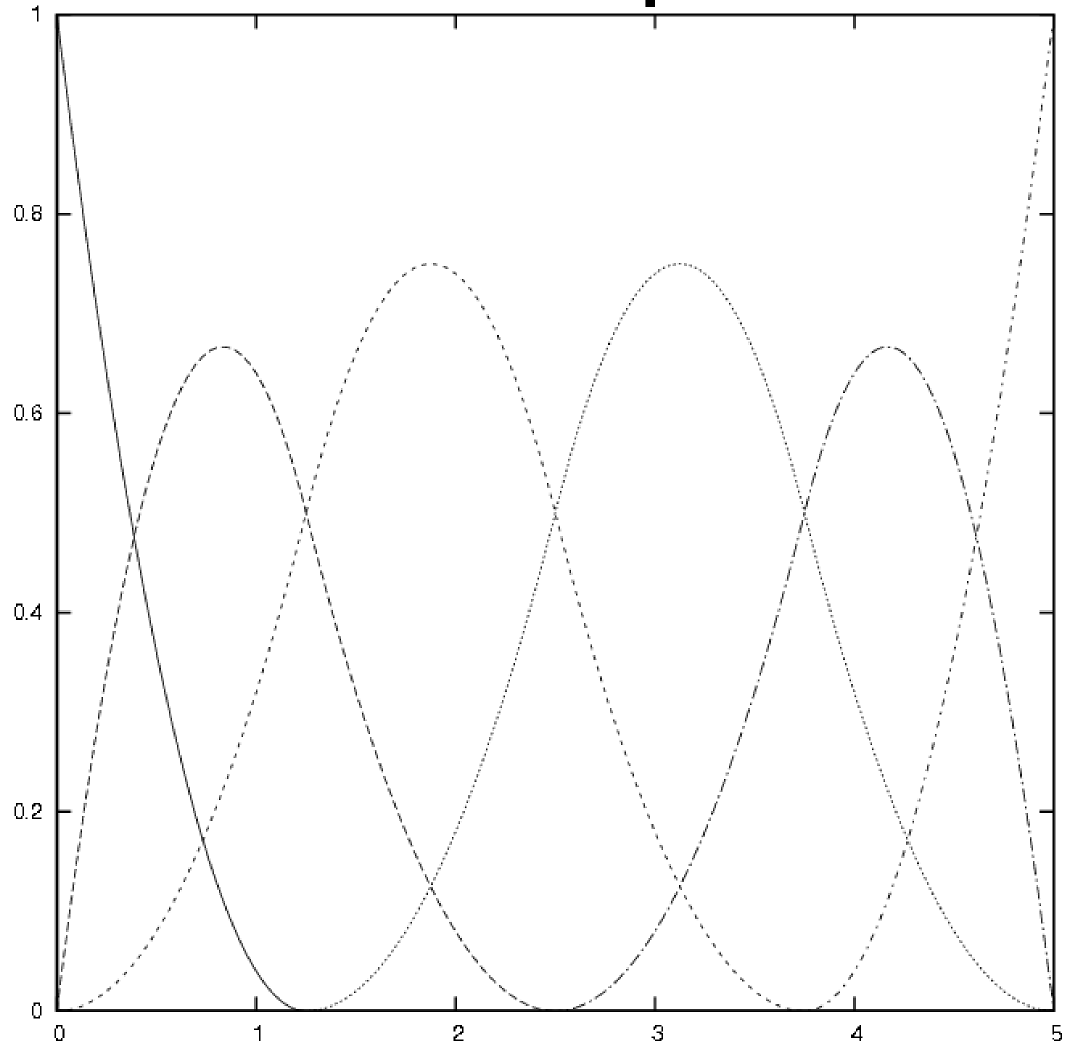
$$U = \left\{0, \frac{5}{6}, \frac{10}{6}, \frac{15}{6}, \frac{20}{6}, \frac{25}{6}, 5\right\} \quad d = 0 \quad m+1 = 7 \quad n+1 = 6$$

B-Splines



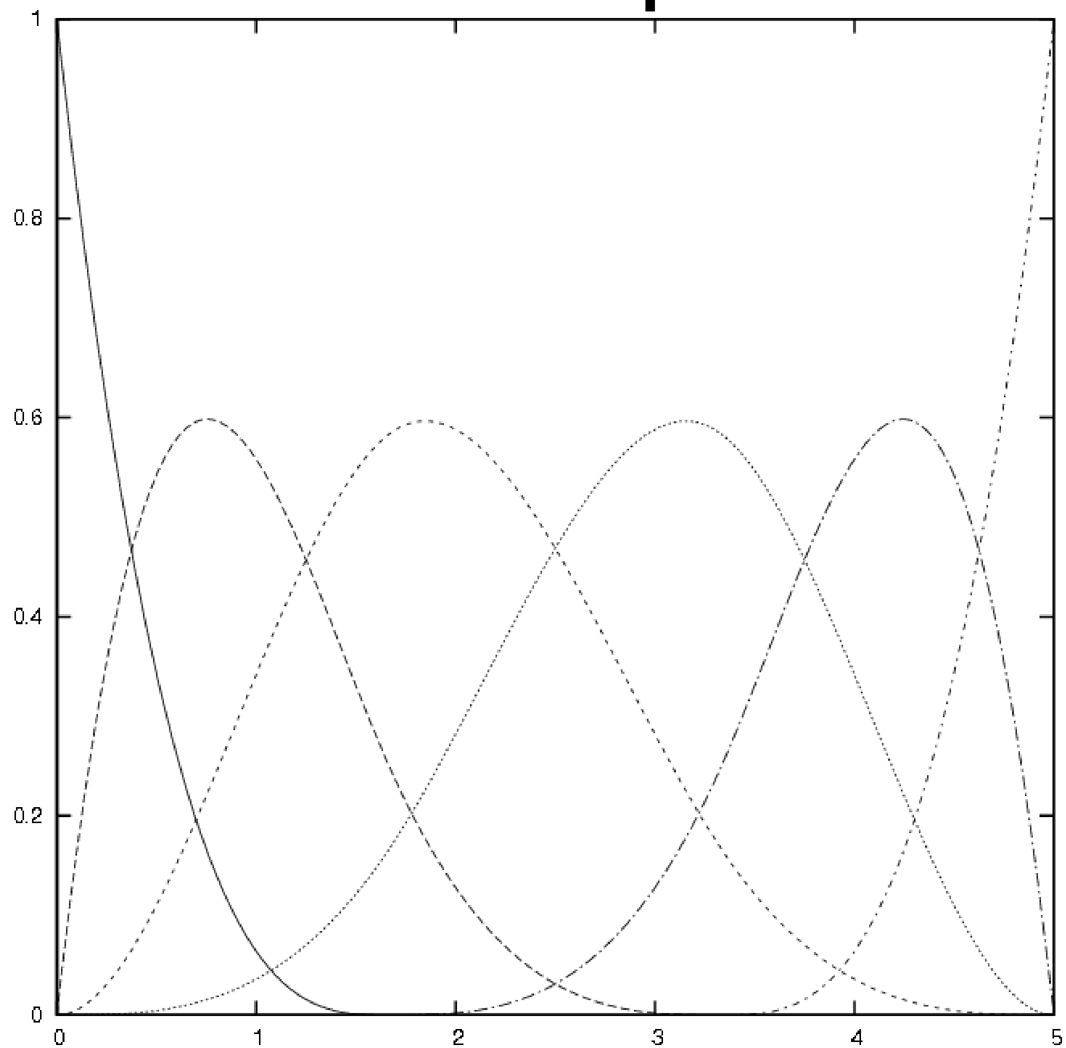
$$U=\{0,0,1,2,3,4,5,5\} \quad d=1 \quad m+1=8 \quad n+1=6$$

B-Splines



$$U=\left\{0,0,0,\frac{5}{4},\frac{10}{4},\frac{15}{4},5,5,5\right\}\quad d=2\quad m+1=9\quad n+1=6$$

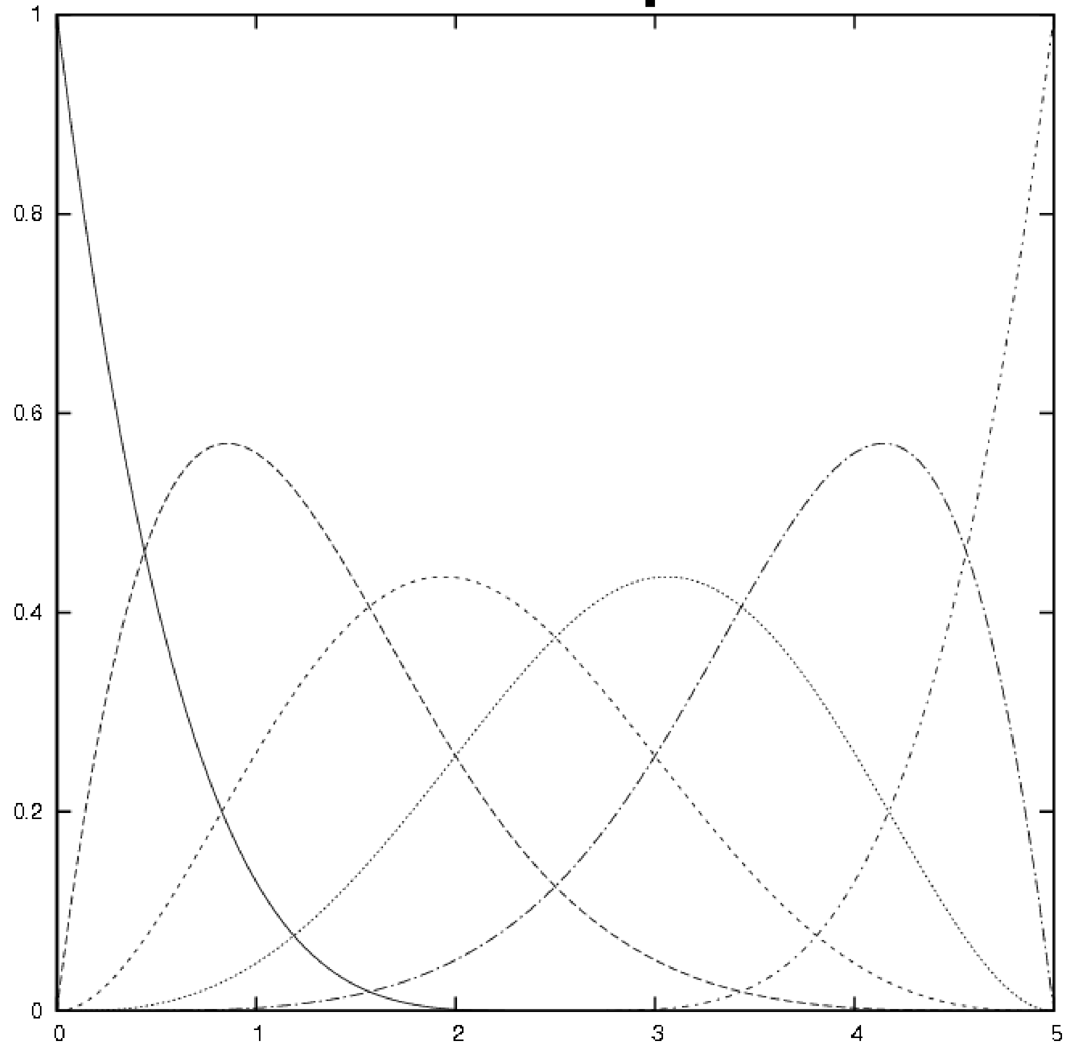
B-Splines



$$U=\left\{0\,,0\,,0\,,0,\frac{5}{3},\frac{10}{3},5\,,5\,,5\,,5\right\}\quad d=3\quad m+1=10\quad n+1=6$$

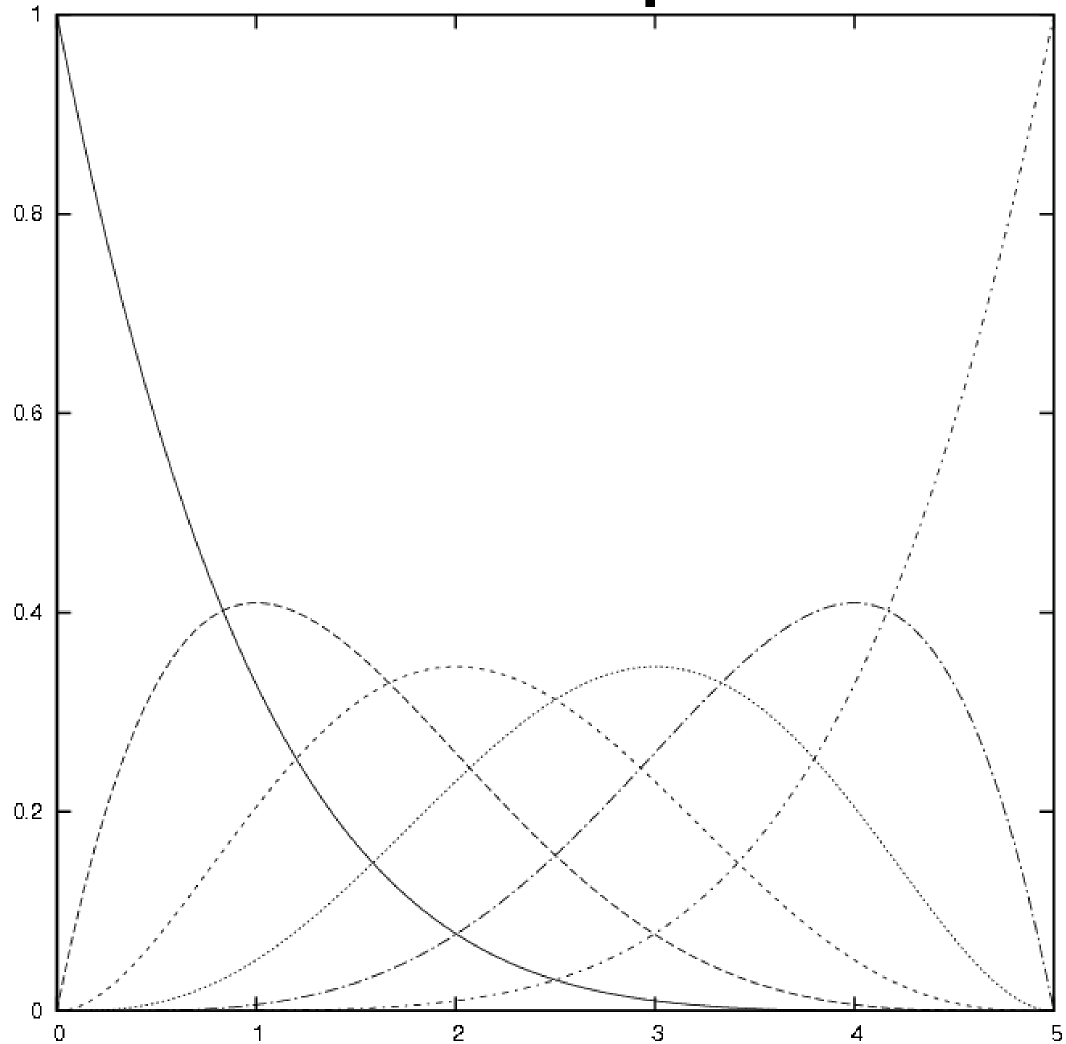
Computer Aided Design

B-Splines



$$U = \left\{0, 0, 0, 0, 0, \frac{5}{2}, 5, 5, 5, 5, 5\right\} \quad d = 4 \quad m+1 = 11 \quad n+1 = 6$$

B-Splines



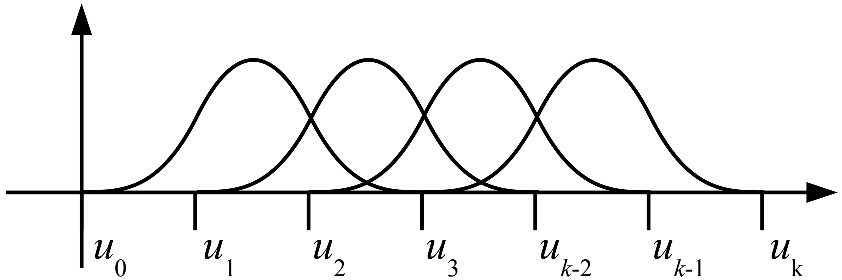
$$U=\{0,0,0,0,0,0,5,5,5,5,5,5\} \quad d=5 \quad m+1=12 \quad n+1=6$$

Bernstein polynomials (with a factor on u)

- Properties of B-spline basis functions
  - $N_i^d(u)=0$ outside the interval $[u_i, u_{i+d+1}[$
  - Inside the interval $[u_i, u_{i+1}[$, at most $d+1$ functions $N_*^d(u)$ are non zero : $N_{i-d}^d, \cdots, N_i^d$
  - $N_i^d(u)\geq 0 \quad \forall\, i\,, d$ and u_i (always positive)
  - For $u\in[u_i, u_{i+1}[\ , \sum_{j=i-d} N_j^d(u)=1$ (forms a partition of unity)
  - All derivatives of $N_i^d(u)$ exist inside the interval $[u_i, u_{i+1}[$. At a knot , $N_i^d(u)$ is d-k times differentiable, k being the node multiplicity.
  - Except for $d=0$, $N_i^d(u)$ reaches exactly one maximum

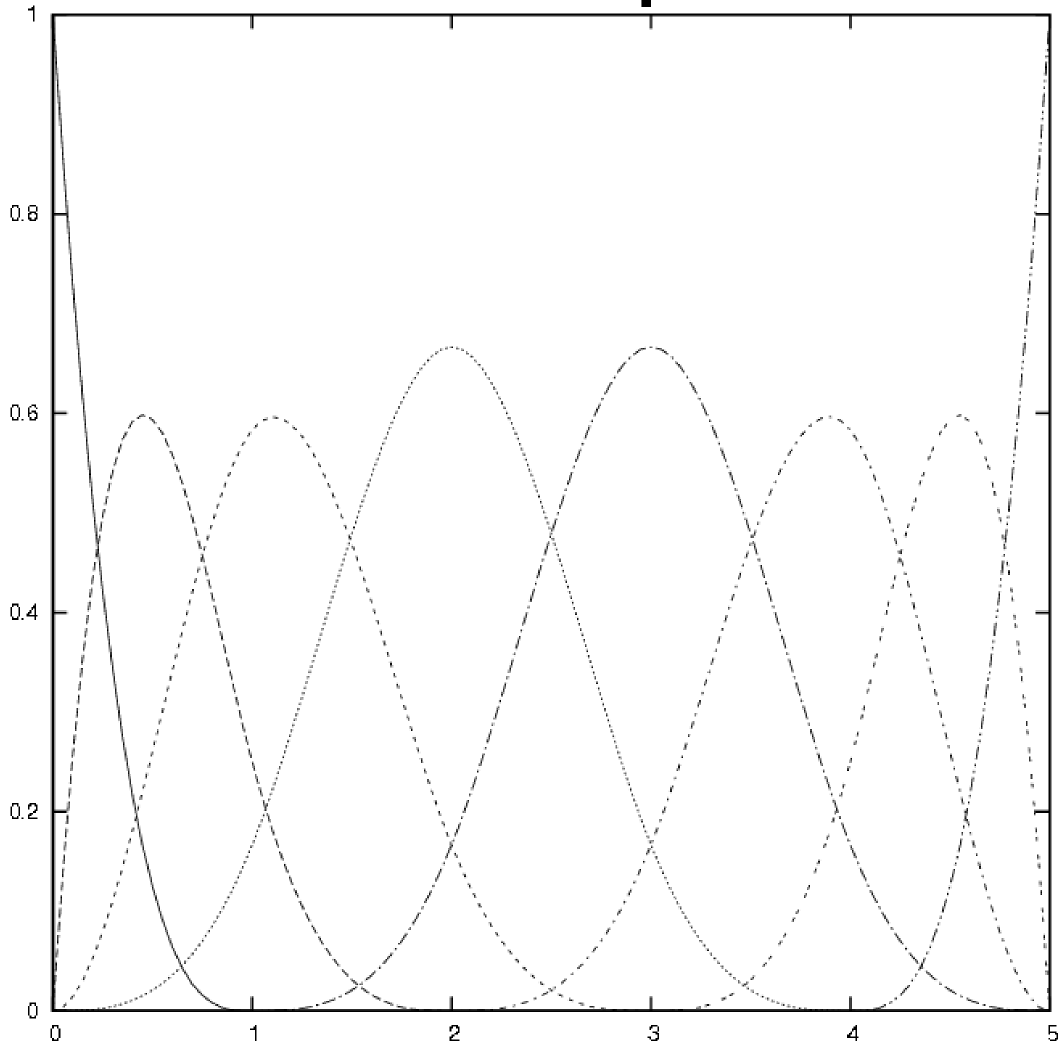
B-Splines

$$N_i^d = (-1)^{d+1} (u_{i+d+1} - u_i) [u_i, \cdots, u_{i+d+1}]_U (u - U)_+^d$$



Computer Aided Design

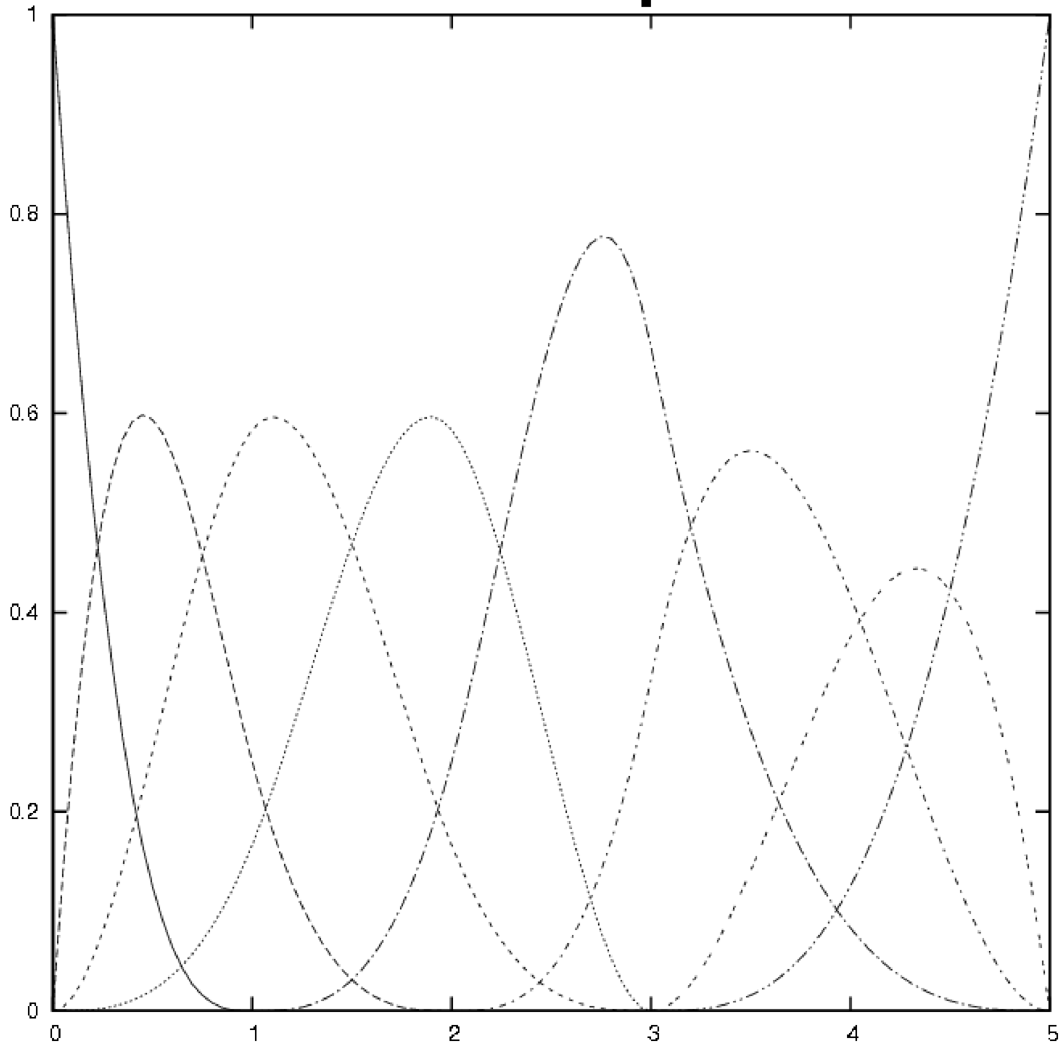
B-Splines



$$U=\{0,0,0,0,1,2,3,4,5,5,5,5\} \quad d=3 \quad m+1=12 \quad n+1=8$$

The knot u=3 is of multiplicity 1

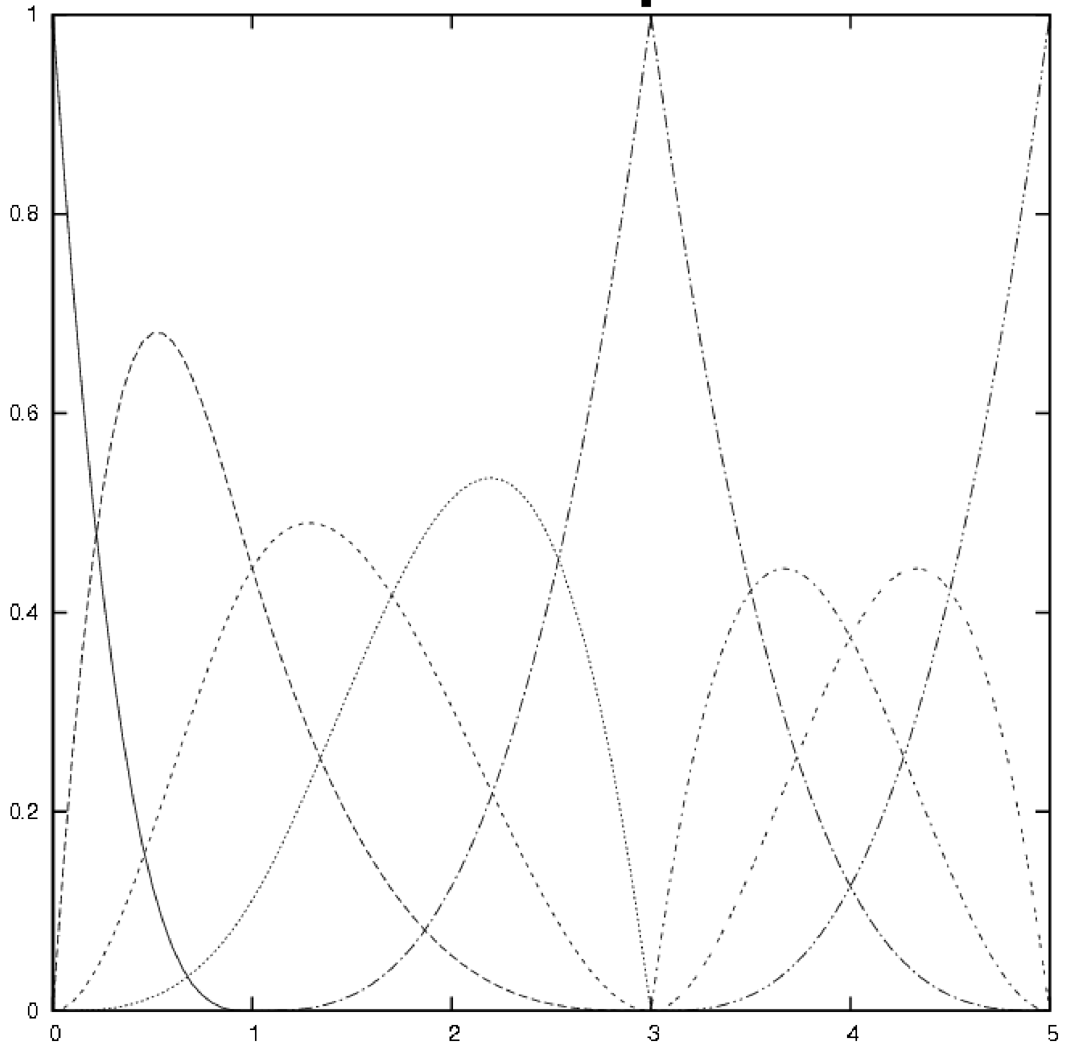
B-Splines



$$U=\{0,0,0,0,1,2,3,3,5,5,5,5\} \quad d=3 \quad m+1=12 \quad n+1=8$$

The node u=3 is of multiplicity 2

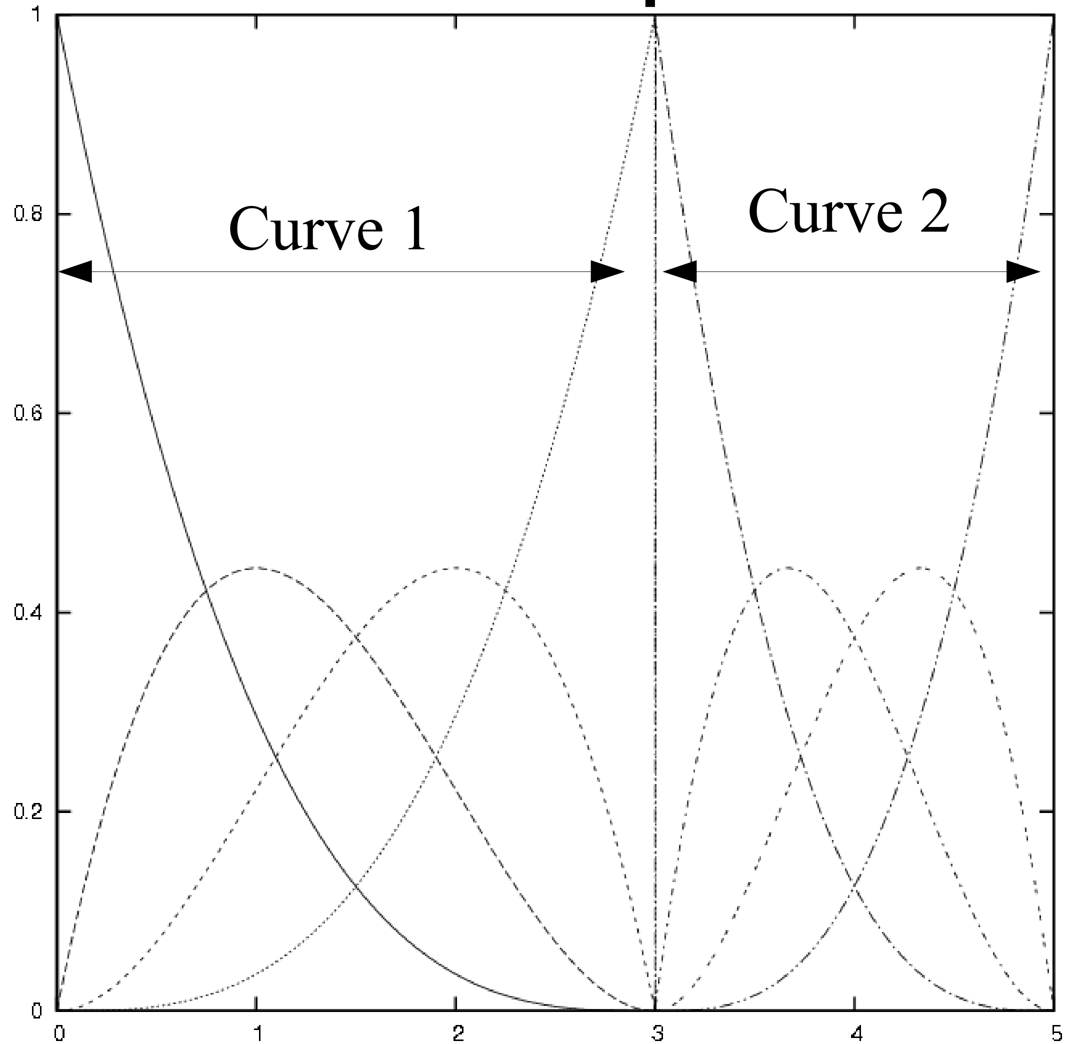
B-Splines



$$U = \{0,0,0,0,1,3,3,3,5,5,5,5\} \quad d=3 \quad m+1=12 \quad n+1=8$$

The node u=3 is of multiplicity 3

B-Splines



$$U=\{0,0,0,0,3,3,3,3,5,5,5,5\} \quad d=3 \quad m+1=12 \quad n+1=8$$

The node u=3 is of multiplicity 4

- Derivatives of B-spline basis functions
  - Definition by recurrence

$$\frac{d^k N_i^d}{d u^k} = N_i^{d,(k)} = d\left(\frac{N_i^{d-1,(k-1)}}{u_{i+d}-u_i} - \frac{N_{i+1}^{d-1,(k-1)}}{u_{i+d+1}-u_{i+1}}\right)$$

  - k should not exceed d : every derivative of higher order vanish.

B-Splines

- The characteristics of basis functions involve that the B-Spline curve

$$P(u)=\sum_{i=0}^{n} P_i N_i^d(u) \qquad U=\{u_0,\cdots,u_m\}\ ,\ u_i \le u_{i+1}\ ,\ i=0\cdots m-1$$

  - interpolates P_0 and P_n , **only if** the nodal sequence admits $d+1$ repetitions at the start and at the end !
  - is invariant by affine transformation ,
  - is contained by the convex hull of the control points (because $P(u)$ is a linear combination of the control points with positive coefficients which sum to one)

(Following)

- Is *variation diminishing* : The number of inflexion points is lower than the number of wiggles of the characteristic polygon
- Is closed and convex if the characteristic polygon is closed and convex,
- Is of length shorter or equal than that of the control polygon.
- Is invariant by linear transformation of the nodal sequence $u'=au+b$, $a>0$

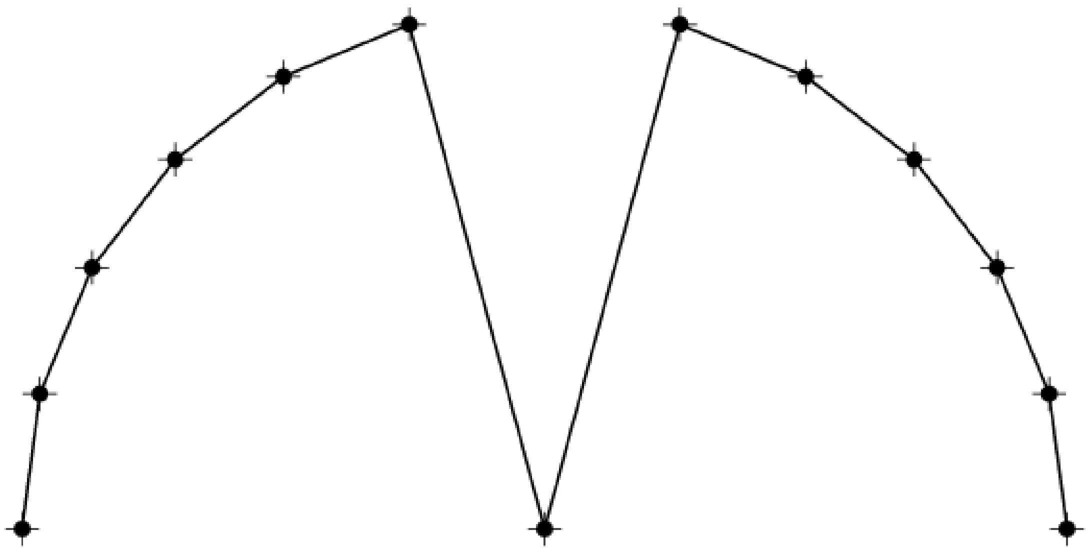
- Control points, degree and nodal sequence
  - We associate a control point for each basis function N_i^*. We have $n+1$ control points.
  - The degree d is chosen by the user.
  - The nodal sequence (that defines the intervals of the parameter on which the curve has a unique polynomial definition) is then built. We have $m+1=n+d+2$ knots (with $d+1$ repetitions at the start and at the end)
    - there remains n-d values of the parameter to set (without taking the boundaries into account)

B-Splines

- Geometric examples
  - Constant number of control points
  - We increase the degree
  - Uniform repartition of knots (except at boundaries)
  - For which degree do we have the best approximation of the control points ??

Computer Aided Design

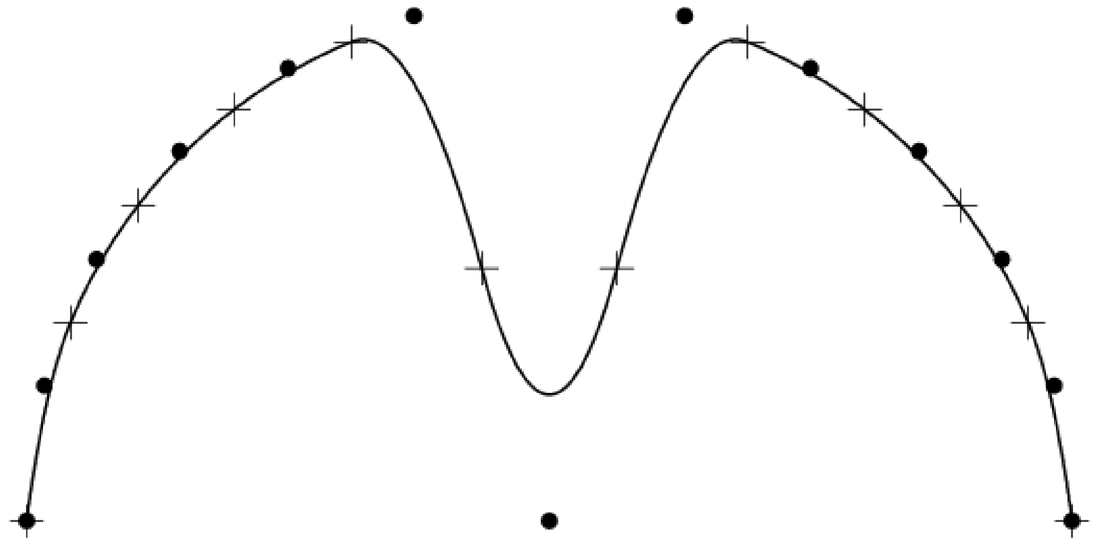
B-Splines



Degree 1
0 0 0.0833333 0.166667 0.25 0.333333 0.416667 0.5 0.583333 0.666667 0.75 0.833333 0.916667 1 1

Computer Aided Design

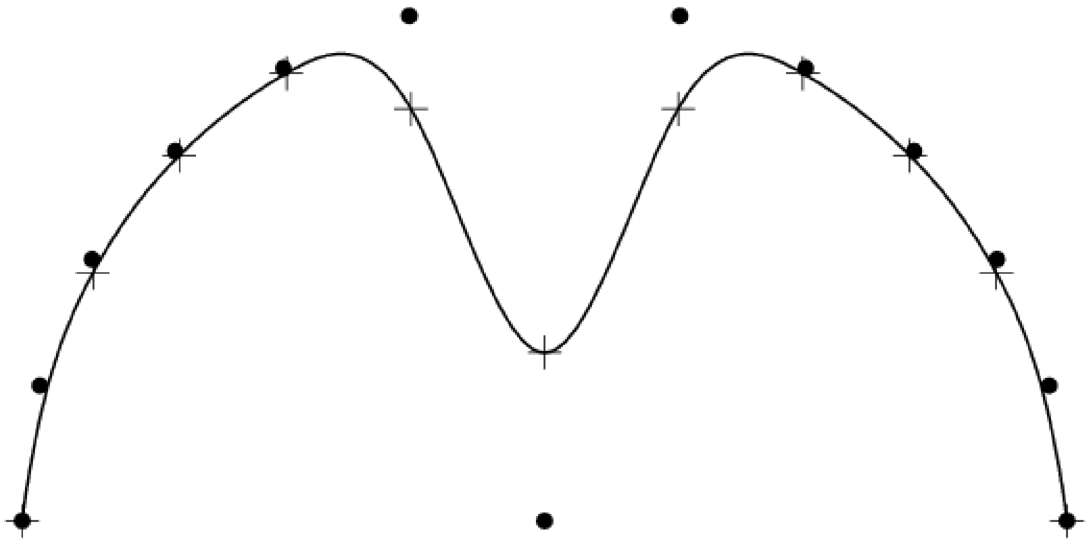
B-Splines



degree 2

0 0 0 0.0909091 0.181818 0.272727 0.363636 0.454545 0.545455 0.636364 0.727273 0.818182 0.909091 1 1 1

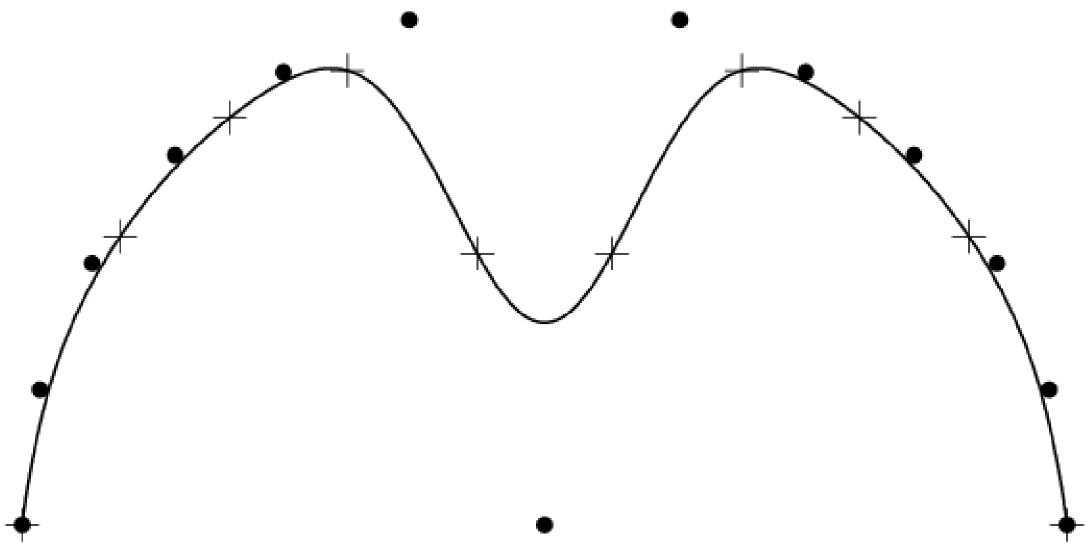
B-Splines



degree 3

0 0 0 0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 1 1 1

B-Splines

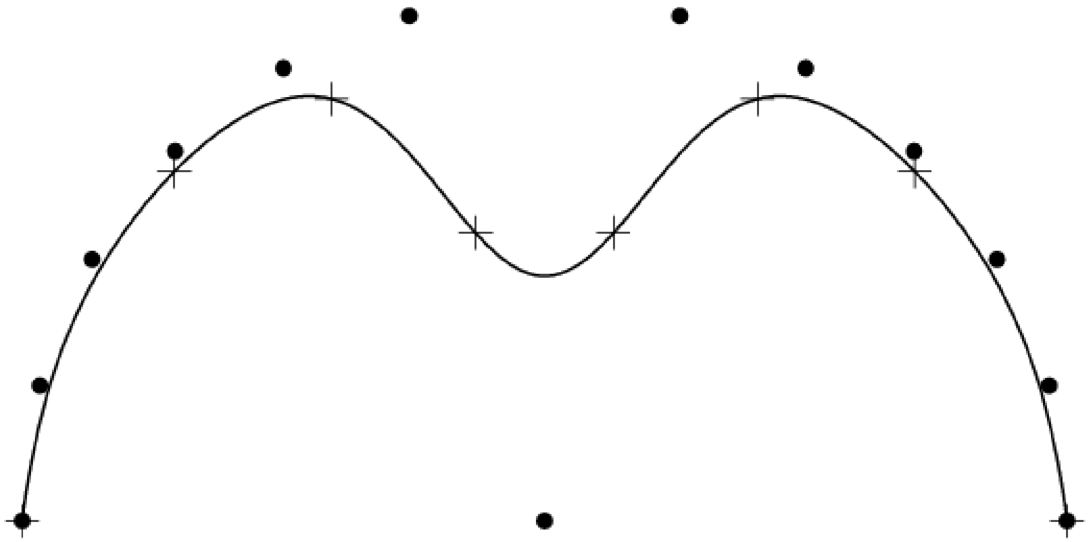


degree 4

0 0 0 0 0 0.111111 0.222222 0.333333 0.444444 0.555556 0.666667 0.777778 0.888889 1 1 1 1 1

Computer Aided Design

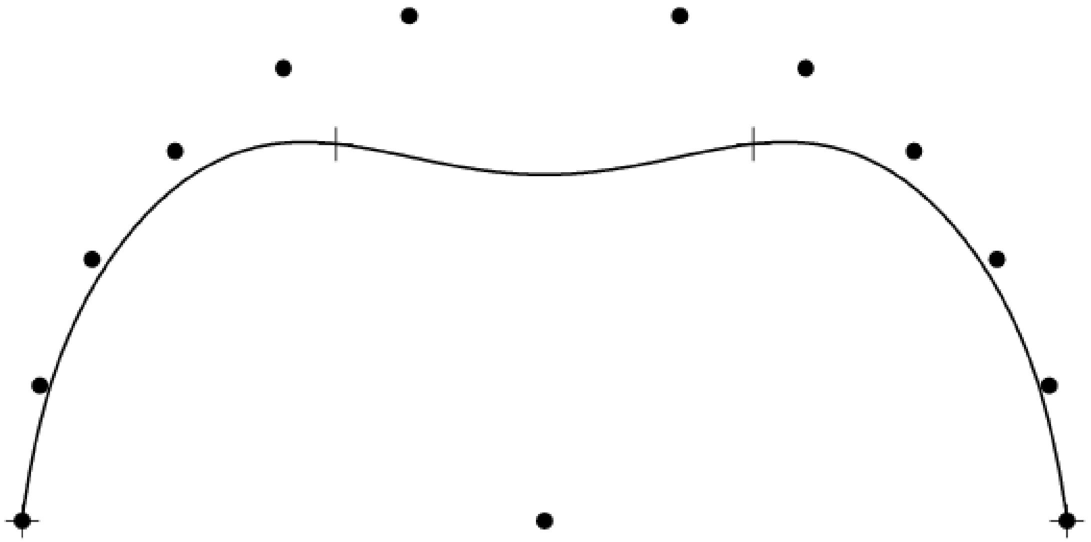
B-Splines



degree 6

0 0 0 0 0 0 0 0.142857 0.285714 0.428571 0.571429 0.714286 0.857143 1 1 1 1 1 1 1

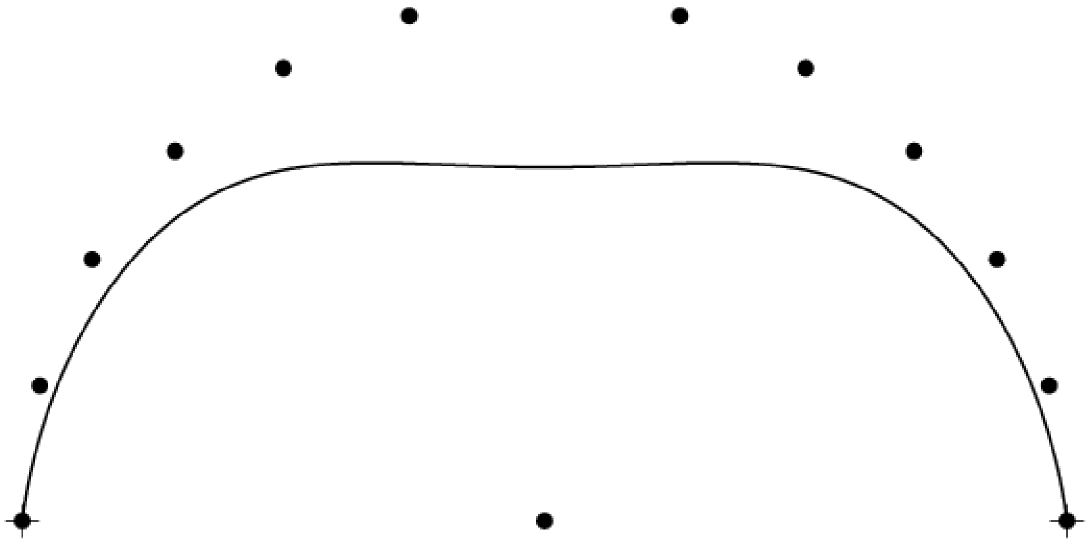
B-Splines



degree 10
0 0 0 0 0 0 0 0 0 0 0 0.333333 0.666667 1 1 1 1 1 1 1 1 1 1 1

Computer Aided Design

B-Splines



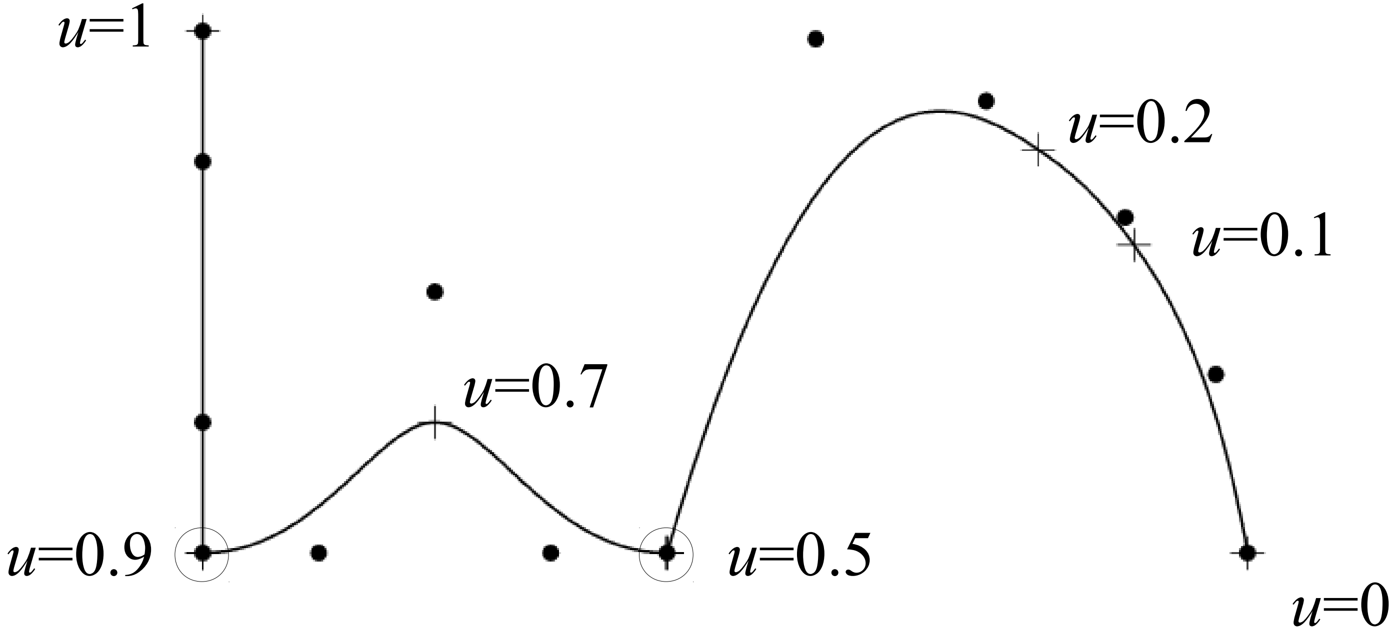
degree 12 (Bézier)

0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1

B-Splines

- Impose interpolation points (and C_0 continuity)
  - It is the same as positioning knots of multiplicity d in the nodal sequence
  - One could also repeat d control points...(not shown here)

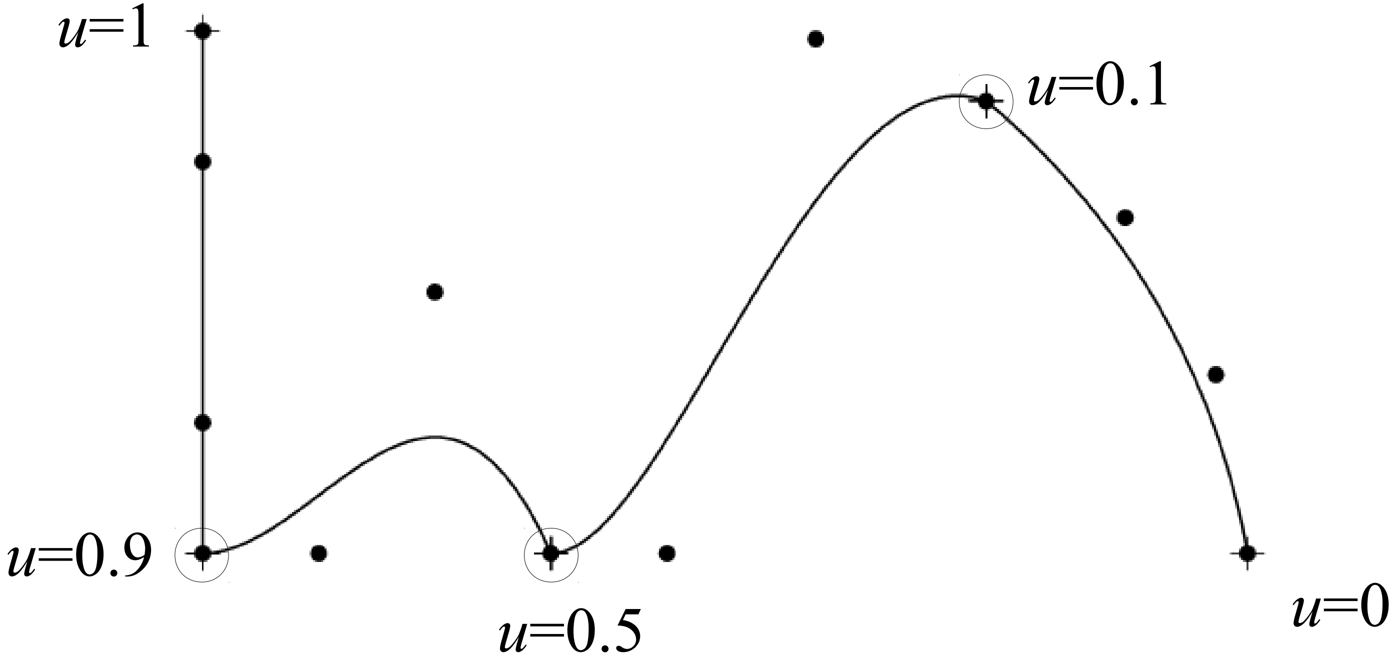
B-Splines



degree 3

0 0 0 0 0.1 0.2 0.5 0.5 0.5 0.7 0.9 0.9 0.9 1 1 1 1

B-Splines

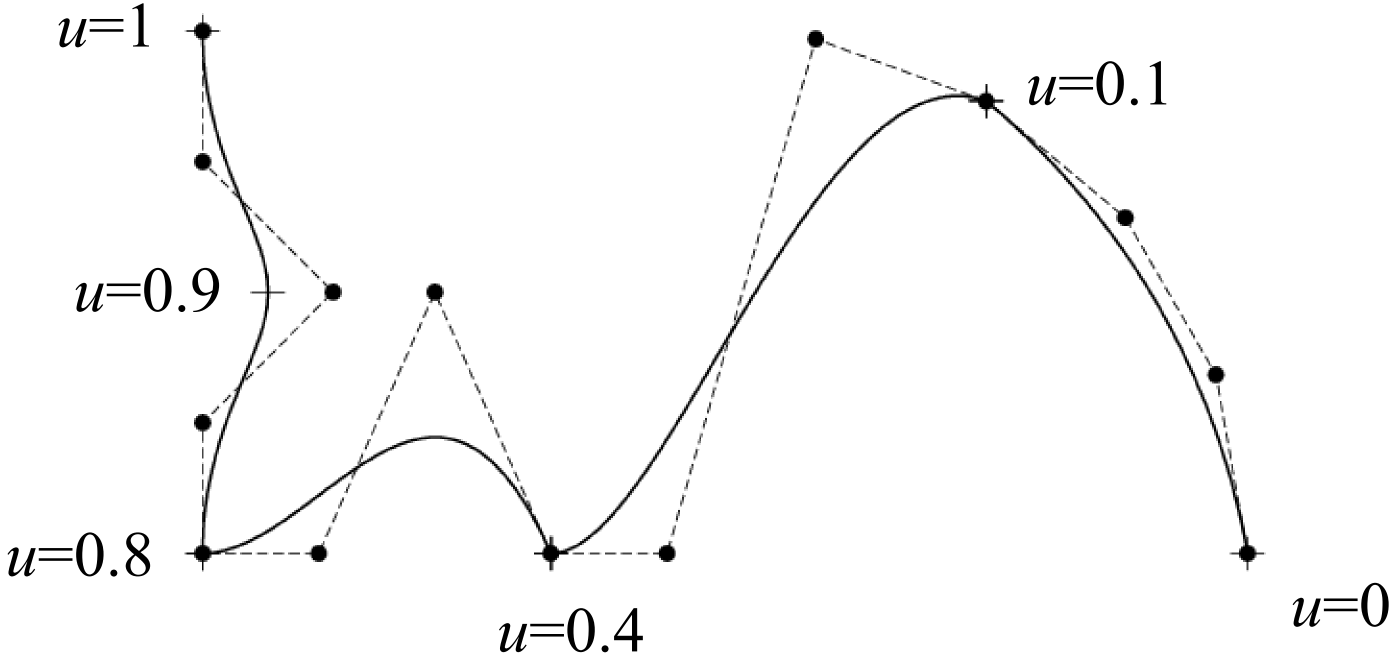


degree 3 (4 Bézier curves of continuity C_0)

0 0 0 0 0.1 0.1 0.1 0.5 0.5 0.5 0.9 0.9 0.9 1 1 1 1

Computer Aided Design

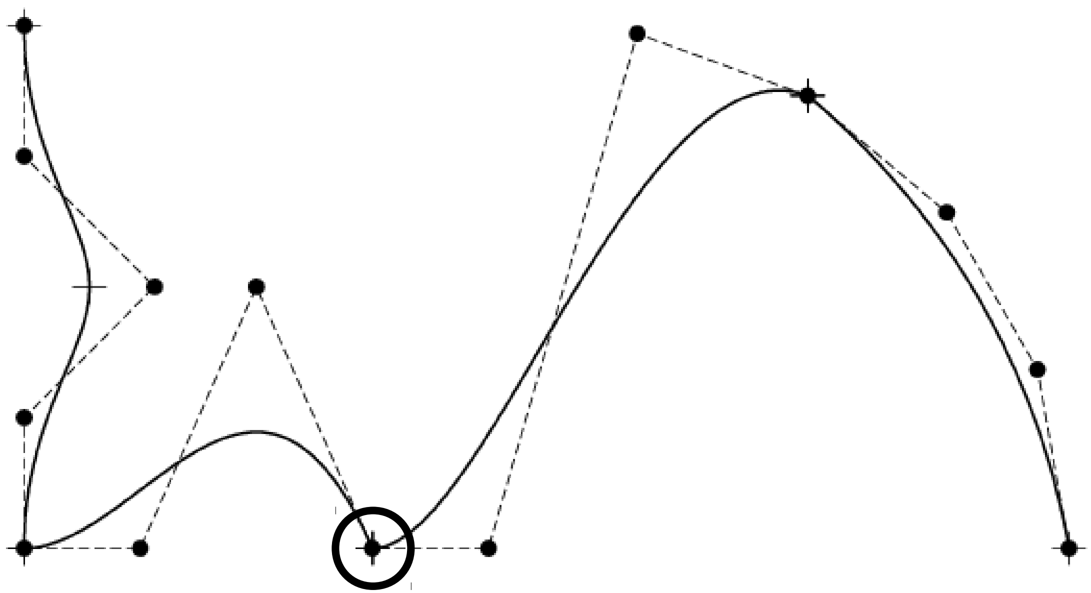
B-Splines



degree 3 (3 Bézier curves of continuity C_0 + 1 bspline deg 3 with 4control pts)

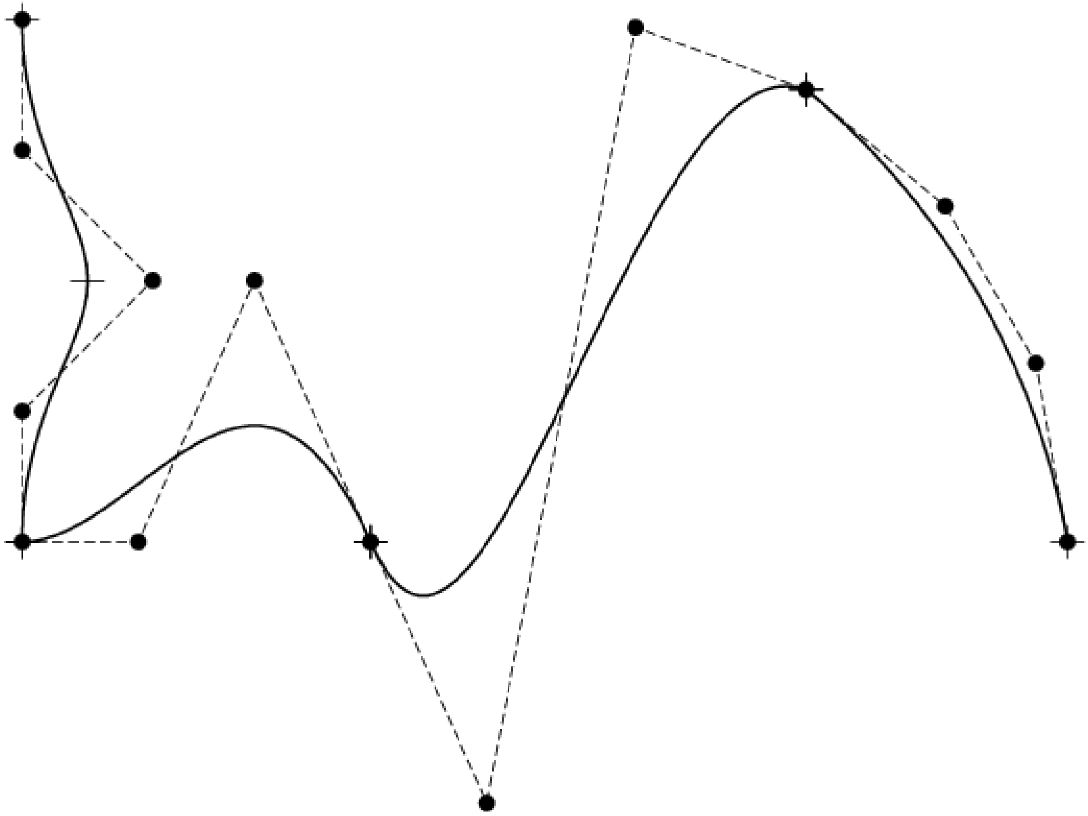
0 0 0 0 0.1 0.1 0.1 0.4 0.4 0.4 0.8 0.8 0.8 0.9 1 1 1 1

- And if we want to impose interpolation points ***and*** a certain continuity C_k ?
  - Add / align control points in a similar way than in the case of Bézier curves.

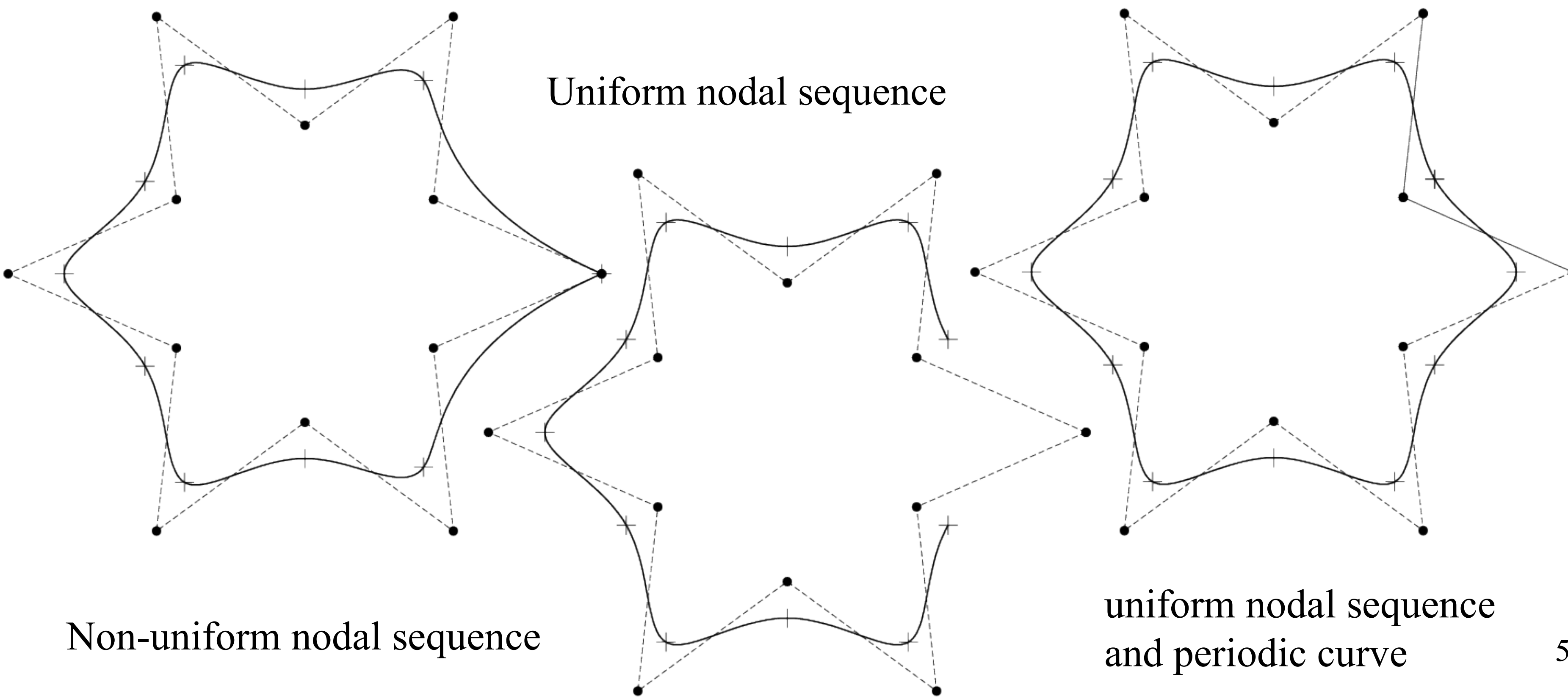


Computer Aided Design

B-Splines



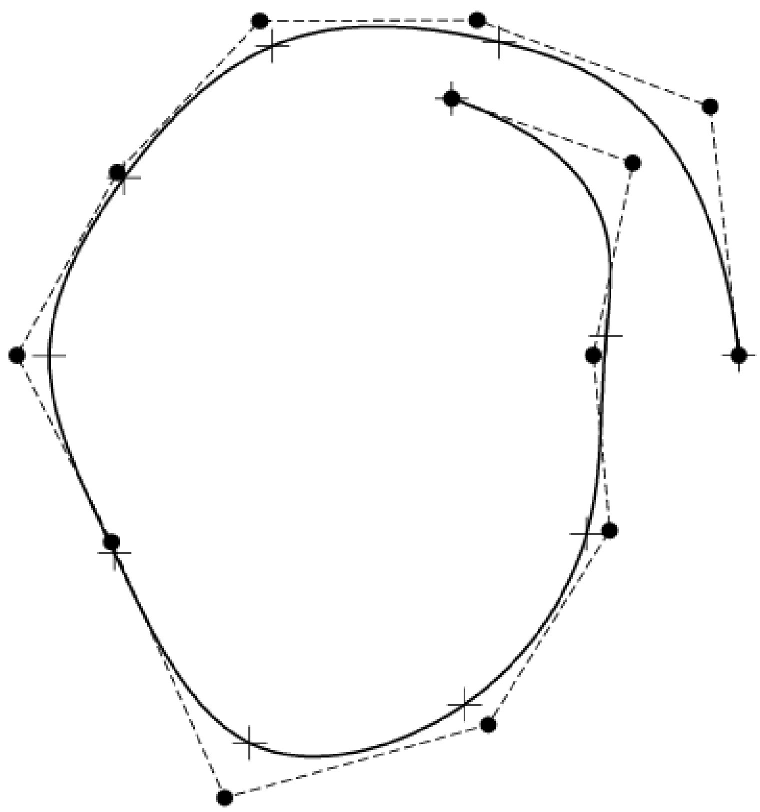
- Periodic curves
  - They may be represented by modifying the nodal sequence and by repeating some control points.



Computer Aided Design

B-Splines

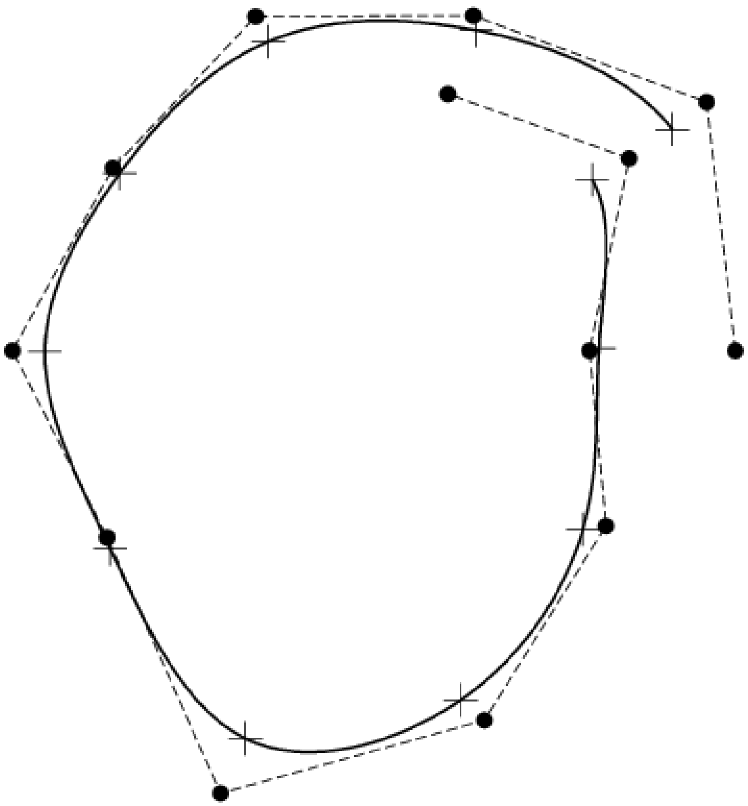
non uniform nodal sequence interpolating
the first and last control points.



degree 3
0 0 0 0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 1 1 1

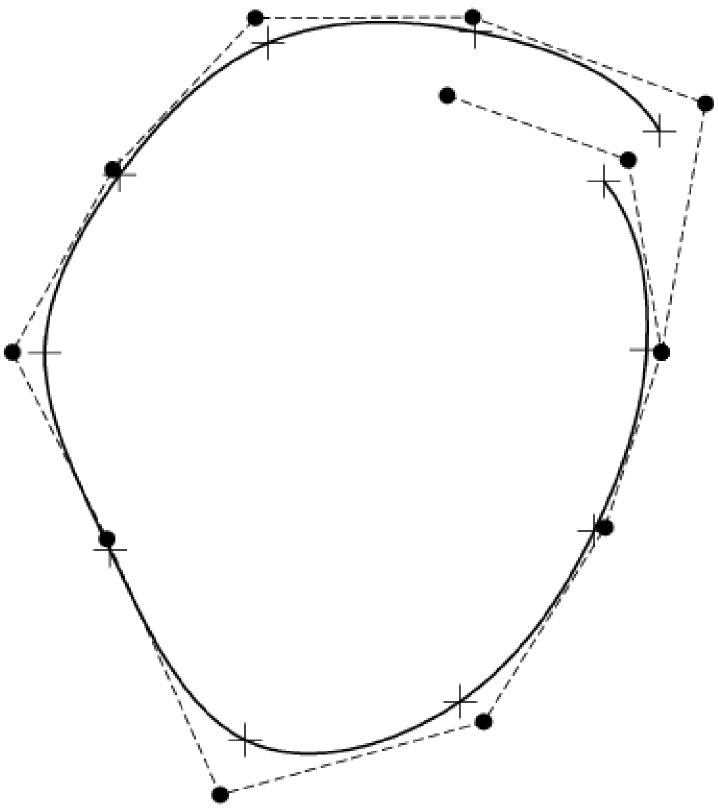
B-Splines

Periodic nodal sequence
(but control points located
inadequately)



degree 3
-0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 1.1 1.2 1.3

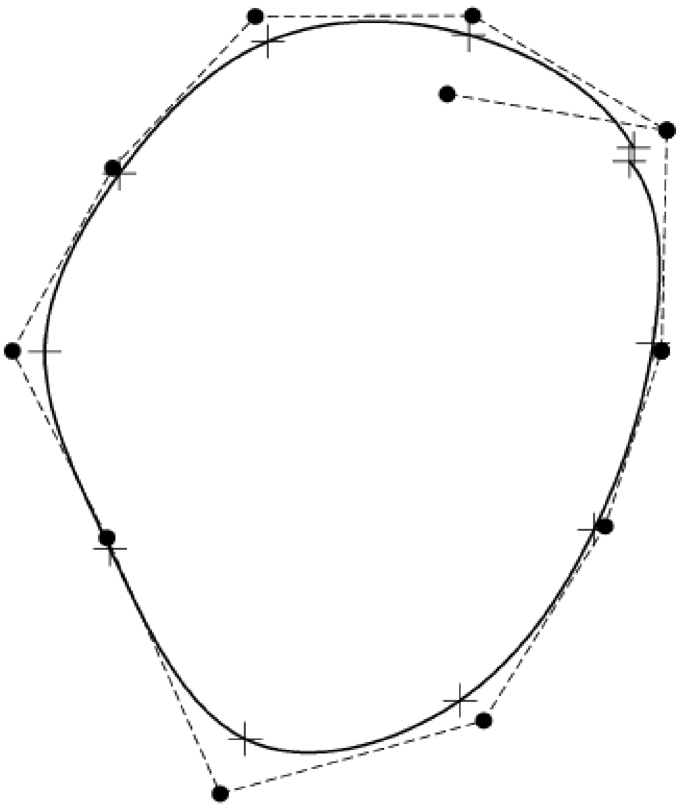
B-Splines



degree 3

-0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 1.1 1.2 1.3

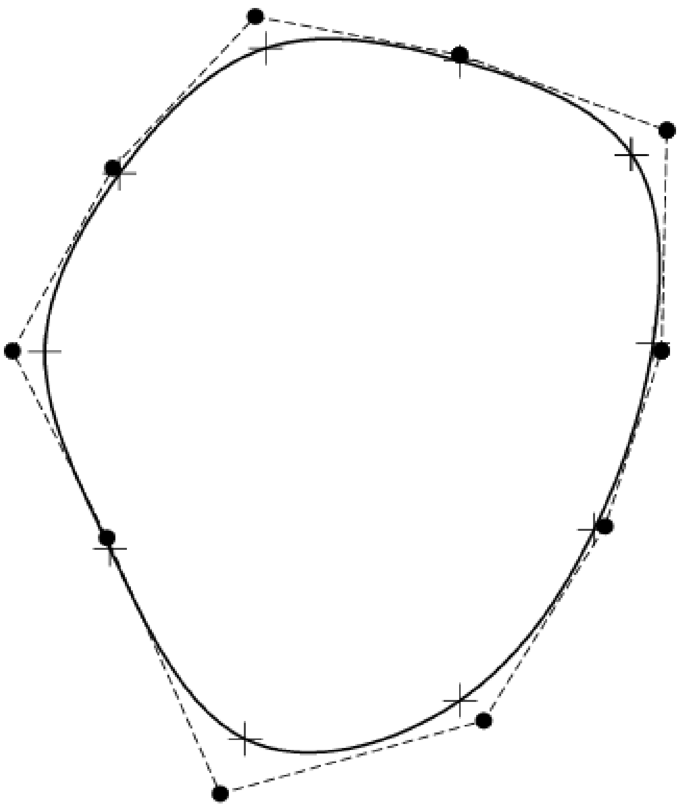
B-Splines



degree 3
-0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 1.1 1.2 1.3

B-Splines

Periodic nodal sequence
+ control points placed adequately (repeated)
= periodic curve



degree 3
-0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 1.1 1.2 1.3

- Algorithms for the manipulation of B-Splines curves
  - Boehm's knot insertion algorithm
  - Evaluation of the curve (Cox-de Boor algorithm)
  - Derivatives and hodographs
  - Restriction/growth of the useful interval of a curve
  - Degree elevation
  - Recursive Subdivision

- Boehm's knot insertion algorithm

The idea is to determine a new control polygon for the same curve after the insertion of one or several knots in the nodal sequence.

The curve is not modified by this change : neither the shape nor the parametrization are affected.

Interest :

- Evaluation of points on the curve
- Subdivision of the curve
- Addition of control points

B-Splines

- Let $P(u)=\sum_{i=0}^{n} P_i N_i^d(u)$ a B-Spline curve built on the nodal sequence : $U=\{u_0,\cdots,u_m\}$
- Let $\bar{u}\in[u_k,u_{k+1}[$ a knot to be inserted
- The new nodal sequence is :

$$\bar{U}=\{\bar{u}_0=u_0,\cdots,\bar{u}_k=u_k,\bar{u}_{k+1}=\bar{u},\cdots,\bar{u}_{m+1}=u_m\}$$

- The new representation of the curve is :

$$P(u)=\sum_{i=0}^{n+1} Q_i \bar{N}_i^d(u)$$

  - The $\bar{N}_i^d(u)$ are the basis functions defined on $\bar{U}$, the Q_i are the $n+2$ new control points.
  - How define the Q_i so that the shape is unchanged ?

B-Splines

- Let's set $P(u)=\sum_{i=0}^{n} P_i N_i^d(u)=\sum_{i=0}^{n+1} Q_i \bar{N}_i^d(u) \quad \forall u$
  - We write the relation for $n+2$ distinct values of u. $(a,b,\ldots)$
  - We obtain a band system, with $3(n+2)$ unknowns (in 3D)

$$\begin{pmatrix} \bar{N}_0^d(a) & \bar{N}_1^d(a) & \cdots \\ \bar{N}_0^d(b) & \bar{N}_1^d(b) & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} Q_0 \\ Q_1 \\ \vdots \end{pmatrix} = \begin{pmatrix} \sum_{i=0}^{n} P_i N_i^d(a) \\ \sum_{i=0}^{n} P_i N_i^d(b) \\ \vdots \end{pmatrix}$$

  - It's costly to solve and it assumes that the values $a,b\ldots$. are carefully set to avoid a singular lin. system

- Use of properties of basis functions
  - We have $\bar{u} \in [u_k, u_{k+1}[$. In this interval $[u_k, u_{k+1}[$, we have $N_i^d(u) \neq 0$ iff $i \in \{k-d, \cdots, k\}$ (compact support)
  - In the same way, $\bar{N}_i^d(u) \neq 0$ iff $i \in \{k-d, \cdots, k+1\}$ and $u \in [u_k, u_{k+1}[$. Thus, we have :

$$P(u) = \sum_{i=k-d}^{k} P_i N_i^d(u) = \sum_{i=k-d}^{k+1} Q_i \bar{N}_i^d(u)$$

$$N_i^d(u) = \bar{N}_i^d(u) \text{ for } i \in \{0, \cdots, k-d-1\}$$

$$N_i^d(u) = \bar{N}_{i+1}^d(u) \text{ for } i \in \{k+1, \cdots, n\}$$

For $i \in \{k-d, \cdots, k\}$:

$$N_i^d(u) = \frac{\bar{u} - \bar{u}_i}{\bar{u}_{i+d+1} - \bar{u}_i} \bar{N}_i^d(u) + \frac{\bar{u}_{i+d+2} - \bar{u}}{\bar{u}_{i+d+2} - \bar{u}_{i+1}} \bar{N}_{i+1}^d(u)$$

- Proof using the definition of shape functions, see Leon's book p.333

- $N_i^d(u) = \bar{N}_i^d(u)$ for $i \in \{0, \cdots, k-d-1\}$
  involves $P_i = Q_i$ for $i \in \{0, \cdots, k-d-1\}$
- $N_i^d(u) = \bar{N}_{i+1}^d(u)$ for $i \in \{k+1, \cdots, n\}$
  involves $P_i = Q_{i+1}$ for $i \in \{k+1, \cdots, n\}$
- $N_i^d(u) = \dfrac{\bar{u} - \bar{u}_i}{\bar{u}_{i+d+1} - \bar{u}_i} \bar{N}_i^d(u) + \dfrac{\bar{u}_{i+d+2} - \bar{u}}{\bar{u}_{i+d+2} - \bar{u}_{i+1}} \bar{N}_{i+1}^d(u)$

  We substitute in

$$\sum_{i=k-d}^{k} P_i N_i^d(u) = \sum_{i=k-d}^{k+1} Q_i \bar{N}_i^d(u)$$

B-Splines

$$N_i^d(u)=\frac{\bar{u}-\bar{u}_i}{\bar{u}_{i+d+1}-\bar{u}_i}\bar{N}_i^d(u)+\frac{\bar{u}_{i+d+2}-\bar{u}}{\bar{u}_{i+d+2}-\bar{u}_{i+1}}\bar{N}_{i+1}^d(u) \quad \text{in} \quad \sum_{i=k-d}^{k} P_i N_i^d(u)=\sum_{i=k-d}^{k+1} Q_i \bar{N}_i^d(u)$$

$$\left(\frac{\bar{u}-\bar{u}_{k-d}}{\bar{u}_{k+1}-\bar{u}_{k-d}}\bar{N}_{k-d}^d(u)+\frac{\bar{u}_{k+2}-\bar{u}}{\bar{u}_{k+2}-\bar{u}_{k-d+1}}\bar{N}_{k-d+1}^d(u)\right)P_{k-d}$$

$$+\left(\frac{\bar{u}-\bar{u}_{k-d+1}}{\bar{u}_{k+2}-\bar{u}_{k-d+1}}\bar{N}_{k-d+1}^d(u)+\frac{\bar{u}_{k+3}-\bar{u}}{\bar{u}_{k+3}-\bar{u}_{k-d+2}}\bar{N}_{k-d+2}^d(u)\right)P_{k-d+1}$$

$$+\cdots$$

$$+\left(\frac{\bar{u}-\bar{u}_{k}}{\bar{u}_{k+d+1}-\bar{u}_{k}}\bar{N}_{k}^d(u)+\frac{\bar{u}_{k+d+2}-\bar{u}}{\bar{u}_{k+d+2}-\bar{u}_{k+1}}\bar{N}_{k+1}^d(u)\right)P_{k}$$

$$=\bar{N}_{k-d}^d(u)\,Q_{k-d}+\cdots+\bar{N}_{k+1}^d(u)\,Q_{k+1}$$

- By factoring and replacing the nodal sequence $\bar{U}$ by U , we obtain :

$$0=\bar{N}^{d}_{k-d}(u)\left(Q_{k-d}-P_{k-d}\right)$$

$$+\bar{N}^{d}_{k-d+1}(u)\left(Q_{k-d+1}-\frac{\bar{u}-u_{k-d+1}}{u_{k+1}-u_{k-d+1}}P_{k-d+1}-\frac{u_{k+1}-\bar{u}}{u_{k+1}-u_{k-d+1}}P_{k-d}\right)$$

$$\vdots$$

$$+\bar{N}^{d}_{k}(u)\left(Q_{k}-\frac{\bar{u}-u_{k}}{u_{k+d}-u_{k}}P_{k}-\frac{u_{k+d}-\bar{u}}{u_{k+d}-u_{k}}P_{k-1}\right)+\bar{N}^{d}_{k+1}(u)\left(Q_{k+1}-P_{k}\right)$$

- We set $\alpha_i=\dfrac{\bar{u}-u_i}{u_{i+d}-u_i}$

$$1-\alpha_i=\frac{u_{i+d}-\bar{u}}{u_{i+d}-u_i}$$ with $i\in\{k-d+1,\cdots,k\}$

…

... we obtain

$$Q_{k-d}=P_{k-d}$$
$$Q_i=\alpha_i P_i+(1-\alpha_i)P_{i-1} \text{ for } i\in\{k-d+1,\cdots,k\}$$
$$Q_{k+1}=P_k$$

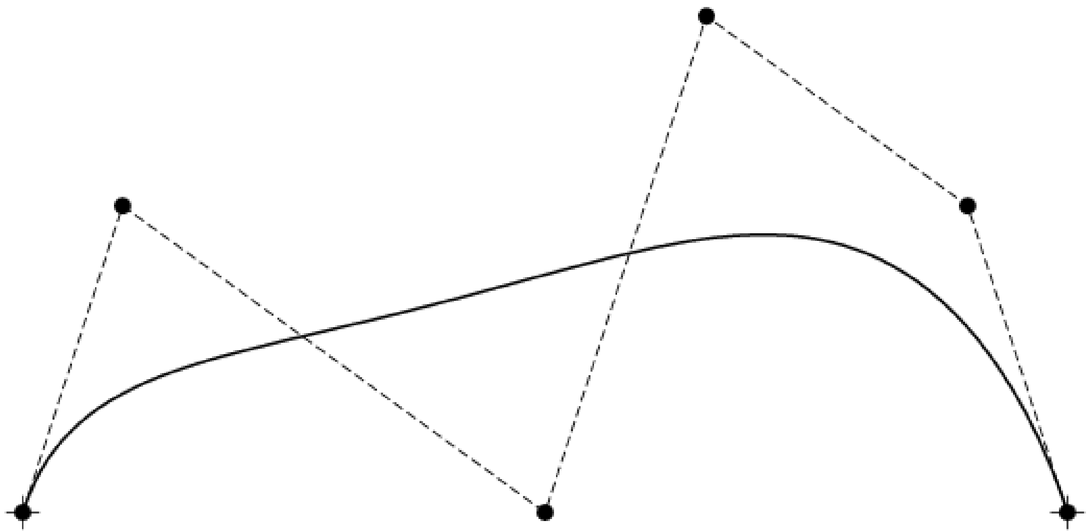
We had :

$$P_i=Q_i \text{ for } i\in\{0,\cdots,k-d-1\}$$
$$P_i=Q_{i+1} \text{ for } i\in\{k+1,\cdots,n\}$$

so finally :

$$Q_i=\alpha_i P_i+(1-\alpha_i)P_{i-1} \quad \text{with } \alpha_i=\begin{cases} 1 & i\leq k-d \\ \dfrac{\bar{u}-u_i}{u_{i+p}-u_i} & k-d+1\leq i\leq k \\ 0 & i\geq k+1 \end{cases}$$

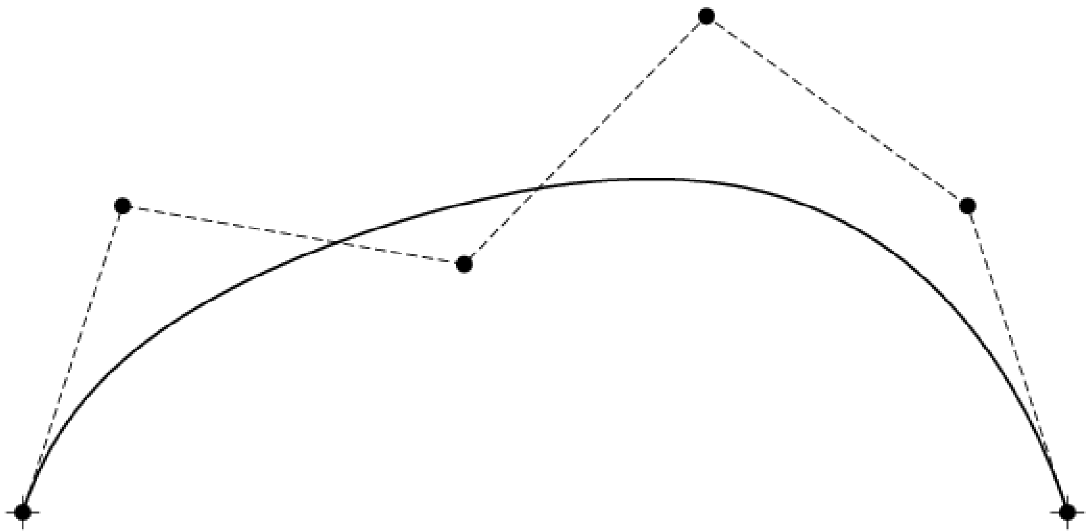
B-Splines



$$U = \{0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1\}$$

degree 5

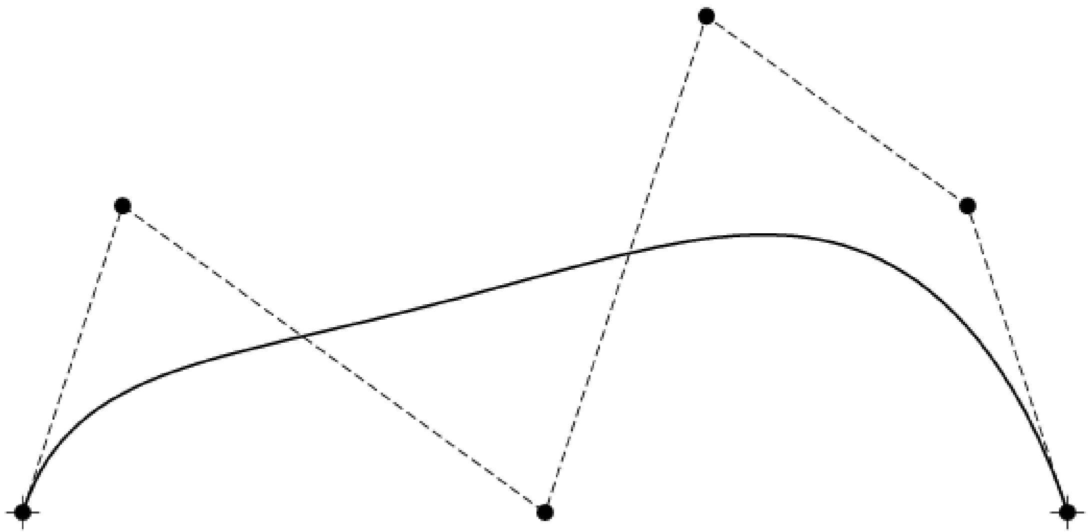
B-Splines



$$U = \{0,0,0,0,0,0,1,1,1,1,1,1\}$$

degree 5

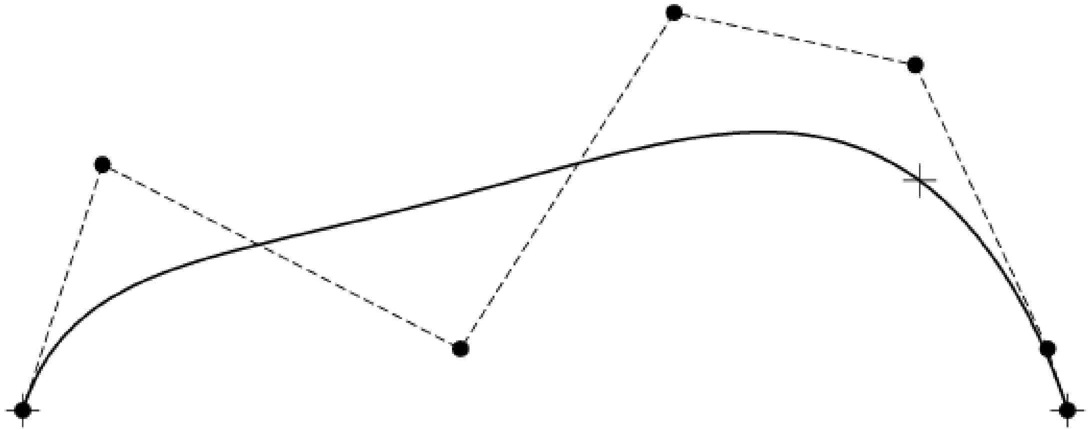
B-Splines



$$U = \{0\,,0\,,0\,,0\,,0\,,0,1\,,1\,,1\,,1\,,1\,,1\}$$

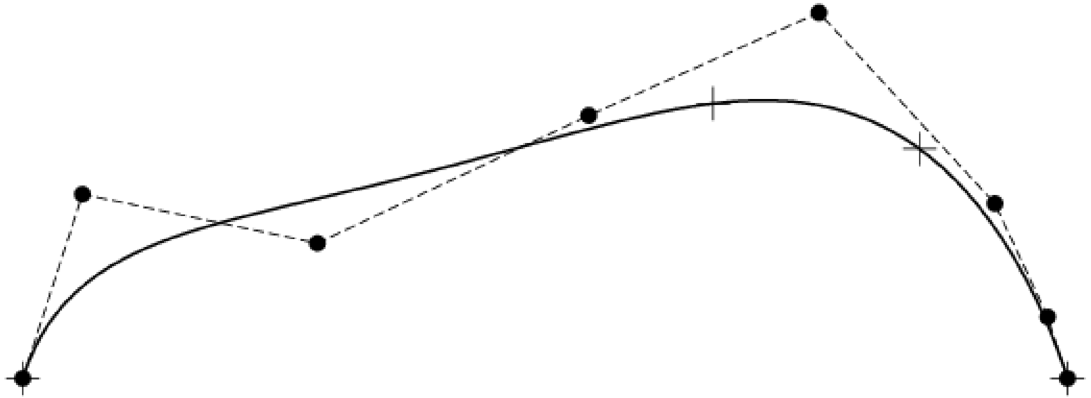
degree 5

B-Splines



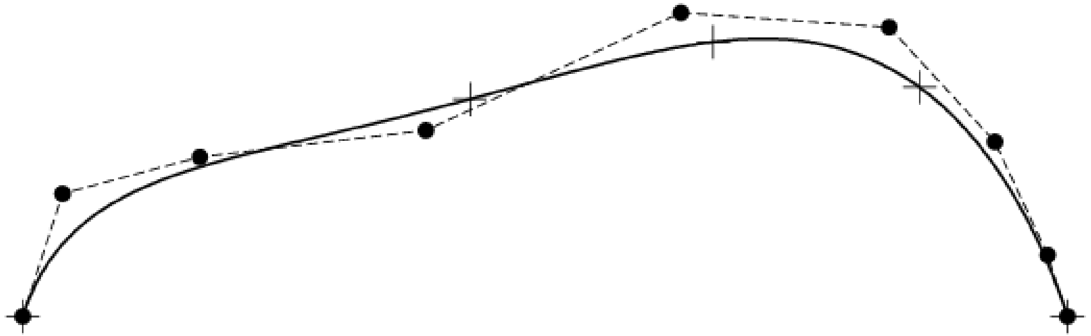
$$U = \{0, 0, 0, 0, 0, 0, 0.2, 1, 1, 1, 1, 1, 1\}$$

degree 5



$$U=\{0,0,0,0,0,0,0.2,0.4,1,1,1,1,1,1\}$$

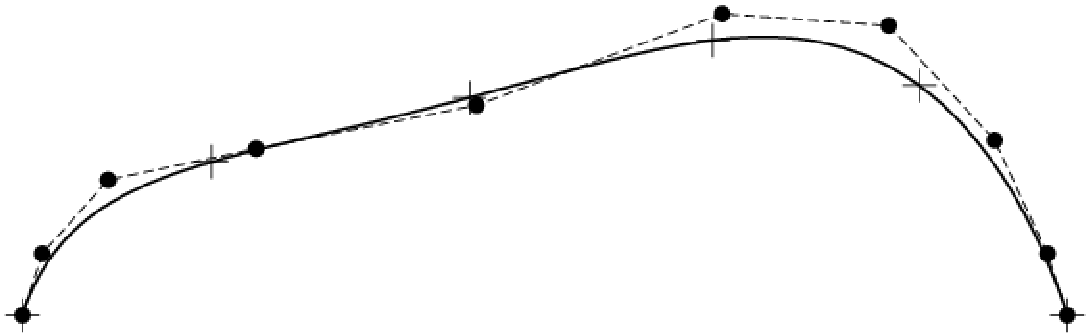
degree 5



$$U=\{0,0,0,0,0,0,0.2,0.4,0.6,1,1,1,1,1,1\}$$

degree 5

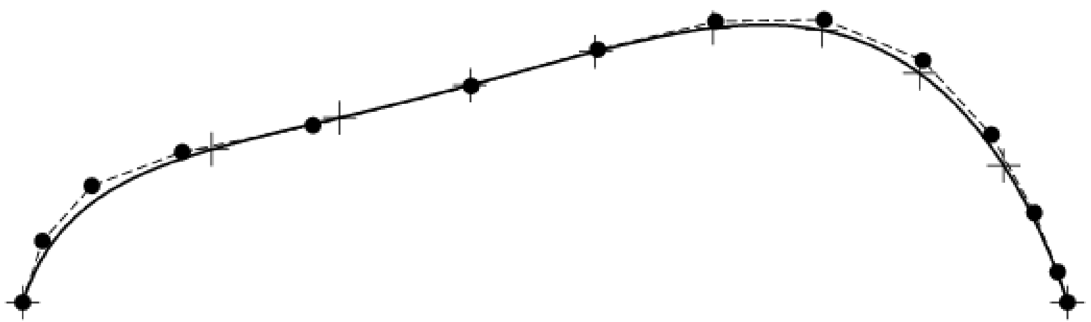
B-Splines



$$U = \{0,0,0,0,0,0,0.2,0.4,0.6,0.8,1,1,1,1,1,1\}$$

degree 5

B-Splines

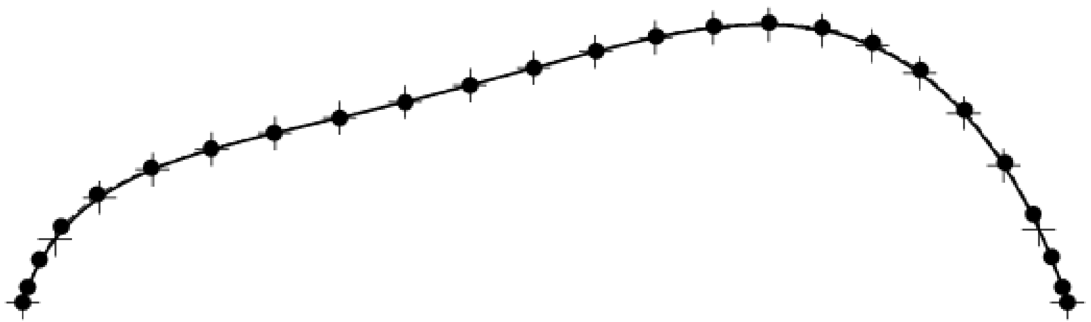


$$U = \{0,0,0,0,0,0,0.1,0.2,\cdots,0.9,1,1,1,1,1,1\}$$

degree 5

Computer Aided Design

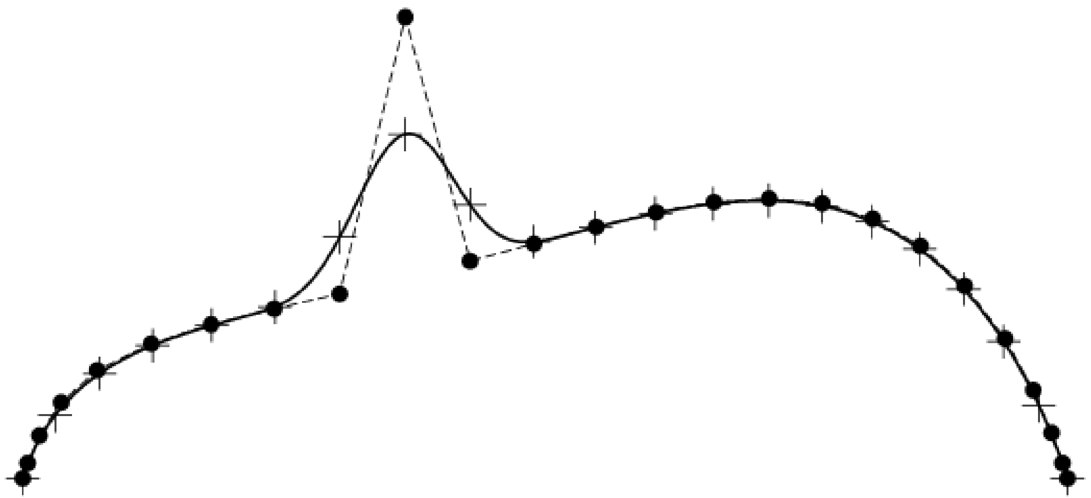
B-Splines



$$U = \{0\,,0\,,0\,,0\,,0\,,0\,,0.05\,,0.1\,,\cdots,0.95\,,1\,,1\,,1\,,1\,,1\,,1\}$$

degree 5

B-Splines



$$U = \{0, 0, 0, 0, 0, 0, 0.05, 0.1, \cdots, 0.95, 1, 1, 1, 1, 1, 1\}$$

degree 5

Local control...

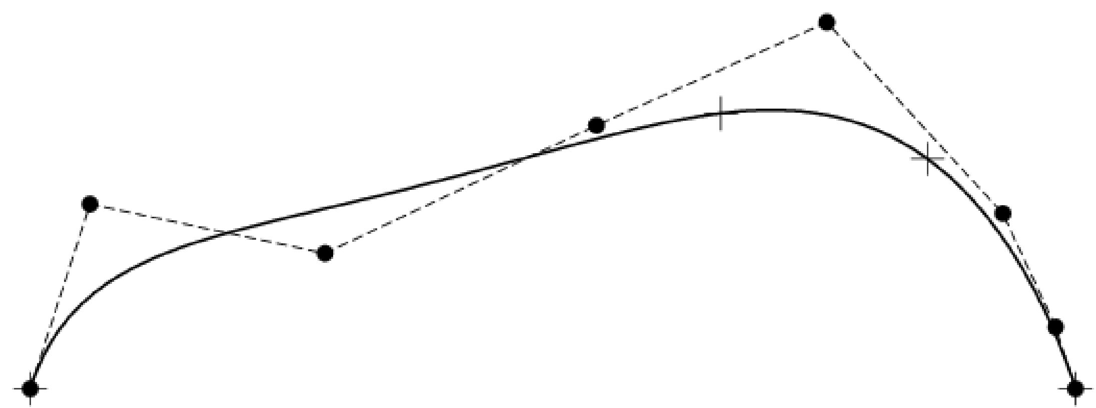
Computer Aided Design

B-Splines

$$N_i^d = (-1)^{d+1}(u_{i+d+1} - u_i)[u_i, \cdots, u_{i+d+1}]_U (u - U)_+^d$$

$$N_i^0(u) = \begin{cases} 1 & \text{if} \quad u_i \le u < u_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_i^d(u) = \frac{u - u_i}{u_{i+d} - u_i} N_i^{d-1}(u) + \frac{u_{i+d+1} - u}{u_{i+d+1} - u_{i+1}} N_{i+1}^{d-1}(u)$$



$$U = \{0, 0, 0, 0, 0, 0, 0.2, 0.4, 1, 1, 1, 1, 1, 1\}$$

degree 5

Boehm's knot insertion formula :

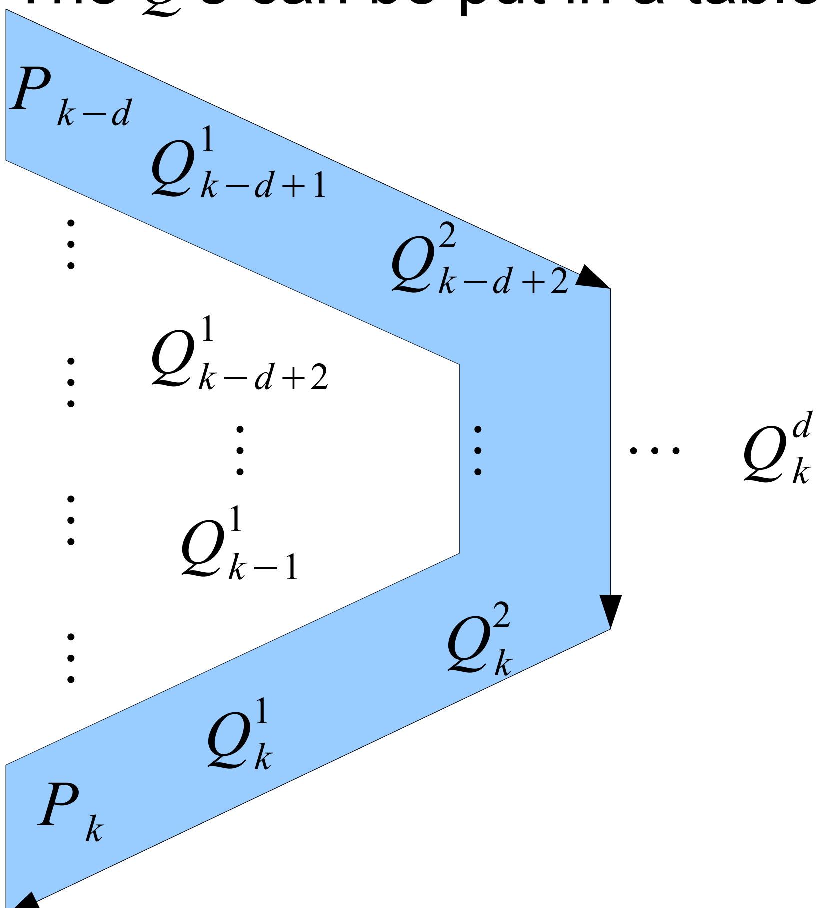
$$Q_i = \alpha_i P_i + (1-\alpha_i) P_{i-1} \quad \text{with } \alpha_i = \begin{cases} 1 & i \le k-d \\ \dfrac{\bar{u} - u_i}{u_{i+p} - u_i} & k-d+1 \le i \le k \\ 0 & i \ge k+1 \end{cases}$$

- Multiple knot insertions
  - Assume $\bar{u} \in [u_k, u_{k+1}[$ of multiplicity s $(0 \leq s < d)$. We want to insert it r times with $r + s \leq d$.
  - We note Q_i^r the control points of the r-th insertion step
  - We have then :

$$Q_i^r = \alpha_i^r Q_i^{r-1} + \left(1 - \alpha_i^r\right) Q_{i-1}^{r-1} \quad \text{with } \alpha_i^r = \begin{cases} 1 & i \leq k - d + r - 1 \\ \dfrac{\bar{u} - u_i}{u_{i+d-r+1} - u_i} & k - d + r \leq i \leq k - s \\ 0 & i \geq k - s + 1 \end{cases}$$

B-Splines

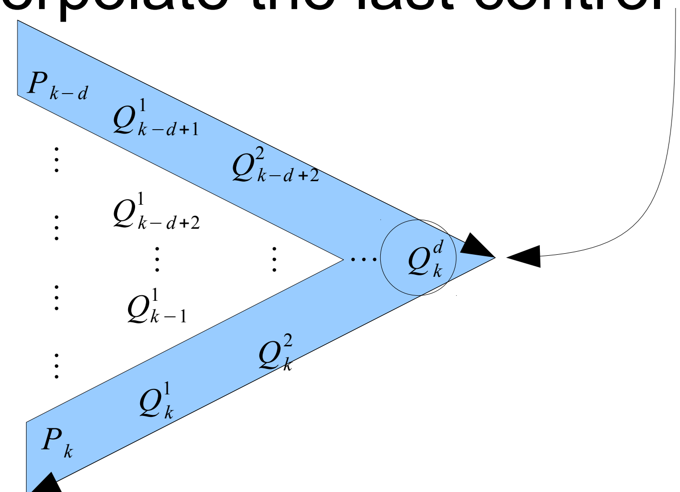
- The Q's can be put in a table:



- The total number of new control points is d-s+r-1 that replace d-s-1.

B-Splines

- The use of the algorithm of node insertion up to multiplicity of $d=r+s$ is such that the curve will interpolate the last control point that is computed.



- Therefore, one can use this algorithm to compute the position of a point of the curve knowing the parameter.
  - It's precisely the Cox-de Boor algorithm. The sequence of points P_i^j is not anything else than the Q_i^j indicated on the graph, cf following

B-Splines

  - Case $r+s=d+1$: We carry out the insertion of multiplicity r-1 then we insert one more knot to « cut » the B-spline curve in two independent parts.
  - The last control point Q_k^d has to be duplicated.
  - Allows to extract a portion of the B-spline.
- There exists an extension of this algorithm in the case of the simultaneous insertion of many knots: it is the somewhat more complex “Oslo” algorithm* , not described here.

* E. Cohen, T. Lyche, R. Riesenfeld “Discrete B-splines and subdivision techniques in computer-aided geometric design and computer graphics”, *Computer Graphics and Image Processing*, **14**(2):87-111, 1980.

- (simplified) Cox-de Boor Algorithm :

Determine the interval of u : $u \in [u_i , u_{i+1} [$

Initialization of P_j^0 $i \in \{d , d+1, \cdots, m-d-1\}$

For k from 1 to d

For j from i-d+k to i

$$P_j^k = \left(\frac{u - u_j}{u_{j+d+1-k} - u_j} \right) P_j^{k-1} + \left(\frac{u_{j+d+1-k} - u}{u_{j+d+1-k} - u_j} \right) P_{j-1}^{k-1}$$

Endfor

Endfor

P_i^d is the point that is sought.

- What is its complexity ?
  - quadratic in function of the degree d.

- On a knot of multiplicity s :

Determine the interval of u : $u \in [u_{i-s} = \cdots = u_i, u_{i+1}[$

Initialization of P_j^0

and $u = u_i$

$i \in \{d, d+1, \cdots, m-d-1\}$

For k from 1 to d-s

For j from i-d+k to i-s

$$P_j^k = \left(\frac{u - u_j}{u_{j+d+1-k} - u_j} \right) P_j^{k-1} + \left(\frac{u_{j+d+1-k} - u}{u_{j+d+1-k} - u_j} \right) P_{j-1}^{k-1}$$

Endfor

Endfor

P_{i-s}^{d-s} is the point that is sought.

- Example of computation

$$P_0^0=(0,1)\quad P_1^0=(2,3)\quad P_2^0=(5,4)\quad P_3^0=(7,1)\quad P_4^0=(6,-1)\quad P_5^0=(6,-2)$$

$$U=\{0,0,0,0,1,2,3,3,3,3\}\qquad d=3\qquad u=3/2$$

  - Determination of the interval

$$1\leq 3/2<2\quad ,u_4=1\ \rightarrow\ i=4$$

$$P_j^k=\left(\frac{u-u_j}{u_{j+d+1-k}-u_j}\right)P_j^{k-1}+\left(\frac{u_{j+d+1-k}-u}{u_{j+d+1-k}-u_j}\right)P_{j-1}^{k-1}$$

  - Iteration 1

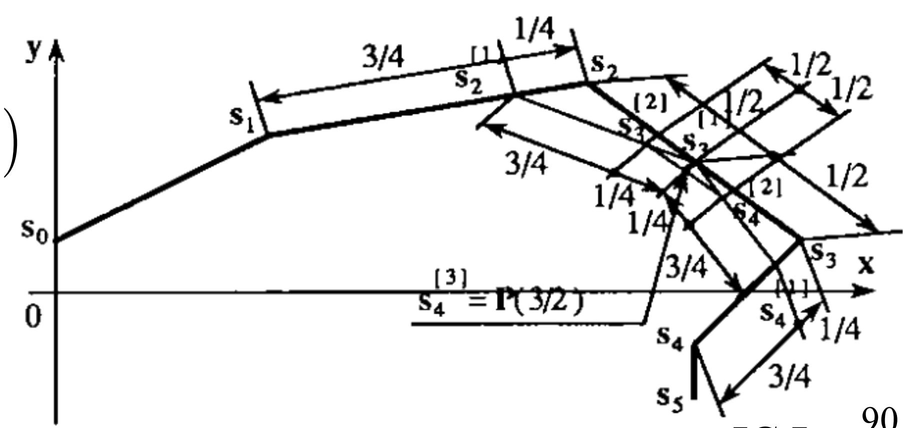
$$P_4^1=(27/4,1/2)\quad P_3^1=(6,5/2)\quad P_2^1=(17/4,15/5)$$

  - Iteration 2

$$P_4^2=(99/16,2)\quad P_3^2=(89/16,45/16)$$

  - Iteration 3

$$P_4^3=(47/8,77/32)=P(3/2)$$

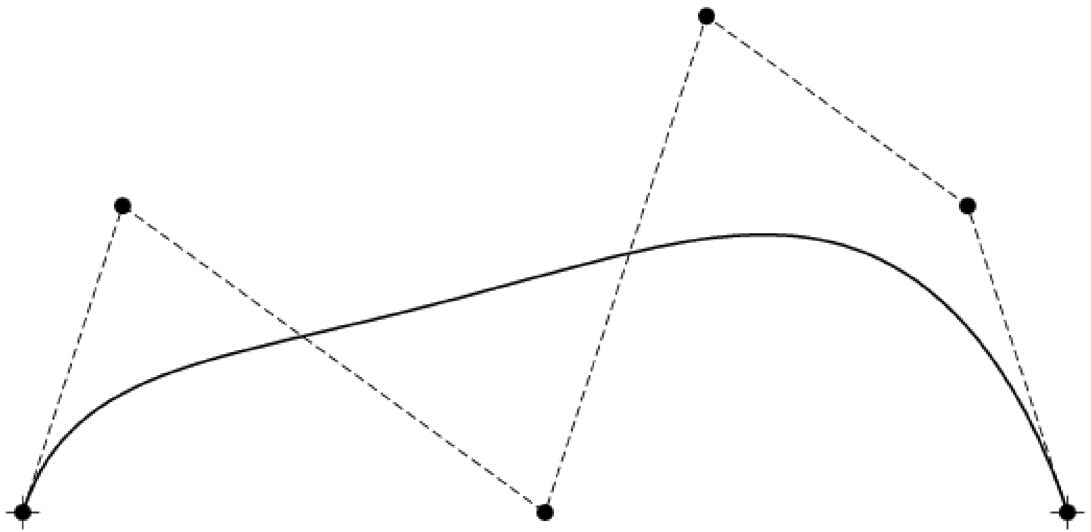


JC Leon

B-Splines

- The algorithm is similar to De Casteljau's algorithm for Bézier curves
  - It is built on a restriction of the set of control points ($d+1$ points)
  - On this restriction, it is nearly identical, except for the coefficients related to the nodal sequence (which is potentially **non uniform**)
  - The complete algorithm is somewhat longer than this one (possibility to have $0/0$: we set conventionally $0/0 = 0$!)

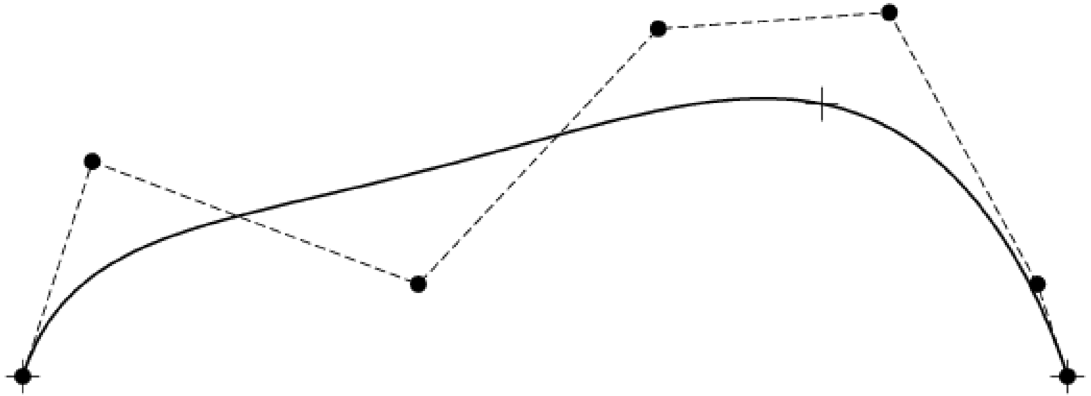
B-Splines



$$U = \{0,0,0,0,0,0,1,1,1,1,1,1\}$$

degree 5

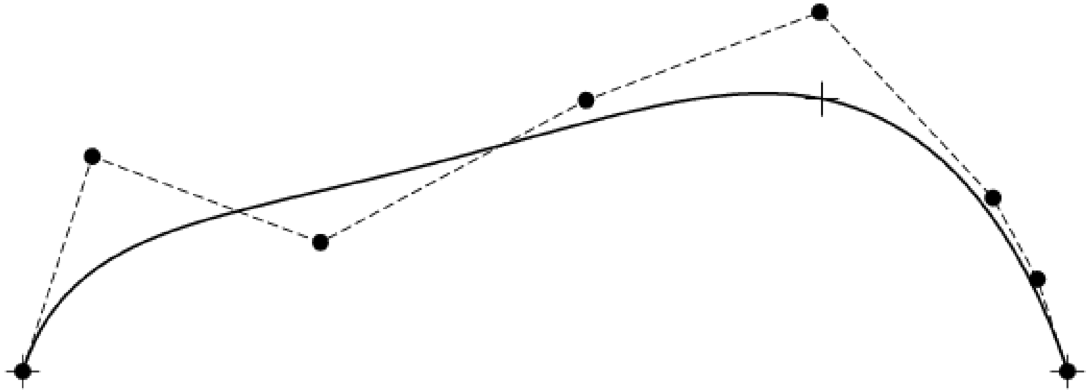
B-Splines



$$U = \{0, 0, 0, 0, 0, 0, 0.3, 1, 1, 1, 1, 1, 1\}$$

degree 5

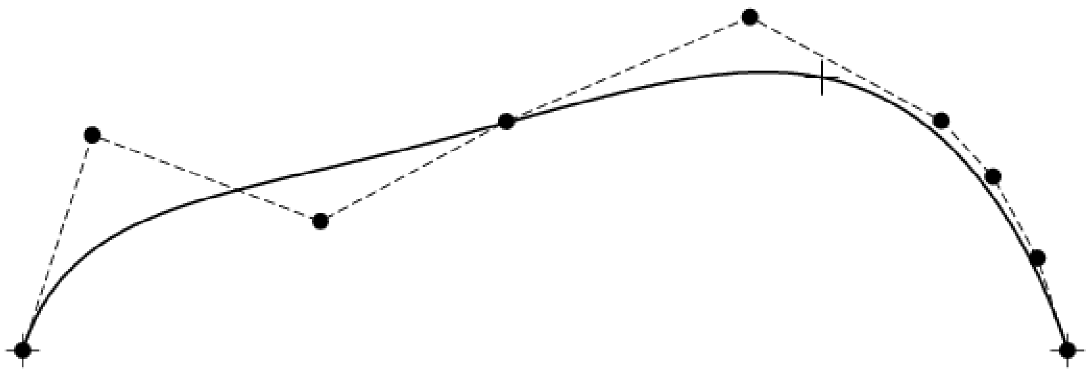
B-Splines



$$U = \{0, 0, 0, 0, 0, 0, 0.3, 0.3, 1, 1, 1, 1, 1, 1\}$$

degree 5

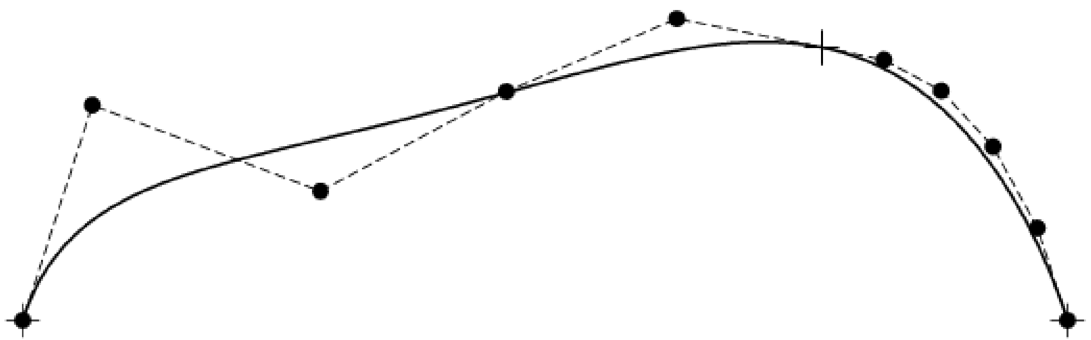
B-Splines



$$U=\{0,0,0,0,0,0,0.3,0.3,0.3,1,1,1,1,1,1\}$$

degree 5

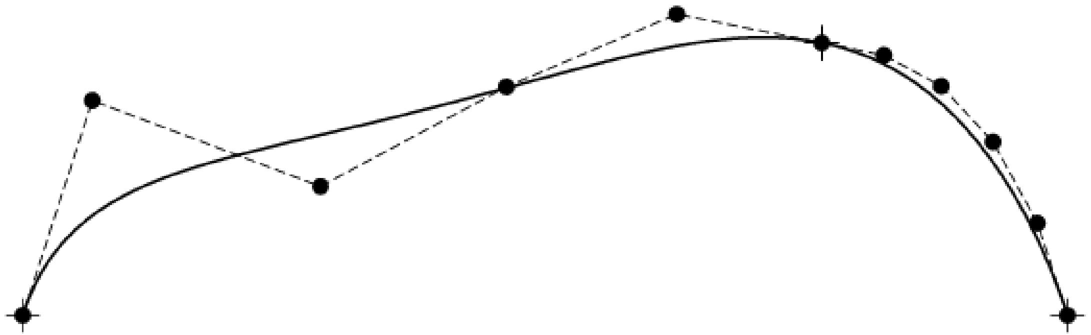
B-Splines



$$U = \{0,0,0,0,0,0,0.3,0.3,0.3,0.3,1,1,1,1,1,1\}$$

degree 5

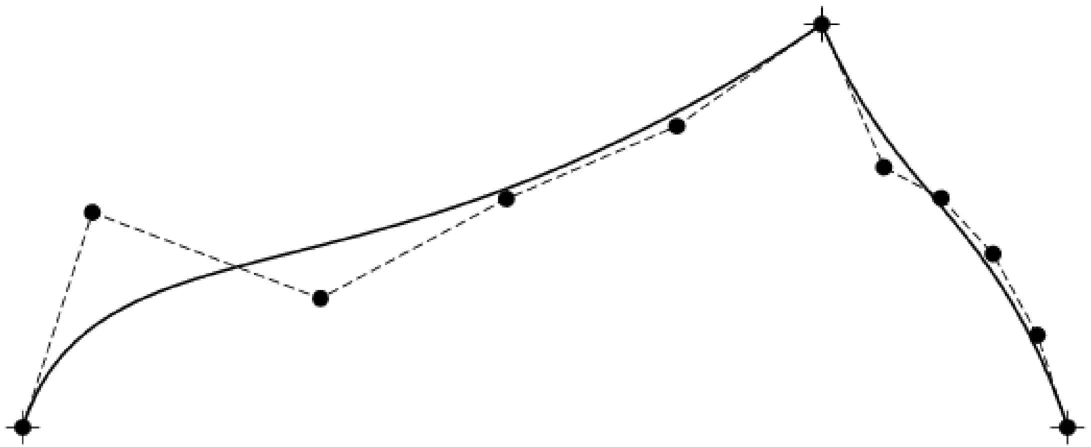
B-Splines



$$U = \{0, 0, 0, 0, 0, 0, 0.3, 0.3, 0.3, 0.3, 0.3, 1, 1, 1, 1, 1, 1\}$$

degree 5

B-Splines



$$U=\{0,0,0,0,0,0,0.3,0.3,0.3,0.3,0.3,1,1,1,1,1,1\}$$

degree 5

- Computation of a point on a B-Spline curve
  - By the use of basis functions

  1 – Find the nodal interval in which u is located

  $$u \in [u_i, u_{i+1}[$$

  2 – Calculate the non vanishing basis functions

  $$N_{i-d}^{d}(u), \cdots, N_i^{d}(u)$$

  3 – Multiply the values of these basis functions with the right control points

  $$P(u) = \sum_k N_k^d(u) P_k \qquad i-d \leq k \leq i$$

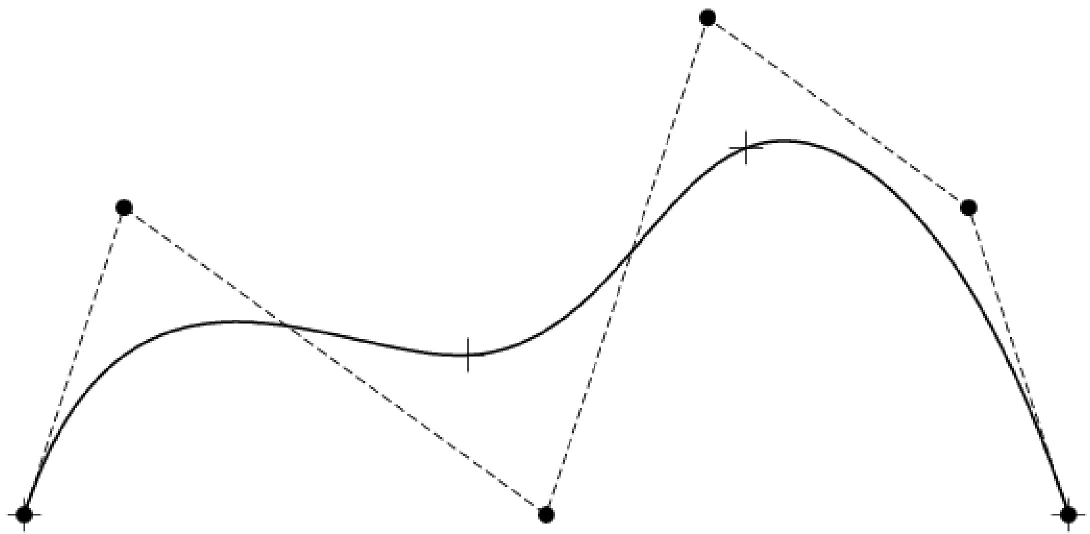
  - By the algorithm of Cox-de Boor

- Transformation of a B-Spline curve into a composite Bézier curve
  - We saturate each distinct knot until its multiplicity is equal to d.
  - This is made with the help of Boehm's algorithm of nodal insertion.
    - The curve is not modified !
  - We obtain a nodal sequence which has the following form :

$$U=\{\underbrace{a,a,a,a}_{d+1\ \text{times}},\underbrace{b,b,b}_{d\ \text{times}},\underbrace{c,c,c}_{d\ \text{times}},\cdots,\underbrace{z,z,z,z}_{d+1\ \text{times}}\}$$

    - Each distinct value of u corresponds to one of the points of the curve.

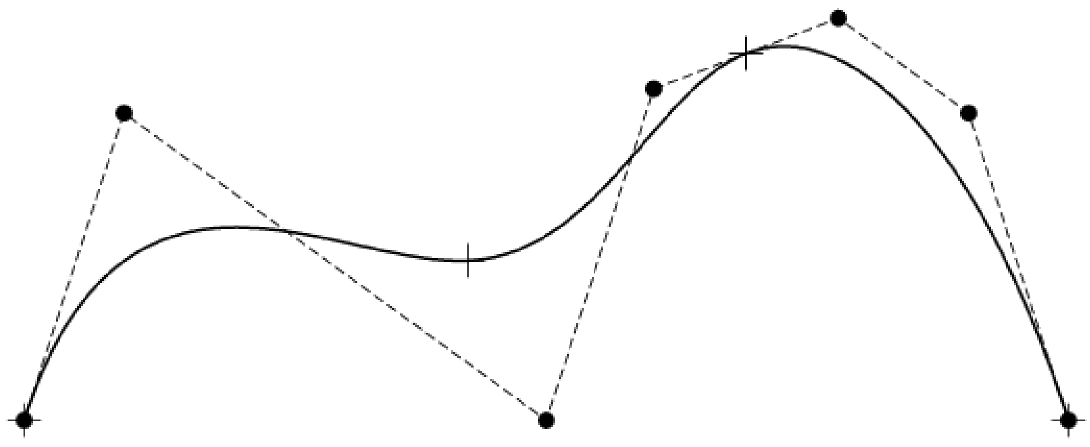
B-Splines



$$U = \{0, 0, 0, 0, 1, 2, 3, 3, 3, 3\}$$

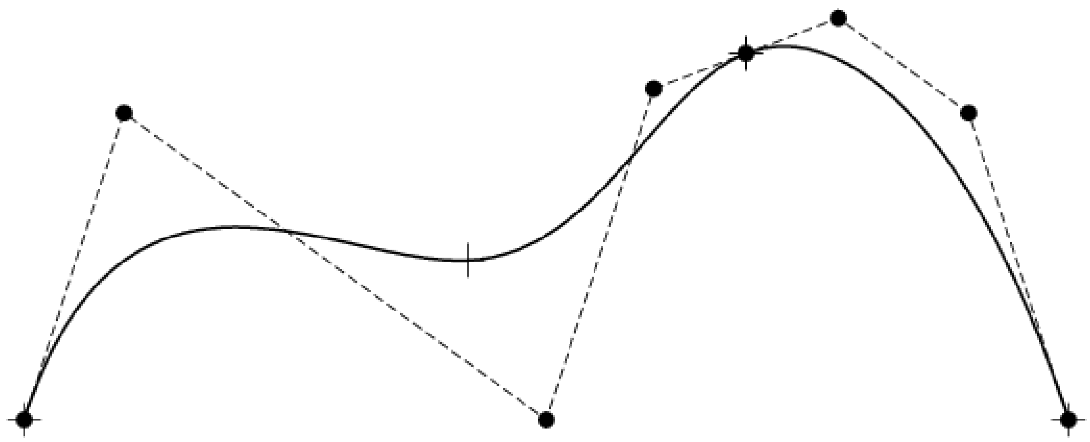
degree 3

B-Splines



$$U = \{0, 0, 0, 0, 1, 1, 2, 3, 3, 3, 3\}$$

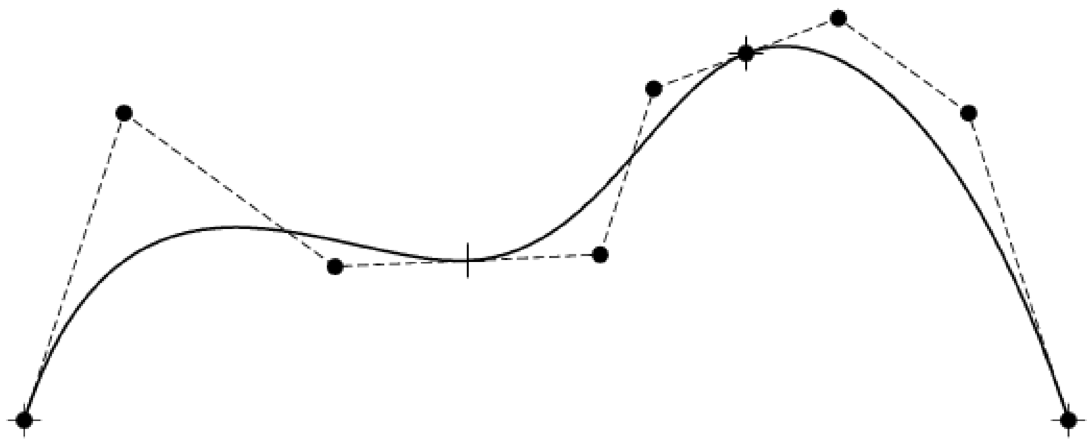
B-Splines



$$U = \{0, 0, 0, 0, 1, 1, 1, 2, 3, 3, 3, 3\}$$

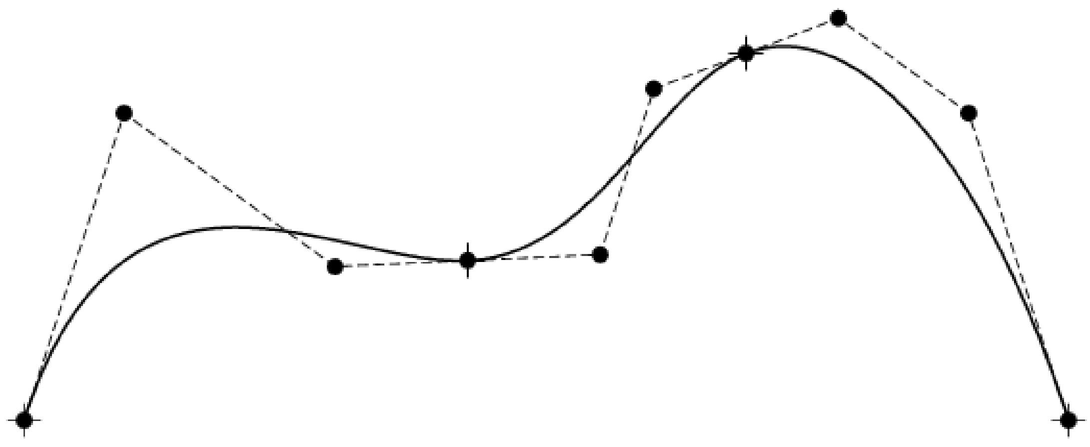
Computer Aided Design

B-Splines



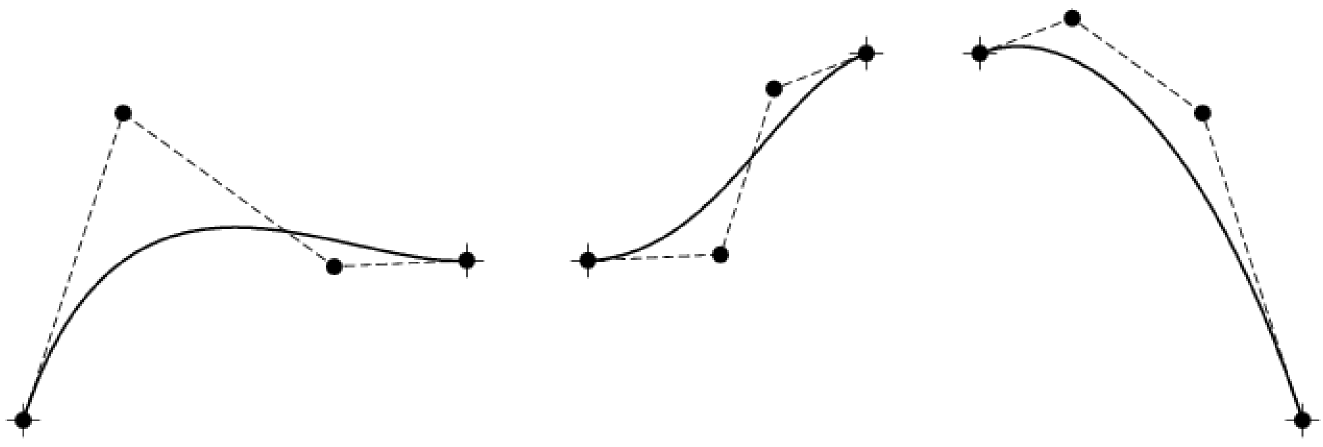
$$U = \{0, 0, 0, 0, 1, 1, 1, 2, 2, 3, 3, 3, 3\}$$

B-Splines



$$U = \{0,0,0,0,1,1,1,2,2,2,3,3,3,3\}$$

B-Splines



Bézier n°3
4 CP
degree 3

Bézier n°2
4 CP
degree 3

Bézier n°1
4 CP
degree 3

- Differentiating B-splines
  - For a B-Spline of typical nodal sequence:
    $$U=\{\underbrace{u_0,\cdots,u_d}_{d+1\ \text{times}},\cdots,\underbrace{u_{m-d},\cdots,u_m}_{d+1\ \text{times}}\}$$
  - We start from the definition of the derivatives of the basis functions:
    $$N_i^{d\,'}=\frac{d}{u_{i+d}-u_i}N_i^{d-1}(u)-\frac{d}{u_{i+d+1}-u_{i+1}}N_{i+1}^{d-1}(u)$$
    - Proof : Starting from N^1 and verify by induction that the relation is true for N^d if it is true for $N^{d\text{-}1}$.
  - At the order k :
    $$\frac{\partial^k N_i^d}{\partial u^k}=\frac{d}{u_{i+d}-u_i}\frac{\partial^{k-1}N_i^{d-1}(u)}{\partial u^{k-1}}-\frac{d}{u_{i+d+1}-u_{i+1}}\frac{\partial^{k-1}N_{i+1}^{d-1}(u)}{\partial u^{k-1}}$$

B-Splines

$$N_i^{d\,'} = \frac{d}{u_{i+d} - u_i} N_i^{d-1}(u) - \frac{d}{u_{i+d+1} - u_{i+1}} N_{i+1}^{d-1}(u)$$

$$P'(u) = \sum_{i=0}^{n} P_i N_i^{d\,'}(u)$$

$$P'(u) = \sum_{i=0}^{n} \left(\frac{d}{u_{i+d} - u_i} N_i^{d-1}(u) - \frac{d}{u_{i+d+1} - u_{i+1}} N_{i+1}^{d-1}(u) \right) P_i$$

$$P'(u) = d \sum_{i=-1}^{n-1} N_{i+1}^{d-1}(u) \frac{P_{i+1}}{u_{i+d+1} - u_{i+1}} - d \sum_{i=0}^{n} N_{i+1}^{d-1}(u) \frac{P_i}{u_{i+d+1} - u_{i+1}}$$

Change of index

B-Splines

$$P'(u) = d\sum_{i=-1}^{n-1} N_{i+1}^{d-1}(u)\frac{P_{i+1}}{u_{i+d+1}-u_{i+1}} - d\sum_{i=0}^{n} N_{i+1}^{d-1}(u)\frac{P_i}{u_{i+d+1}-u_{i+1}}$$

$$P'(u) = d\sum_{i=0}^{n-1} N_{i+1}^{d-1}(u)\frac{P_{i+1}-P_i}{u_{i+d+1}-u_{i+1}} + d\left(\frac{P_0 \overbrace{\boxed{N_0^{d-1}(u)}}^{0}}{\underbrace{\boxed{u_d-u_0}}_{\text{Empty Interval}}} - \frac{P_{n+1}\overbrace{\boxed{N_{n+1}^{d-1}(u)}}^{0}}{\underbrace{\boxed{u_{n+d+1}-u_{n+1}}}_{\text{Empty Interval}}}\right)$$

$$P'(u) = \sum_{i=0}^{n-1} N_{i+1}^{d-1}(u) Q_i \quad \text{with} \quad Q_i = d\frac{P_{i+1}-P_i}{u_{i+d+1}-u_{i+1}}$$

$$P'(u)=\sum_{i=0}^{n-1} N_{i+1}^{d-1}(u)Q_i \quad \text{with} \quad Q_i=d\,\frac{P_{i+1}-P_i}{u_{i+d+1}-u_{i+1}}$$

- the nodal sequence used here remains :

$$U=\{\underbrace{u_0,\cdots,u_d}_{d+1 \text{ times}},\cdots,\underbrace{u_{m-d},\cdots,u_m}_{d+1 \text{ times}}\}$$

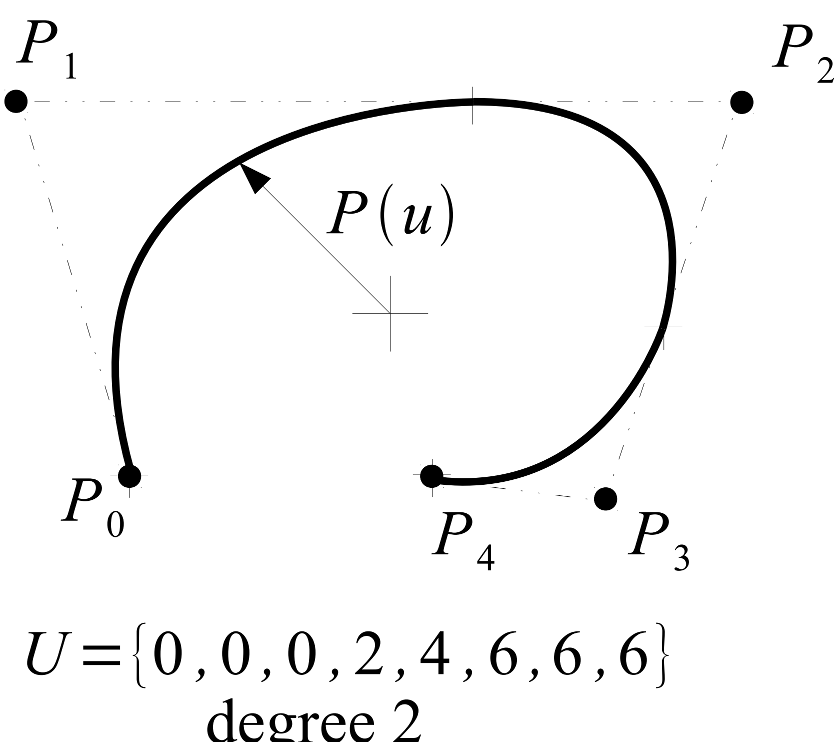
- We set

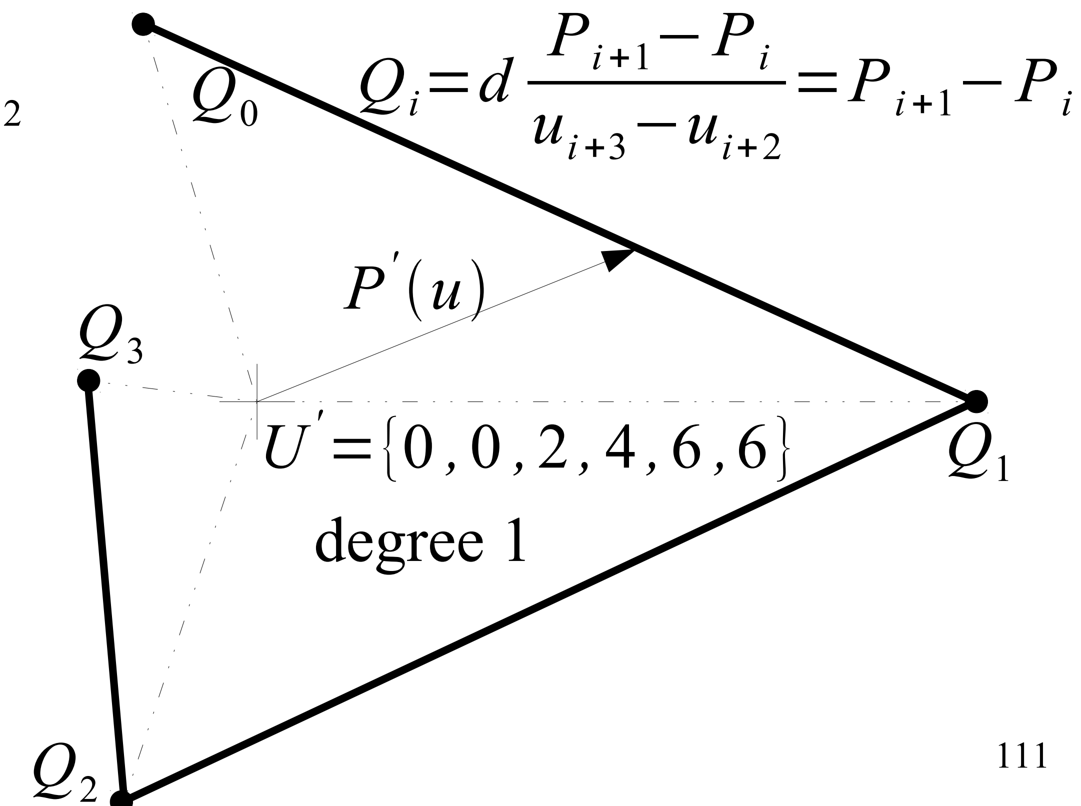
$$U'=\{\underbrace{u'_0,\cdots,u'_{d-1}}_{d \text{ times}},\cdots,\underbrace{u'_{m-d},\cdots,u'_{m-2}}_{d \text{ times}}\} \text{ with } u'_i=u_{i+1}$$

$$P'(u)=\sum_{i=0}^{n-1} N_i^{d-1}(u)Q_i \quad \text{with} \quad N_i^{d-1} \text{ defined on } U'$$

- We've got a new B-Spline !

- The 1st hodograph of a B-Spline of degree d is a B-Spline of degree d-1.
- By successive derivation, the same holds for the k^{th} hodograph.





- Derivative of order k

$$P^{(k)}(u)=\sum_{i=0}^{n-k} N_i^{d-k}(u)\,P_i^{(k)} \quad \text{with} \quad P_i^{(k)}=\begin{cases} P_i & k=0 \\ \dfrac{(d-k+1)\left(P_{i+1}^{(k-1)}-P_i^{(k-1)}\right)}{u_{i+d+1}-u_{i+k}} & k>0 \end{cases}$$

$$U^{(k)}=\{\underbrace{u_0^{(k)},\cdots,u_{d-k}^{(k)}}_{d+1-k\ \text{times}},\cdots,\underbrace{u_{m-d-k+1}^{(k)},\cdots,u_{m-2k}^{(k)}}_{d+1-k\ \text{times}}\} \quad \text{with} \quad u_i^{(k)}=u_{i+k}$$

- Knot removal
  - Inverse of Boehm's algorithm
  - Given here without demonstration – the manipulations leading to this result are simple but tedious
    - Cf JC Leon «Modelisation de courbes et surfaces pour la CFAO» , hermès (1991) - pp352-353

- Knot removal
  - At each node, the continuity is at least C^{d-s} where s is the multiplicity of the node.
    - A curve of degree 3 has a continuity C^2 if the nodes are simple (multiplicity 1).
  - In the case where the *effective* continuity C^r of the curve at a given node is higher than C^{d-s} , one can decrease the multiplicity of the node by $t=d-s-r$.
    - If it is not the case, one cannot remove the node without changing the **shape** of the curve.
  - Before effectively removing a node, one must check the continuity of the curve on both sides of the node. The next algorithm allows to decide this while computing the new control points.

- We try to delete the knot $u=u_r \neq u_{r+1}$ of multiplicity s.

$i = r - d \,;\, j = r - s$

while ($j - i > 0$)

compute $\alpha_i = \dfrac{u - u_i}{u_{i+d+1} - u_i}$ $\qquad \alpha_j = \dfrac{u - u_j}{u_{j+d+1} - u_j}$

compute $P_i^1 = \dfrac{P_i^0 - (1-\alpha_i) P_{i-1}^1}{\alpha_i}$ $\qquad P_j^1 = \dfrac{P_j^0 - \alpha_j P_{j+1}^1}{1-\alpha_j}$

$i = i + 1 \,;\, j = j - 1$

end while

These two terms are known for the first iteration they are respectively P^0_{r-d-1} and P^0_{r-s+1} .

- Testing the possibility to remove the knot

if $(i=j)$

compute $P=\alpha_i P^1_{i-1}+(1-\alpha_i)P^1_{i+1}$

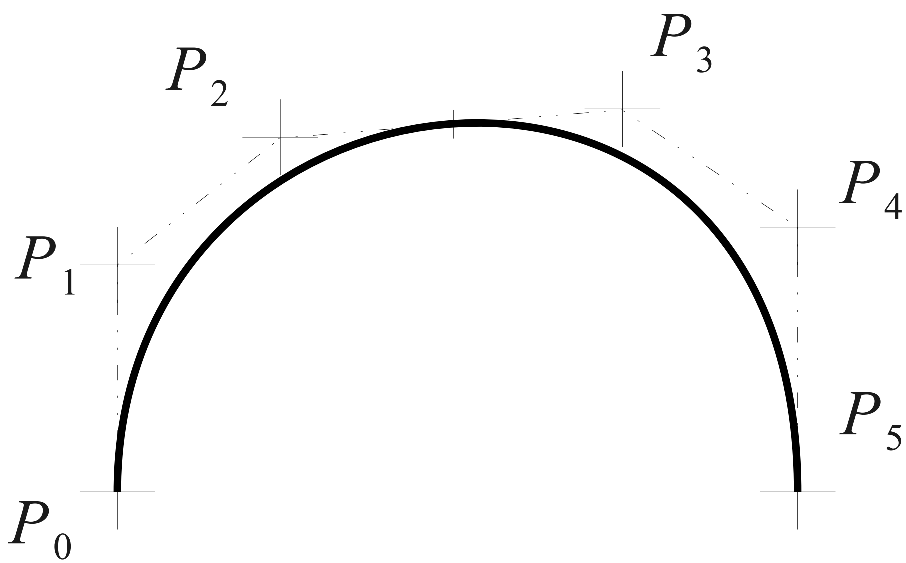
compute the distance between P and the control point P_i

otherwise

compute the distance between P^1_{i-1} and P^1_{j+1}

If the distance is lower than a tolerance T, replace the control points P_k by the new control points P_k^1 and update the nodal sequence as well.

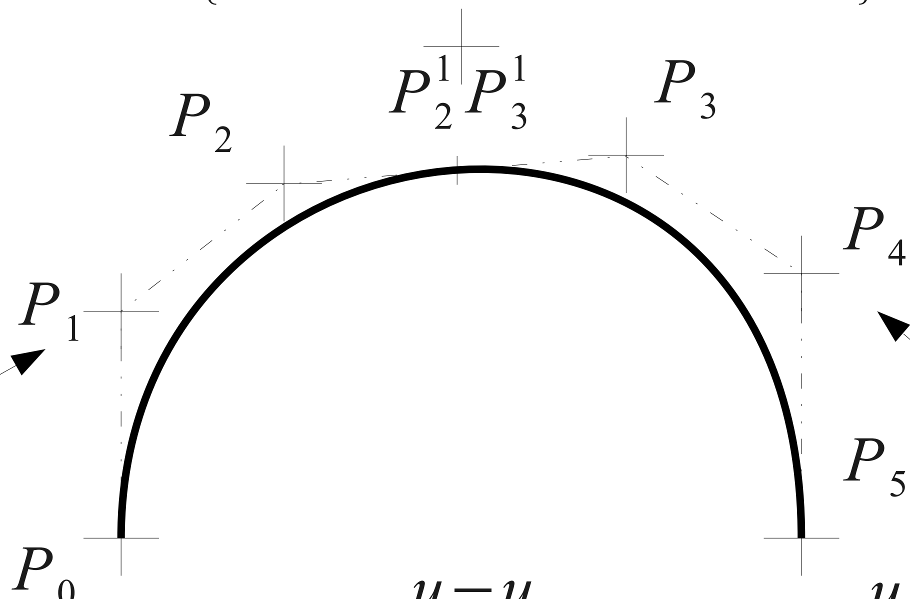
- Practical example: deletion of $u = u_5$



$$U = \{0, 0, 0, 0, 1, 1, 2, 2, 2, 2\}$$

- Deletion of $u = u_5 \quad (r=5\,,\, d=3\,,\, s=2)$

$$U = \{0\,,0\,,0\,,0\,,1\,,1\,,2\,,2\,,2\,,2\}$$

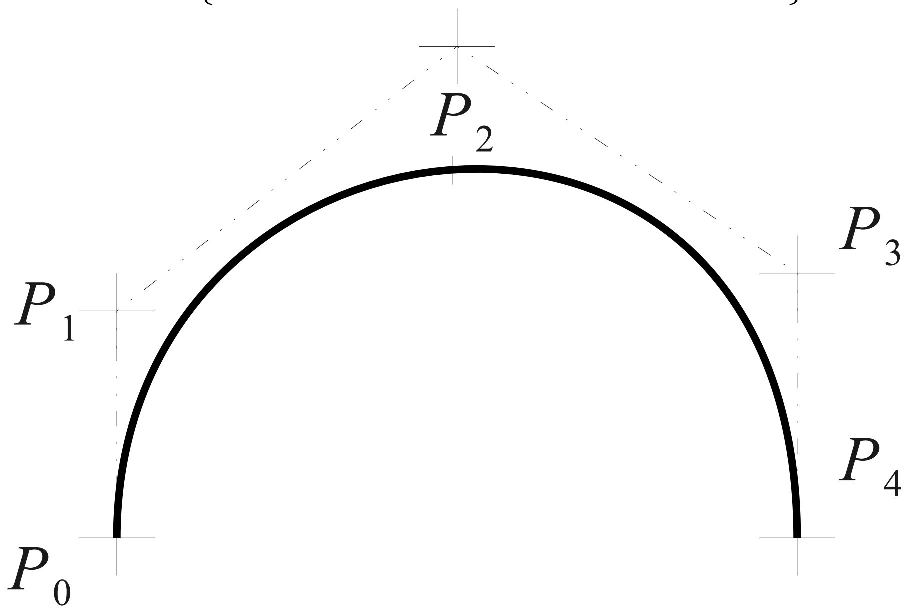


$$i = r\text{-}d = 2 \quad j = r\text{-}s = 3 \quad \alpha_2 = \frac{u - u_2}{u_6 - u_2} = 1/2 \quad \alpha_3 = \frac{u - u_3}{u_7 - u_3} = 1/2$$

$$P_2^1 = \frac{P_2^0 - (1-\alpha_2)\, P_1^1}{\alpha_2} = 2\, P_2^0 - P_1^1 \qquad P_3^1 = \frac{P_3^0 - \alpha_3\, P_4^1}{1-\alpha_3} = 2\, P_3^0 - P_4^1$$

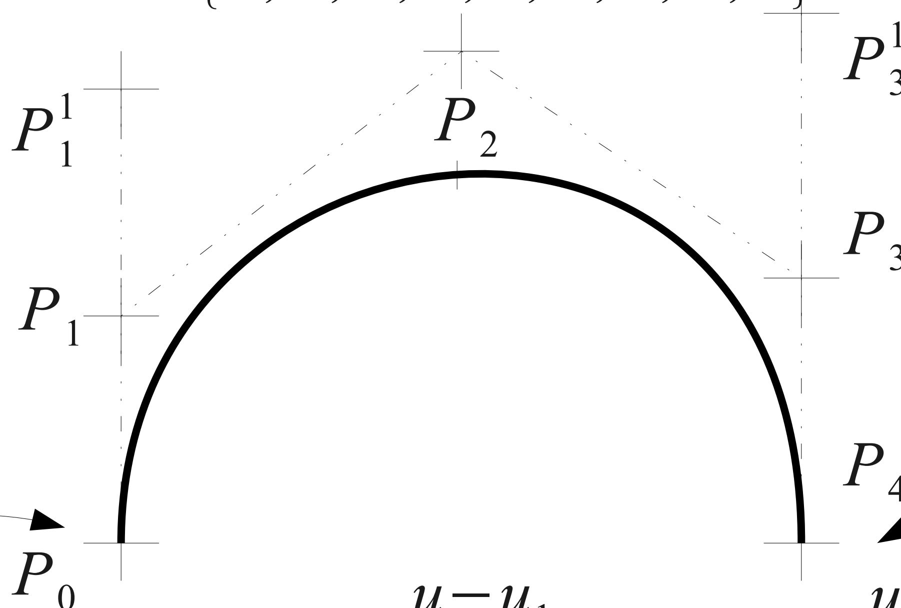
- New control polygon

$$U = \{0, 0, 0, 0, 1, 2, 2, 2, 2\}$$



- Now, deletion of $u=u_4$ $(r=4\ ,\ d=3\ ,\ s=1)$

$$U=\{0,0,0,0,1,2,2,2,2\}$$

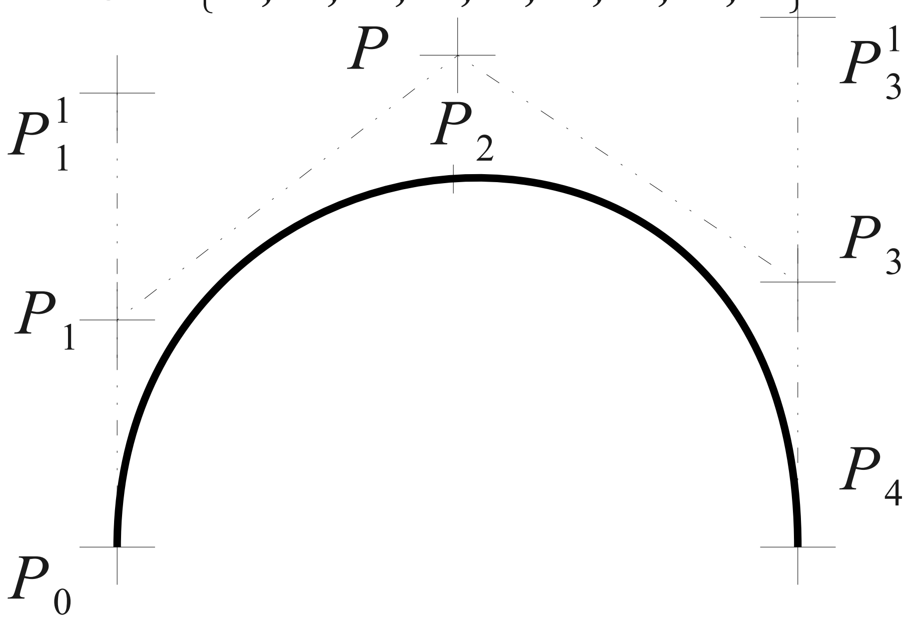


$$i=r\text{-}d=1 \quad j=r\text{-}s=3 \quad \alpha_1=\frac{u-u_1}{u_5-u_1}=1/2 \quad \alpha_3=\frac{u-u_3}{u_7-u_3}=1/2$$

$$P_1^1=\frac{P_1^0-(1-\alpha_1)P_0^1}{\alpha_1}=2P_1^0-P_0^1 \qquad P_3^1=\frac{P_3^0-\alpha_3 P_4^1}{1-\alpha_3}=2P_3^0-P_4^1$$

- Verification

$$U=\{0,0,0,0,1,2,2,2,2\}$$

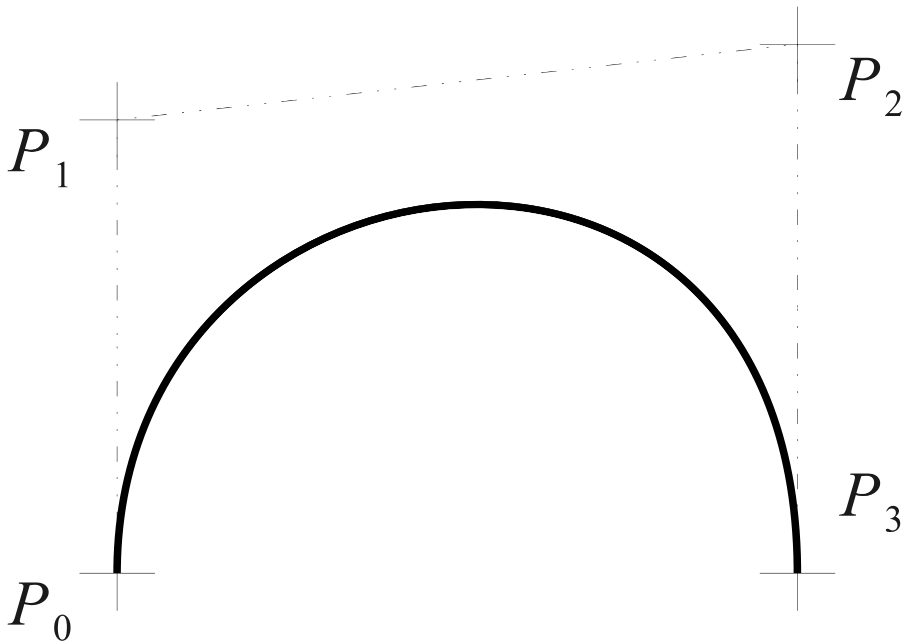


$i=r\text{-}d=1 \quad j=r\text{-}s=3$

$$\alpha_2=\frac{u-u_2}{u_6-u_2}=1/2 \quad P=\alpha_2 P_1^1+(1-\alpha_2)P_3^1=\frac{P_1^1+P_3^1}{2} \longrightarrow \text{OK !}$$

- Final control polygon

$$U = \{0, 0, 0, 0, 2, 2, 2, 2\}$$



- Restriction of the useful interval of a curve

Starting with a curve $P(u)=\sum_{i=0}^{n} N_i^d(u) P_i$

defined on the nodal sequence:

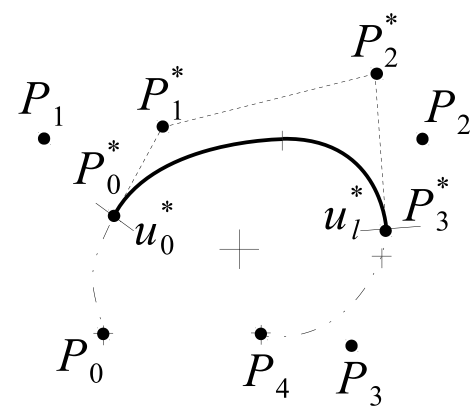
$$U=\{\underbrace{u_0,\cdots,u_d}_{d+1\ \text{times}},\cdots,\underbrace{u_{m-d},\cdots,u_m}_{d+1\ \text{times}}\}$$

We want to limit this curve at u_0^* and u_l^*

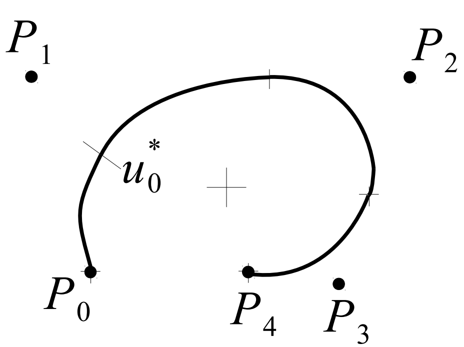
- Build the new control points P_i^*
- Determine the new nodal sequence U^*

$$P^*(u)=\sum_{i=0}^{k} N_i^d(u) P_i^*$$

$$U^*=\{\underbrace{u_0^*,\cdots,u_d^*}_{d+1\ \text{times}},\cdots,\underbrace{u_{l-d}^*,\cdots,u_l^*}_{d+1\ \text{times}}\}$$

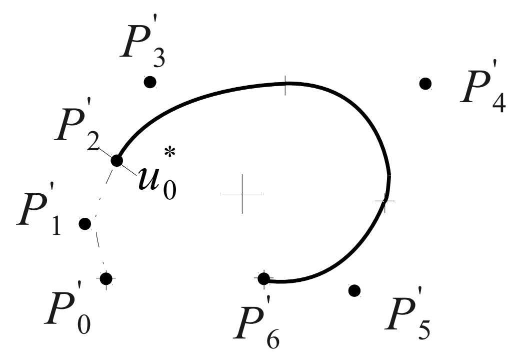


- Use of the Boehm's algorithm
  - Knot insertion $u = u_0^*$ up to a multiplicity equal to d.



$$U = \{0, 0, 0, 2, 4, 6, 6, 6\}$$

$$u_0^* = 1$$

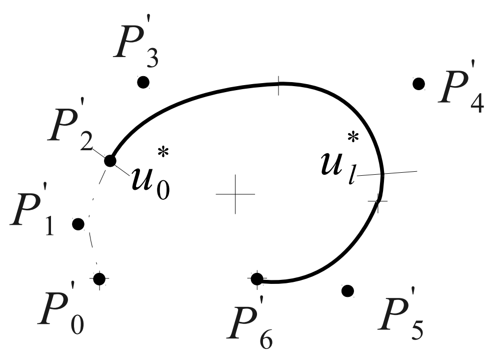


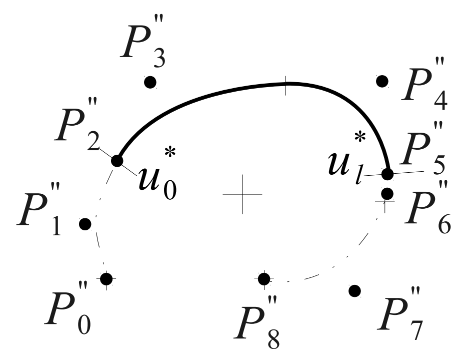
$$U' = \{0, 0, 0, 1, 1, 2, 4, 6, 6, 6\}$$

Computer Aided Design

B-Splines

- Use of the Boehm's algorithm
  - Knot insertion at $u = u_0^*$ until multplicity is equal to d.
  - Knot insertion at $u = u_l^*$ until multiplicity iq equal to d.
    - The sequence of these operations is not important.





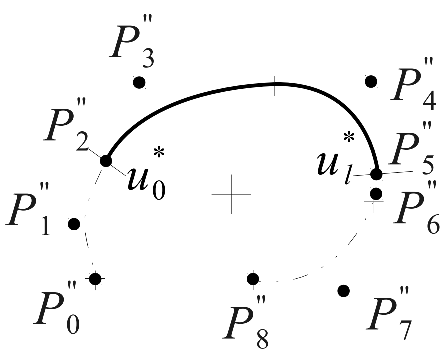
$$U' = \{0, 0, 0, 1, 1, 2, 4, 6, 6, 6\}$$

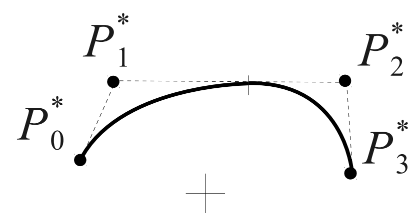
$$u_l^* = 3.5$$

$$U'' = \{0, 0, 0, 1, 1, 2, 3.5, 3.5, 4, 6, 6, 6\}$$

B-Splines

- Use of the Boehm's algorithm
  - Knot insertion in $u = u_0^*$ until multplicity is equal to d.
  - Knot insertion in $u = u_l^*$ until multiplicity is equal to d.
    - The order of these operations is not important.
  - Extraction of interesting part of the curve and addition of knots at the start and end of the nodal sequence so as to have a multiplicity equal to $d+1$ at both extremities

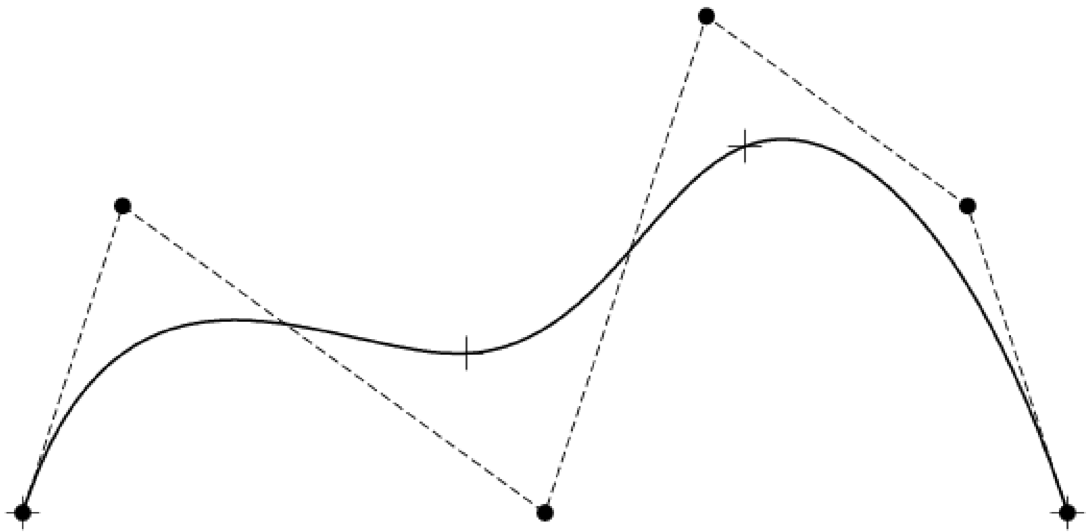




$$U^* = \{1, 1, 1, 2, 3.5, 3.5, 3.5\}$$

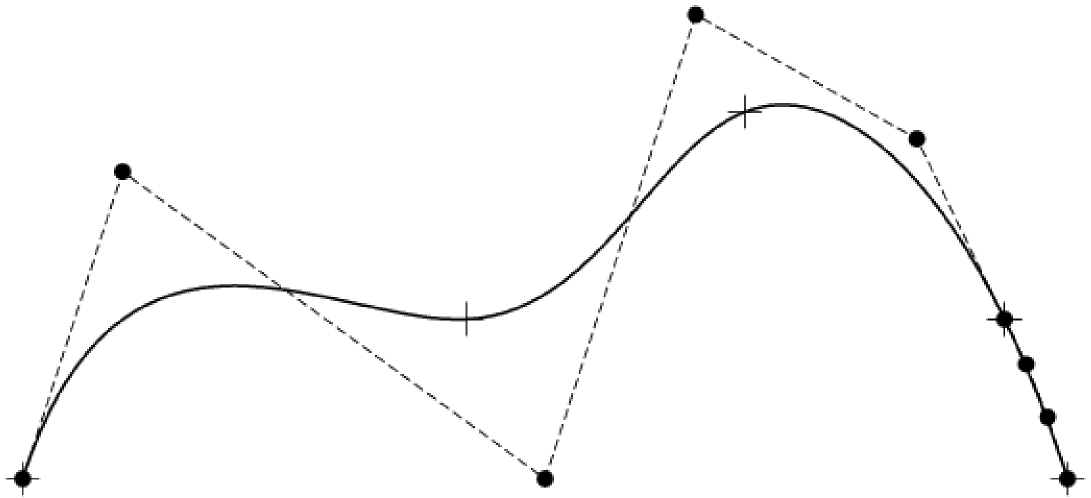
$$U'' = \{0, 0, 0, 1, 1, 2, 3.5, 3.5, 4, 6, 6, 6\}$$

B-Splines



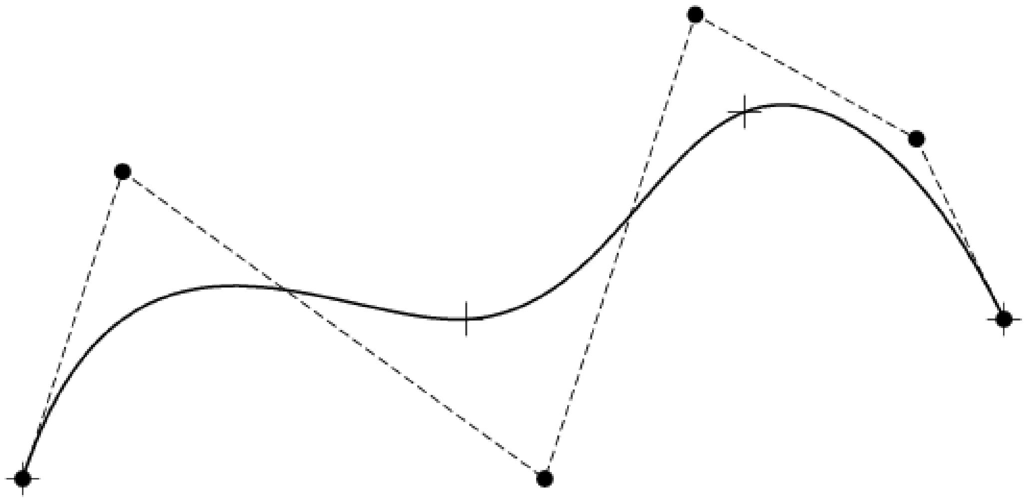
$$U=\{0,0,0,0,1,2,3,3,3,3\}$$

B-Splines



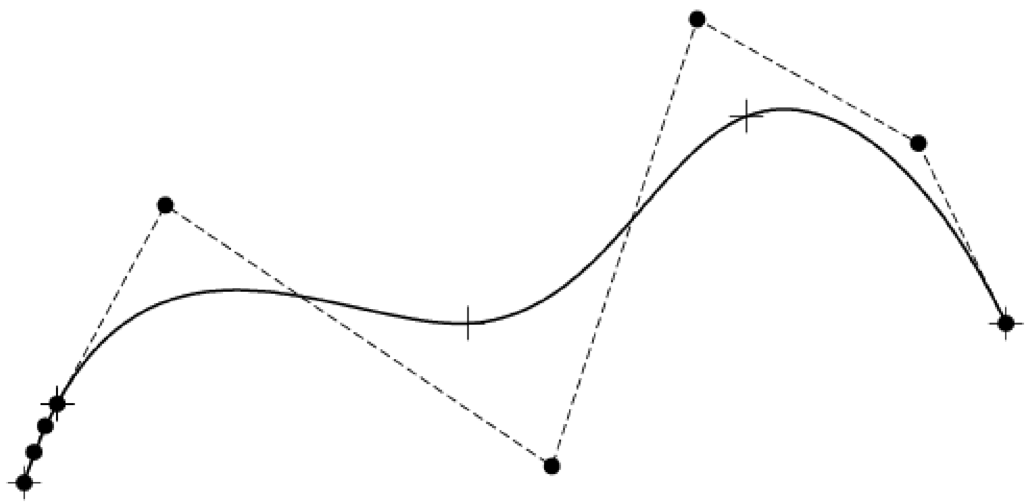
$$U = \{0, 0, 0, 0, 0.2, 0.2, 0.2, 1, 2, 3, 3, 3, 3\}$$

B-Splines



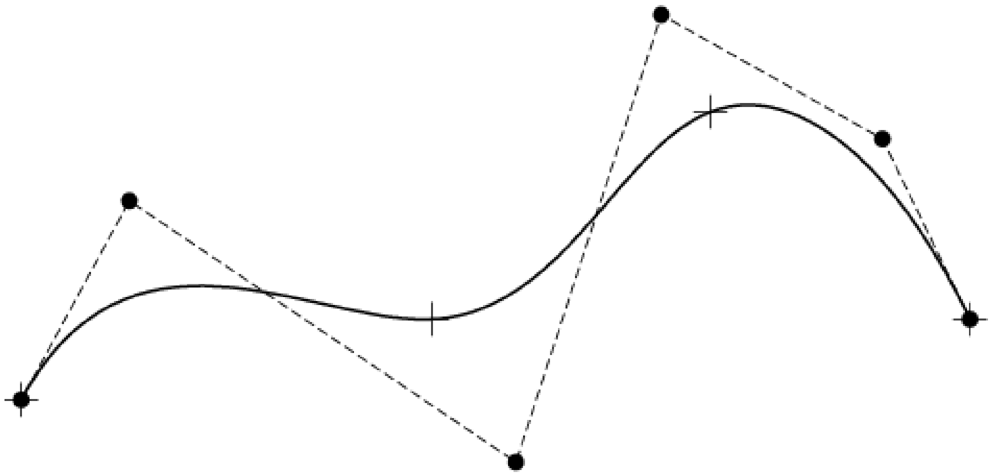
$$U = \{0.2\,,0.2\,,0.2\,,0.2\,,1\,,2\,,3\,,3\,,3\,,3\}$$

B-Splines



$$U = \{0.2\,,0.2\,,0.2\,,0.2\,,1\,,2\,,2.9\,,2.9\,,2.9\,,3\,,3\,,3\,,3\}$$

B-Splines

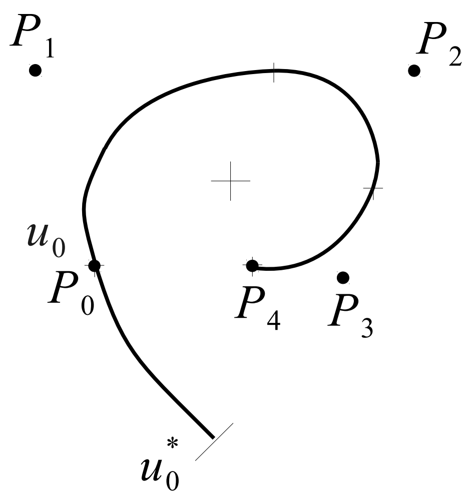


$$U = \{0.2\,, 0.2\,, 0.2\,, 0.2\,, 1\,, 2\,, 2.9\,, 2.9\,, 2.9\,, 2.9\}$$

- Increase of a curve's useful interval

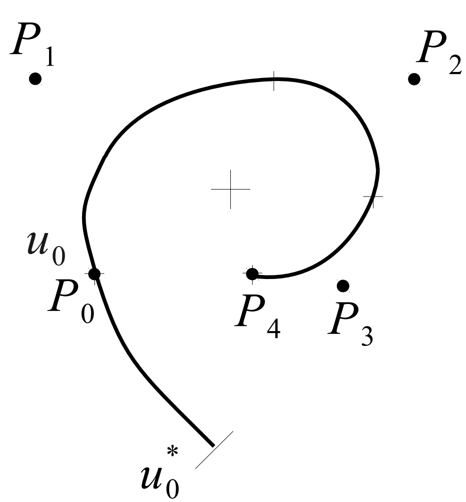
$$P(u)=\sum_{i=0}^{n} N_i^d(u)\,P_i \qquad U=\{\underbrace{u_0,\cdots,u_d}_{d+1\ \text{times}},\cdots,\underbrace{u_{m-d},\cdots,u_m}_{d+1\ \text{times}}\}$$

  - We want to extend the curve to $u=u_0^*<u_0$
    or to $u=u_m^*>u_m$



B-Splines

- We cannot use the node insertion algorithm directly
- Instead, use of Cox-de Boor's algorithm « backwards » to determine the new position of the control points P_i^*
  - We move the node u_0 to u_0^*. The new nodal sequence is :



$$U^* = \{\underbrace{u_0^*, \cdots, u_d^*}_{d+1 \text{ times}}, \underbrace{u_{d+1}, \cdots, \underbrace{u_{m-d}, \cdots, u_m}_{d+1 \text{ times}}}_{\text{unchanged}}\}$$

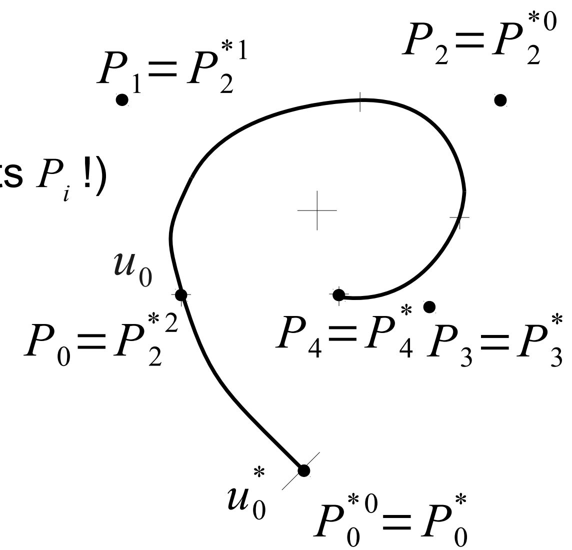
- We calculate the point corresponding to u_0 from unknown positions of control points P_i^*
  - Sequence of points obtained with the Cox-de Boor algorithm :

For k from 1 to d
  For j from i to i-d+k

$$P_j^{*k} = \left(\frac{u_0 - u_j^*}{u_{j+d+1-k}^* - u_j^*} \right) P_j^{*k-1} + \left(\frac{u_{j+d+1-k}^* - u_0}{u_{j+d+1-k}^* - u_j^*} \right) P_{j-1}^{*k-1}$$

  - Among these points :
    - The points P_d^{*d-i} are known (they are the points P_i !)
    - Only the $d+1$ first control points are affected (the interval concerned by Cox-de Boor is $[u_i, u_{i+1}[$, $i=d$).

for k from 1 to d
  for j from d to k



B-Splines

known

For k from 1 to d
For j from d to k

$$\boxed{P_j^{*k}} = \left(\frac{u_0 - u_j^*}{u_{j+d+1-k}^* - u_j^*}\right)\boxed{P_j^{*k-1}} + \left(\frac{u_{j+d+1-k}^* - u_0}{u_{j+d+1-k}^* - u_j^*}\right)P_{j-1}^{*k-1}$$

Is transformed in :

$$P_{j-1}^{*k-1} = \left(\frac{u_{j+d+1-k}^* - u_j^*}{u_{j+d+1-k}^* - u_0}\right)P_j^{*k} - \left(\frac{u_0 - u_j^*}{u_{j+d+1-k}^* - u_j^*}\right)\left(\frac{u_{j+d+1-k}^* - u_j^*}{u_{j+d+1-k}^* - u_0}\right)P_j^{*k-1}$$

Initialise $P_d^{*d-i} = P_i$
For j from d to 1
For k from j to 1

$$P_{j-1}^{*k-1} = \left(\frac{u_{j+d+1-k}^* - u_j^*}{u_{j+d+1-k}^* - u_0}\right)P_j^{*k} + \left(\frac{u_j^* - u_0}{u_{j+d+1-k}^* - u_0}\right)P_j^{*k-1}$$

B-Splines

$d=2 \quad U=\{0,0,0,1,2,3,3,3\} \quad u_0=0$

$U^*=\{-0.1,-0.1,-0.1,1,2,3,3,3\} \quad u_0^*=-0.1$

For j from 2 to 1
For k from j to 1

$$P_{j-1}^{*k-1}=\left(\frac{u_{j+3-k}^*-u_j^*}{u_{j+3-k}^*}\right)P_j^{*k}+\left(\frac{u_j^*}{u_{j+3-k}^*}\right)P_j^{*k-1}$$

j=2 k=2

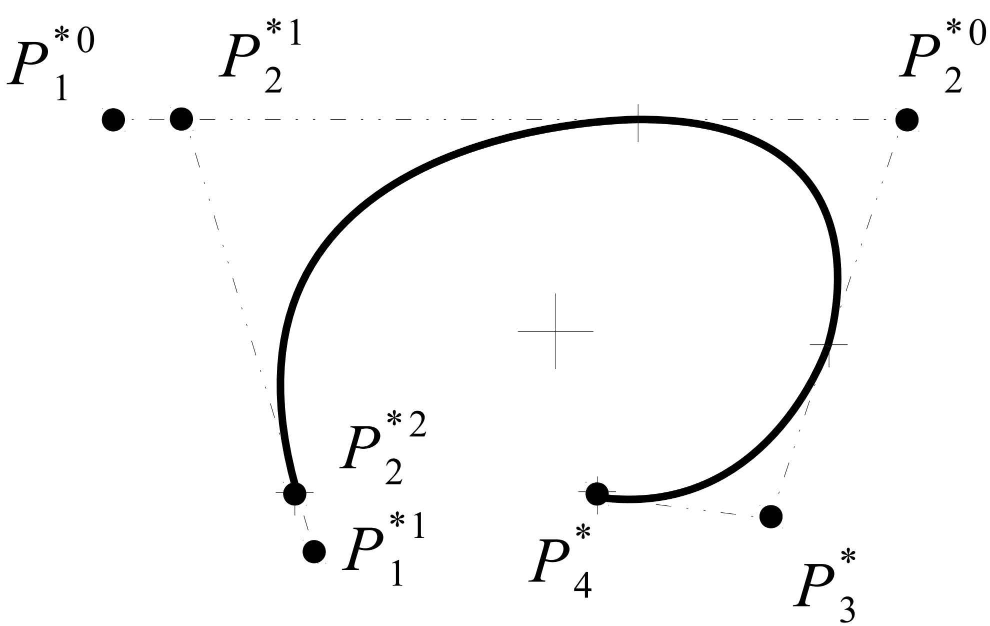
$$P_1^{*1}=\left(\frac{u_3^*-u_2^*}{u_3^*}\right)P_2^{*2}+\left(\frac{u_2^*}{u_3^*}\right)P_2^{*1}$$

$$P_1^{*1}=1.1\,P_2^{*2}-0.1\,P_2^{*1}$$

j=2 k=1

$$P_1^{*0}=\left(\frac{u_4^*-u_2^*}{u_4^*}\right)P_2^{*1}+\left(\frac{u_2^*}{u_4^*}\right)P_2^{*0}$$

$$P_1^{*0}=1.05\,P_2^{*1}-0.05\,P_2^{*0}$$



B-Splines

$$d=2 \quad U=\{0,0,0,1,2,3,3,3\} \quad u_0=0$$

$$U^*=\{-0.1,-0.1,-0.1,1,2,3,3,3\} \quad u_0^*=-0.1$$

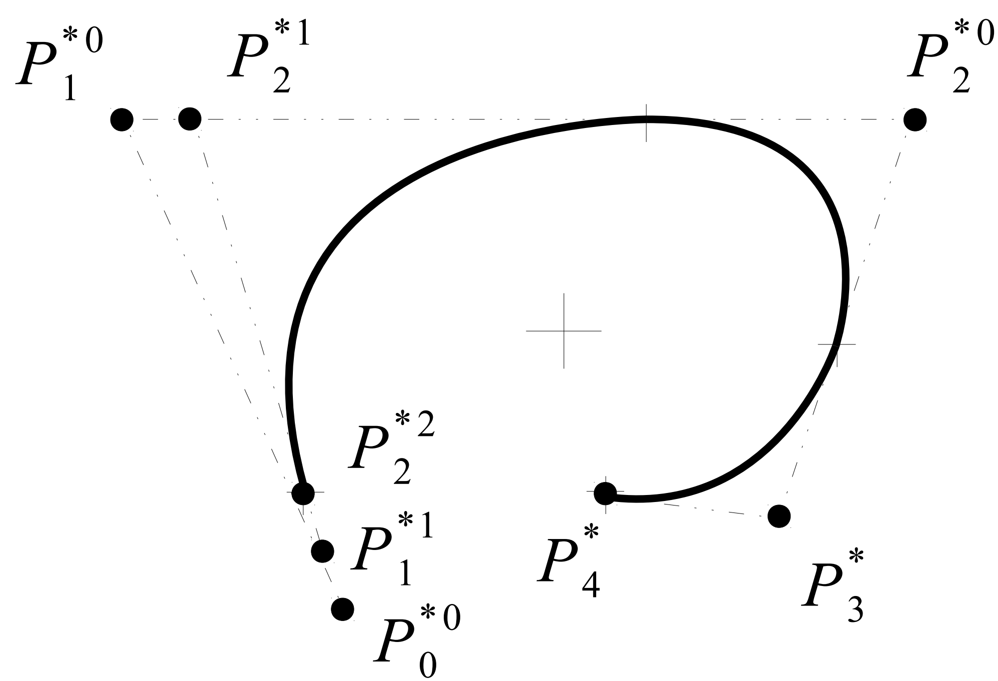
For j from 2 to 1
  For k from j to 1

$$P_{j-1}^{*k-1}=\left(\frac{u_{j+3-k}^*-u_j^*}{u_{j+3-k}^*}\right)P_j^{*k}+\left(\frac{u_j^*}{u_{j+3-k}^*}\right)P_j^{*k-1}$$

j=1 k=1

$$P_0^{*0}=\left(\frac{u_3^*-u_1^*}{u_3^*}\right)P_1^{*1}+\left(\frac{u_1^*}{u_3^*}\right)P_1^{*0}$$

$$P_0^{*0}=1.1\,P_1^{*1}-0.1\,P_1^{*0}$$

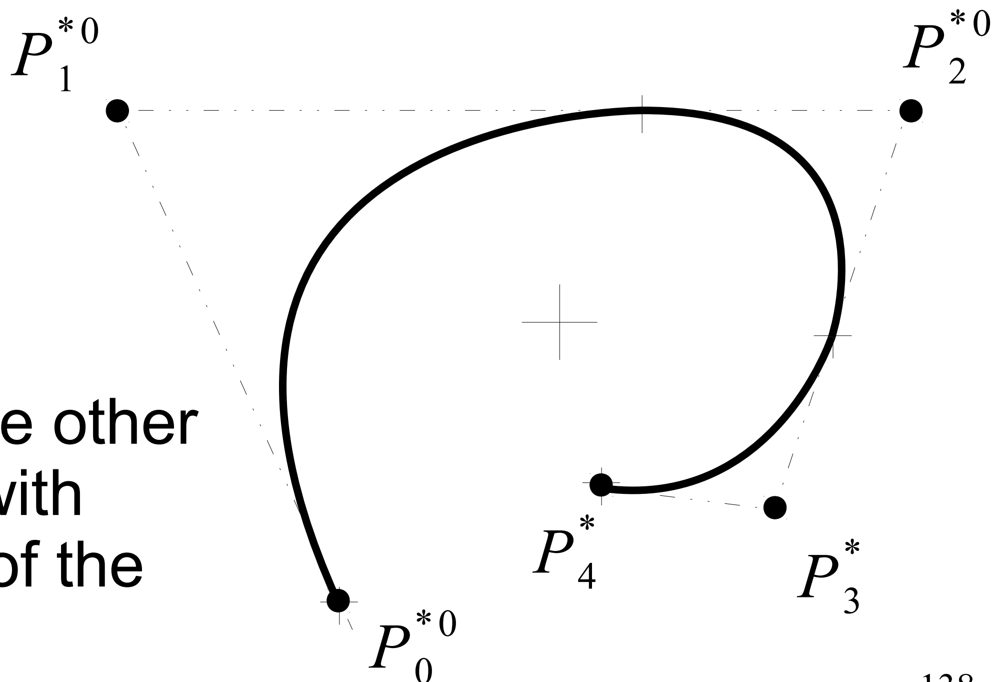


- Control polygon and modified nodal sequence

$$U^* = \{-0.1, -0.1, -0.1, 1, 2, 3, 3, 3\}$$

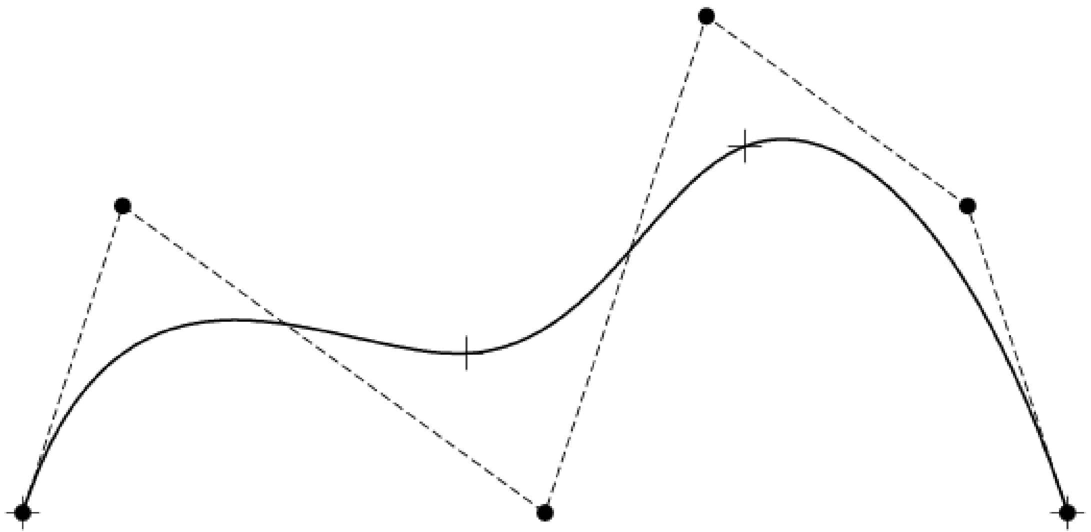
$$d = 2$$

$$\begin{cases} P_i^* = P_i^{*0} & , i < d \\ P_i^* = P_i & , i \geq d \end{cases}$$



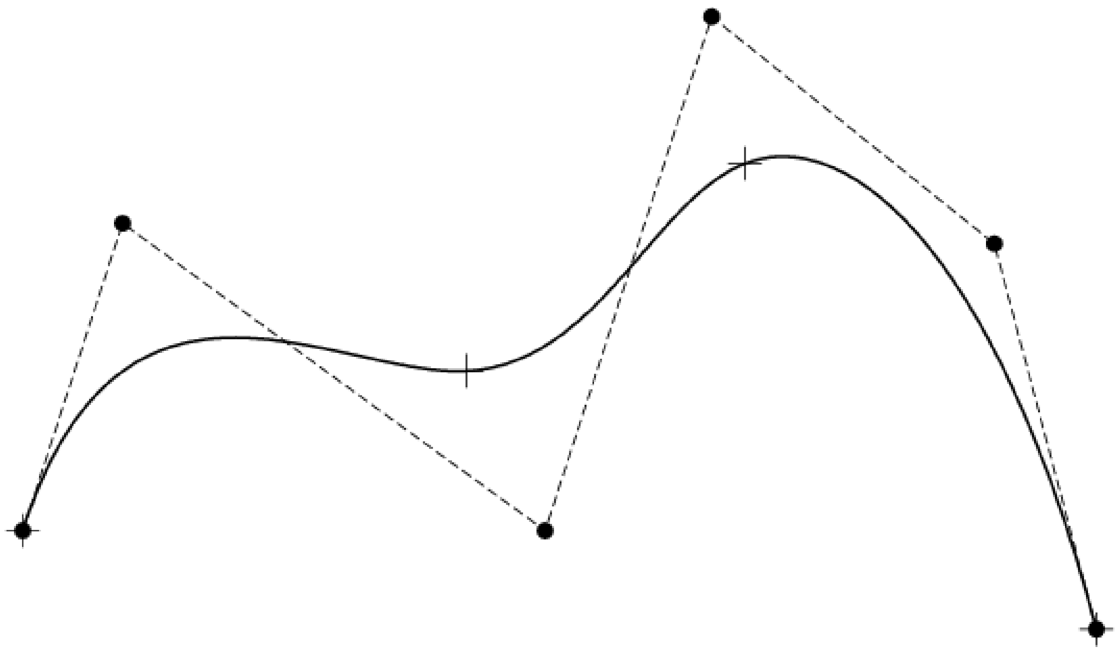
- For an extension on the other side; same algorithm with change of the indices of the affected points.

B-Splines



$$U = \{0, 0, 0, 0, 1, 2, 3, 3, 3, 3\}$$

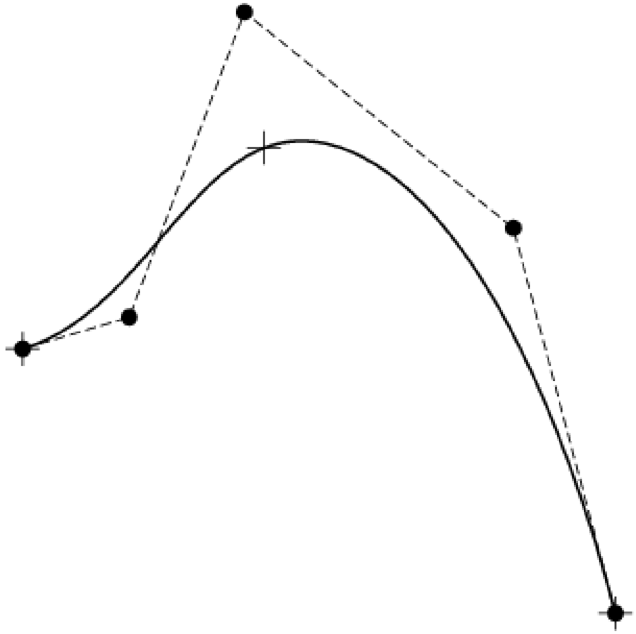
B-Splines



Increase for u=-0.1

$$U=\{-0.1\,,-0.1\,,-0.1\,,-0.1\,,1\,,2\,,3\,,3\,,3\,,3\}$$

B-Splines



$$U=\{-0.1,-0.1,-0.1,-0.1,1,1.9,1.9,1.9,1.9\}$$

- Degree elevation
  - A B-spline curve of degree d can be converted without loss into a curve of degree $d+1$.
    - Proof : shape functions of degree $d+1$ "contain" those of degree d.
  - The continuity of the curve of degree d at the nodes is given by the nodal sequence.
  - For a same multiplicity of a node, the curve of degree $d+1$ offers a higher continuity.
    - As a consequence, in the nodal sequence of the new curve, one must increment the multiplicity of each node.

- Degree elevation algorithm
  - Several algorithms exist :

Tiller et al (1983)

- Direct resolution of a linear system

Prautzch et al. (1991)

- Cf JC Leon, « Modélisation de curves et surfaces pour la CFAO »

Piegl et al. (1994)

- Competitive with Prautzch (specially for multiple elevations)
- Transform the curve in succession of Bézier curves by insertion of nodes (multiplicity = d)
- Increase the degree of Bézier curves (ref. *ad-hoc* algorithm as seen for Bézier curves)
- Removal of most of nodes that were introduced

- Recursive subdivision
  - As in the case of Bézier curves, one can increase the number of CPs , this time using Boehm's knot insertion algorithm instead of De Casteljau's.
    - The control polygon converges to the curve as one inserts new knots (each time in the middle of the widest knot interval)
  - The usual order of B-Spline curves is small (3 or 4, maybe 5), so savings in computational complexity with respect to a systematic evaluation with Cox-de Boor's algorithm are not so dramatic.

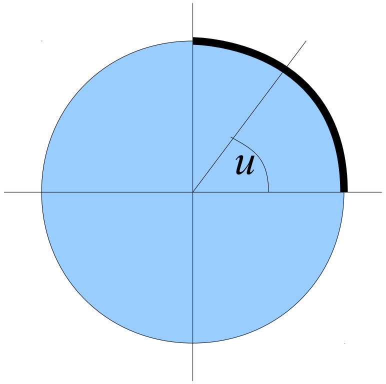
- Some conclusions
  - Flexibility
  - Low order
  - Continuity
  - Periodic curves
  - Conics ?

Rational curves
« Non Uniform Rational B-Splines »

NURBS

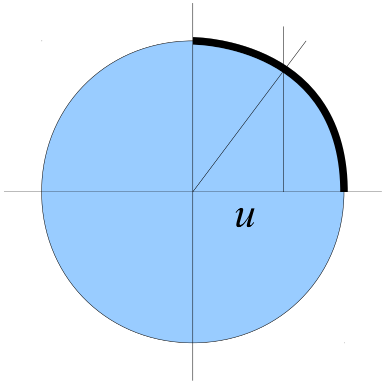
- Representation of conic sections
  - Circles , ellipses, hyperbolas...
  - A fair number of « industrial » geometries do have conic sections in their definition
  - Bézier curves, B-Splines and other polynomial representations cannot exactly represent all conic sections

- Example : the circular arc
  - Parametric form A
    u is the angle



$$\begin{cases} x(u)=\cos u \\ y(u)=\sin u \end{cases}, \quad u\in[0,\frac{\pi}{2}]$$

  - No polynomial representation (or as a series)
  - Parametric form B



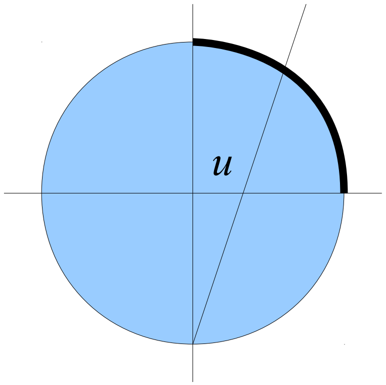
$$\begin{cases} x(u)=1-u \\ y(u)=\sqrt{u(1-u)} \end{cases}, \quad u\in[0,1]$$

  - Problem : square root

NURBS

- Parametric form C

$$\begin{cases} x(u) = \dfrac{2u}{1+u^2} \\ y(u) = \dfrac{1-u^2}{1+u^2} \end{cases}, \quad u \in [0,1]$$



- Ratio of two polynomials of degree 2
- $x(u)$ and $y(u)$ have a common denominator

- Thus, one can set the following :

$$\begin{cases} x(u)\cdot w(u)=2u \\ y(u)\cdot w(u)=1-u^2 \quad , \; u\in[0,1] \\ w(u)=1+u^2 \end{cases}$$

$$P(u)=\frac{P^w(u)}{w(u)}$$

Vectorial term

Weight function (scalar)

NURBS

- Non rational curves

$$P(u) = \sum_i N_i^d(u) P_i$$

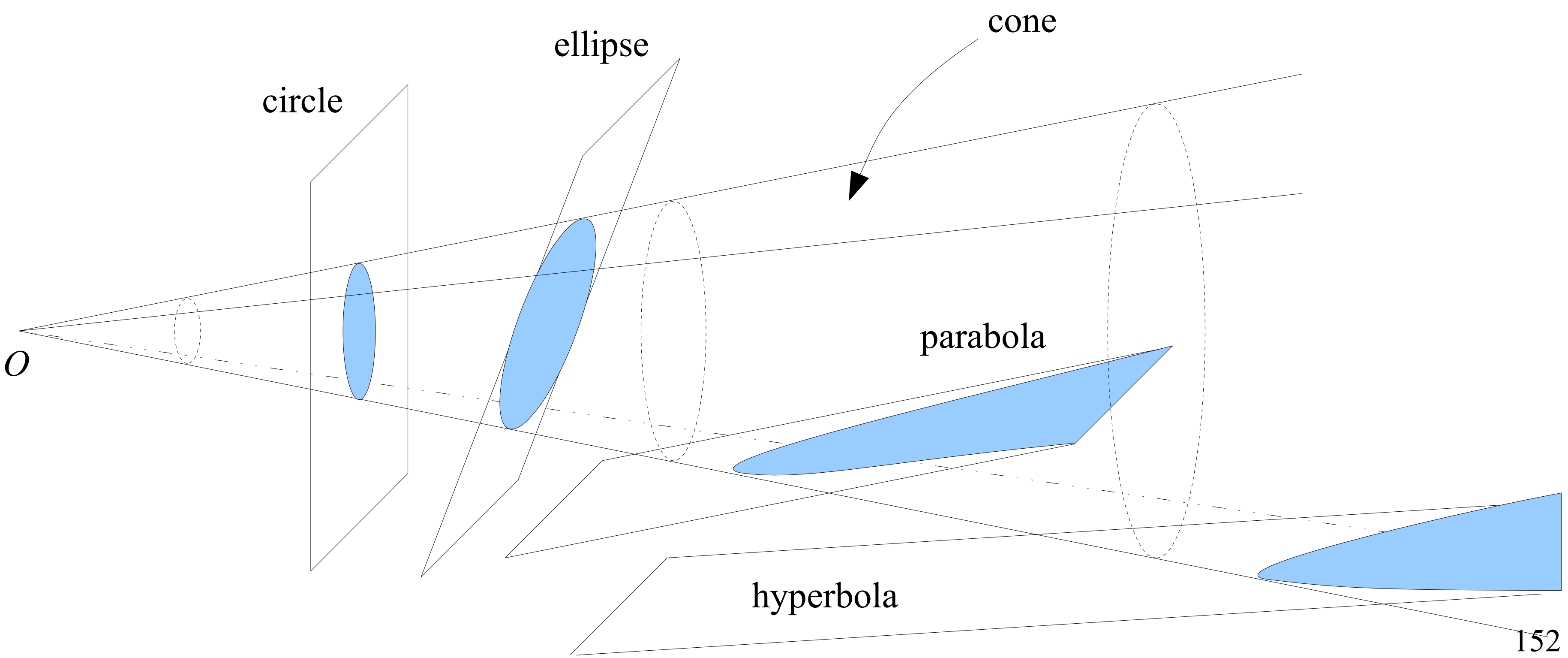
- Rational curves

$$P(u) = \frac{\sum_i N_i^d(u) P_i w_i}{\sum_j N_j^d(u) w_j} = \frac{P^w(u)}{w(u)} = \sum_i R_i^d(u) P_i$$

$$\text{with} \quad R_i^d(u) = \frac{N_i^d(u) w_i}{\sum_j N_j^d(u) w_j}$$

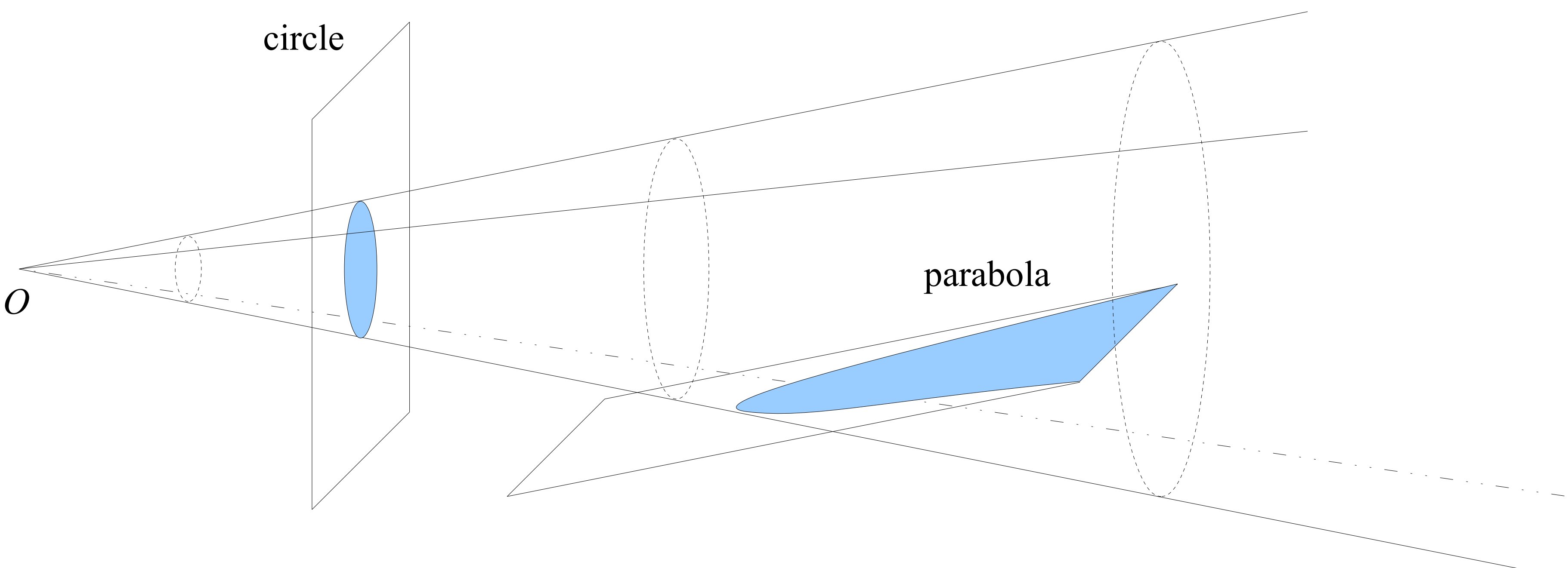
  - A weight w_i is associated to every control point P_i ,
  - A scalar approximation $w(u)$ is built on the same model as for regular B-Splines

- Relation between conics and the central projection
  - With respect to central projection; all conics are equivalent.



NURBS

- In particular, parabolas may be modelled with polynomials.
- Any other conic section can be obtained by projecting a parabola onto an appropriately oriented plane.



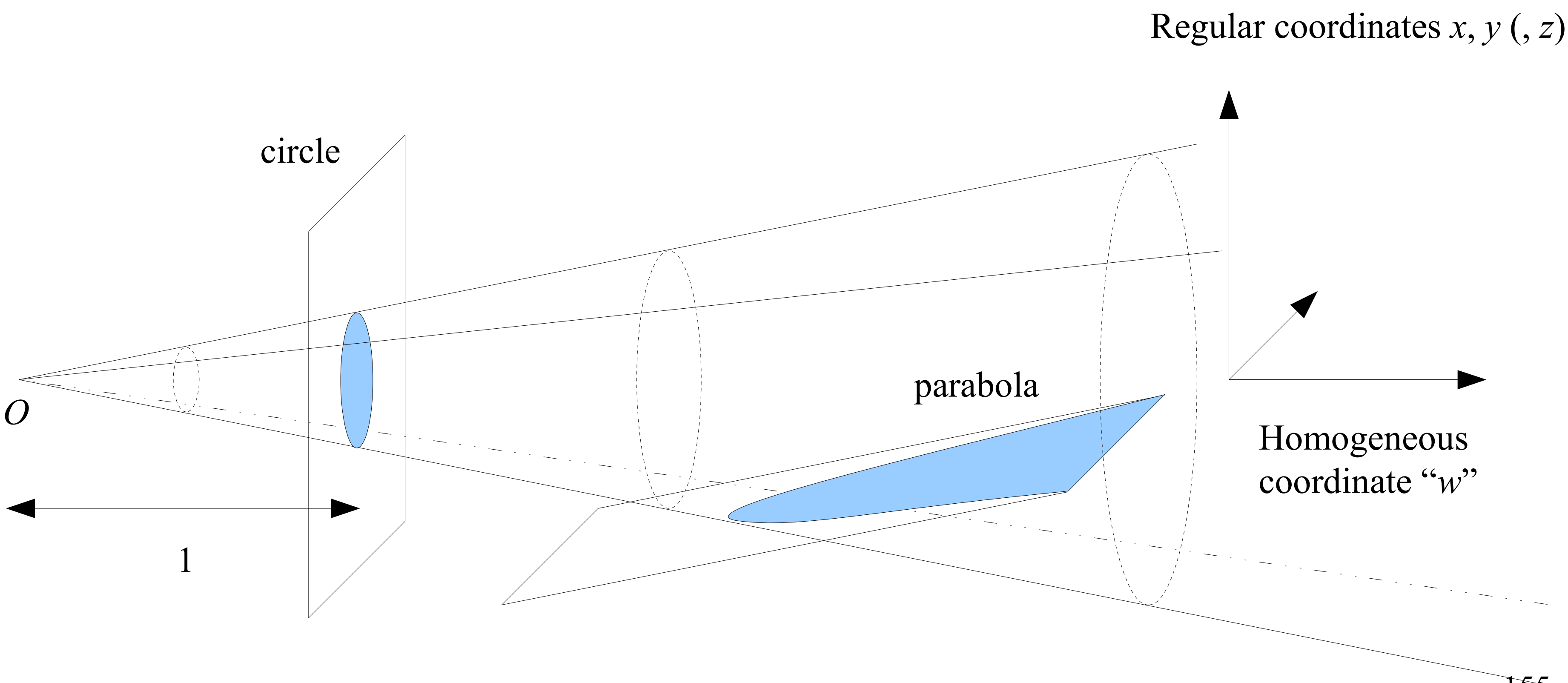
NURBS

- Central projection and homogeneous coordinates
  - Everything takes place in a 4-dimensional space
  - We have a class of equivalence :

$$[wx, wy, wz, w] \equiv [x, y, z, 1] \qquad \forall w > 0$$

    - This equivalence class is a central projection of all the points located on a straight line passing by the origin onto the plane of equation $w=1$
  - How to get cartesian 3D coordinates from the homogeneous 4D coordinates :
    - Just divide all the components by the 4th component
    - Discard the 4th component (it is equal to one)

- Projection of a parabola to a circle



NURBS

- In practice .. one works on the 4 homogeneous coordinates

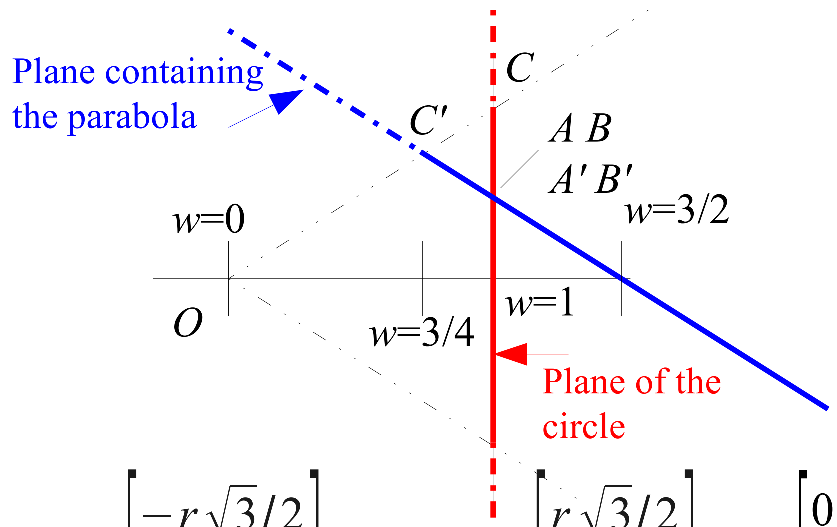
$$P^w(u) = \sum_i N_i^d(u)\, P_i^w \quad \text{with} \quad P_i^w = \begin{pmatrix} x_i \cdot w_i \\ y_i \cdot w_i \\ z_i \cdot w_i \\ w_i \end{pmatrix}$$

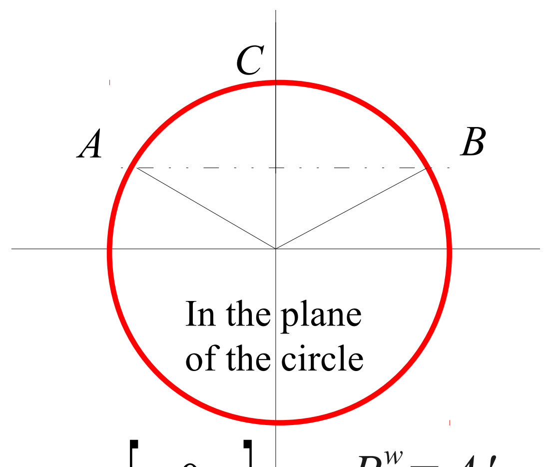
- In this space, we just know how to represent parabolas.
- The equivalence class allows us to divide by the last component and bring us back to euclidean 3D space
- We can therefore represent any conic section.
- How to determine the right control points P_i^w ?

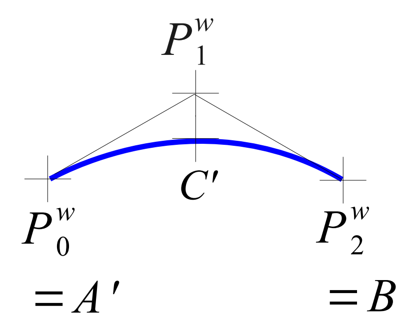
Computer Aided Design

NURBS

- Representation of 1/3rd of a circle with the help of B-splines (or Bézier) curves...
  - 3 control points, nodal sequence [0,0,0,1,1,1] (a parabola)
  - We must determine homogeneous coordinates of each control point so that the central projection yields points exactly on the arc.







$$A = A' = \begin{bmatrix} -r\sqrt{3}/2 \\ r/2 \\ 0 \\ 1 \end{bmatrix} \quad B = B' = \begin{bmatrix} r\sqrt{3}/2 \\ r/2 \\ 0 \\ 1 \end{bmatrix} \quad C = \begin{bmatrix} 0 \\ r \\ 0 \\ 1 \end{bmatrix} \quad C' = \begin{bmatrix} 0 \\ 3r/4 \\ 0 \\ 3/4 \end{bmatrix}$$

$$P_0^w = A'$$

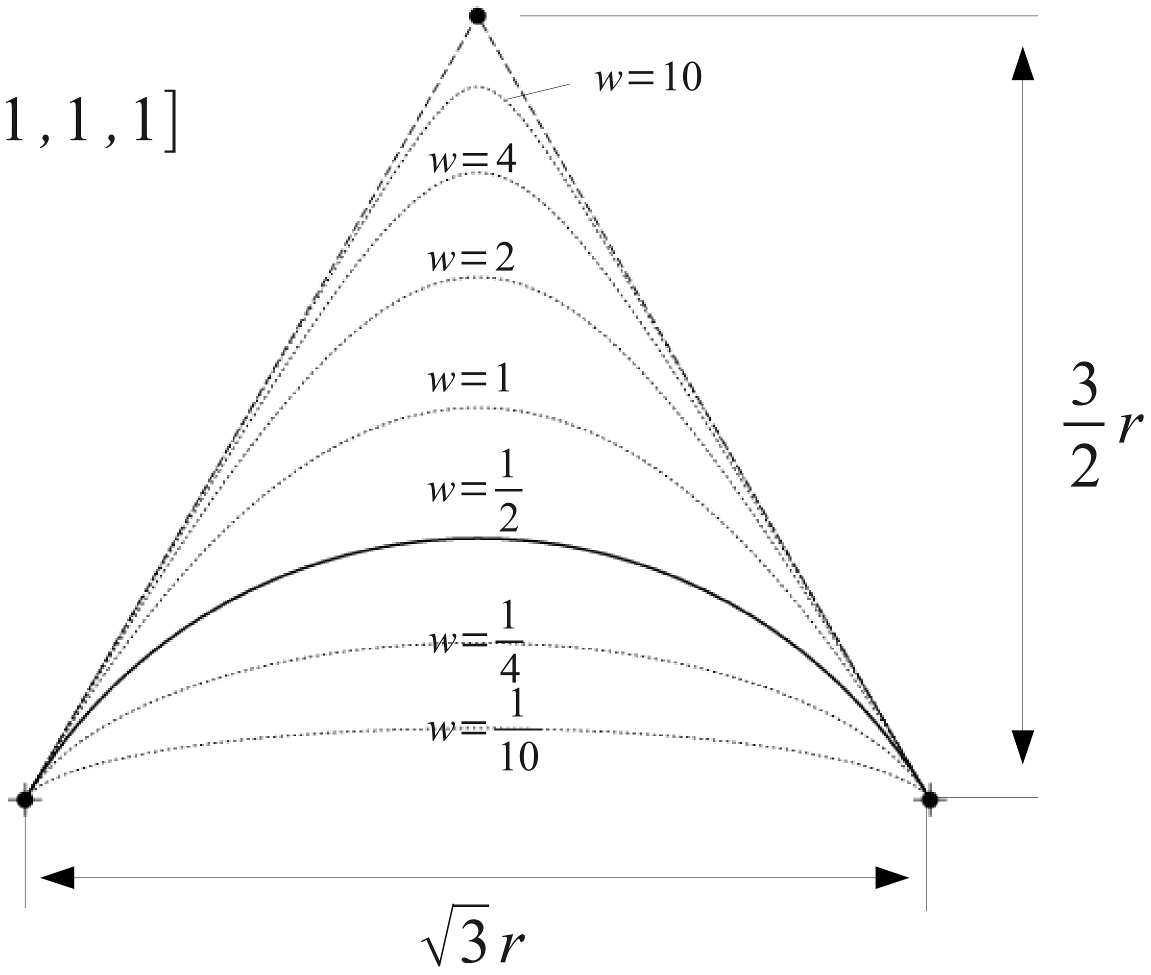
$$P_1^w = 2C' - \frac{(A'+B')}{2} = \begin{bmatrix} 0 \\ r \\ 0 \\ 1/2 \end{bmatrix}$$

$$P_2^w = B'$$

NURBS

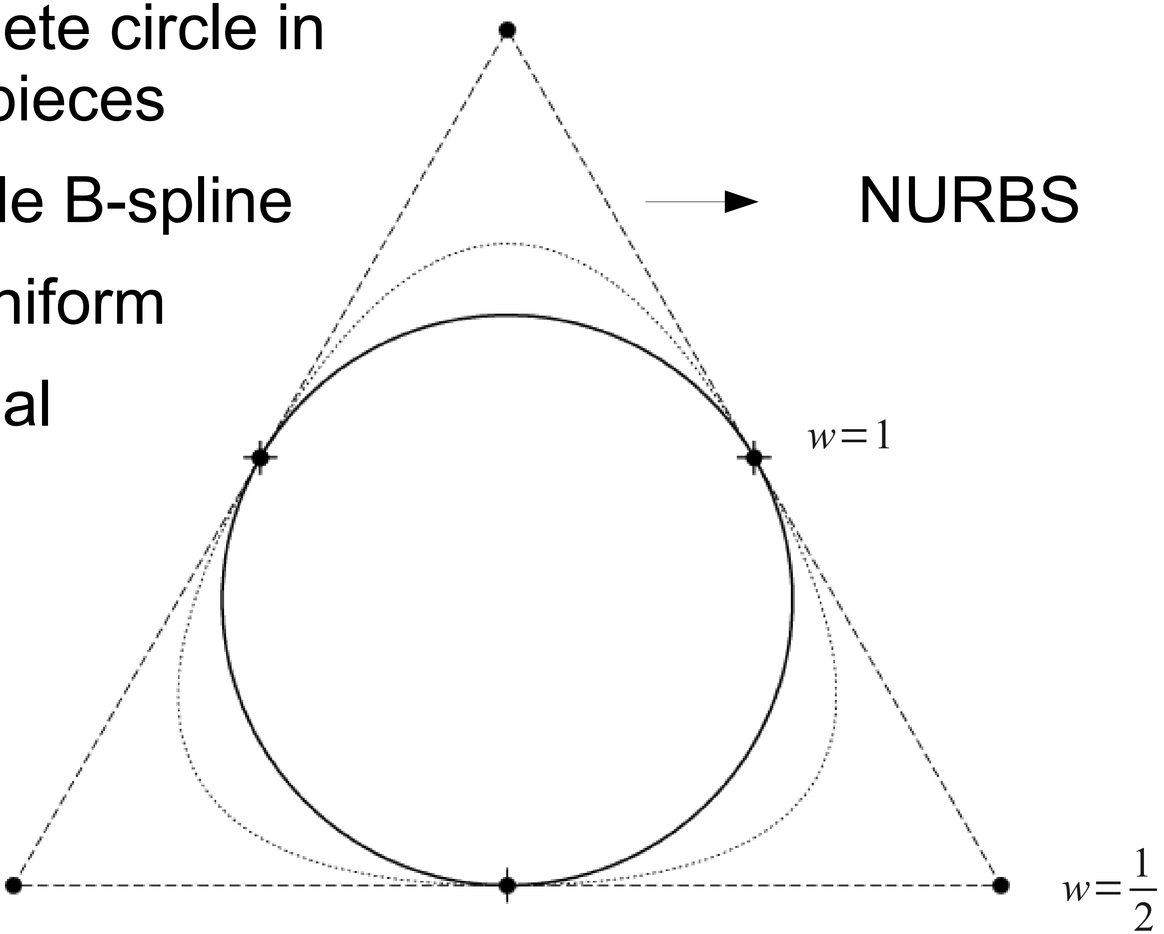
- 1/3 of a circle

$$U = [0, 0, 0, 1, 1, 1]$$



NURBS

- Complete circle in three pieces
- A single B-spline
- Non uniform
- Rational

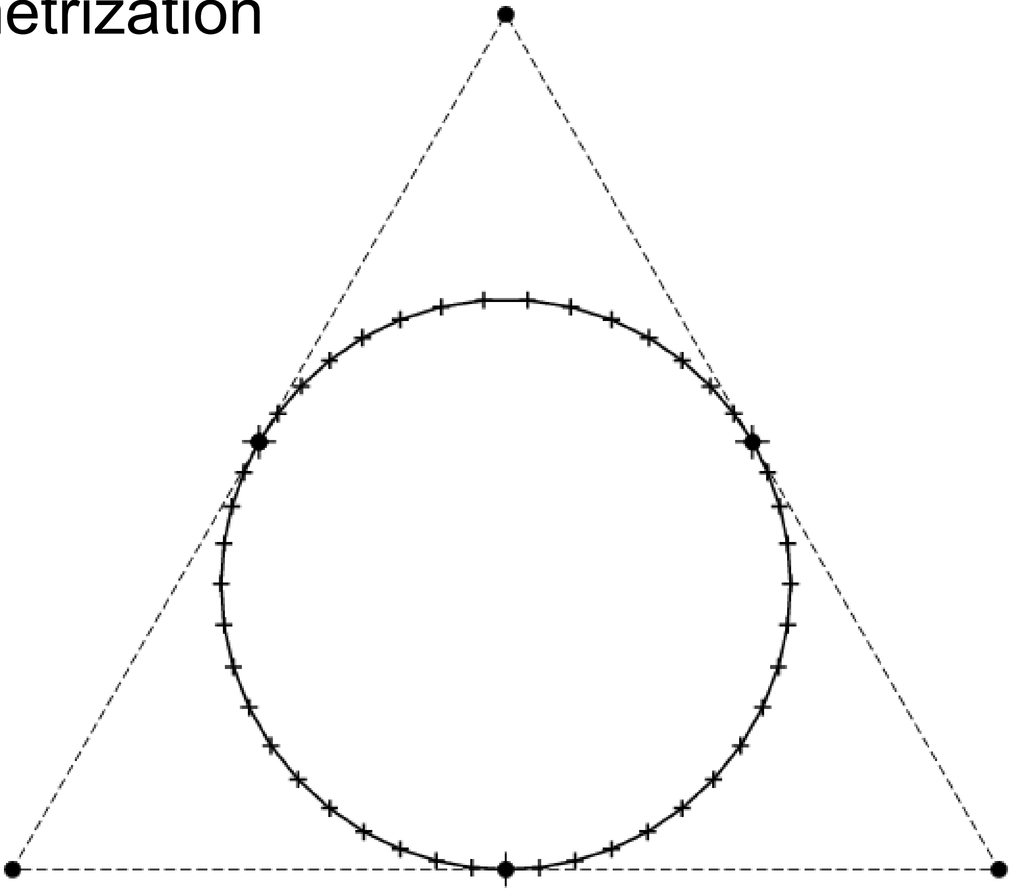


$$U=[0,0,0,1,1,2,2,3,3,3]$$

Computer Aided Design

NURBS

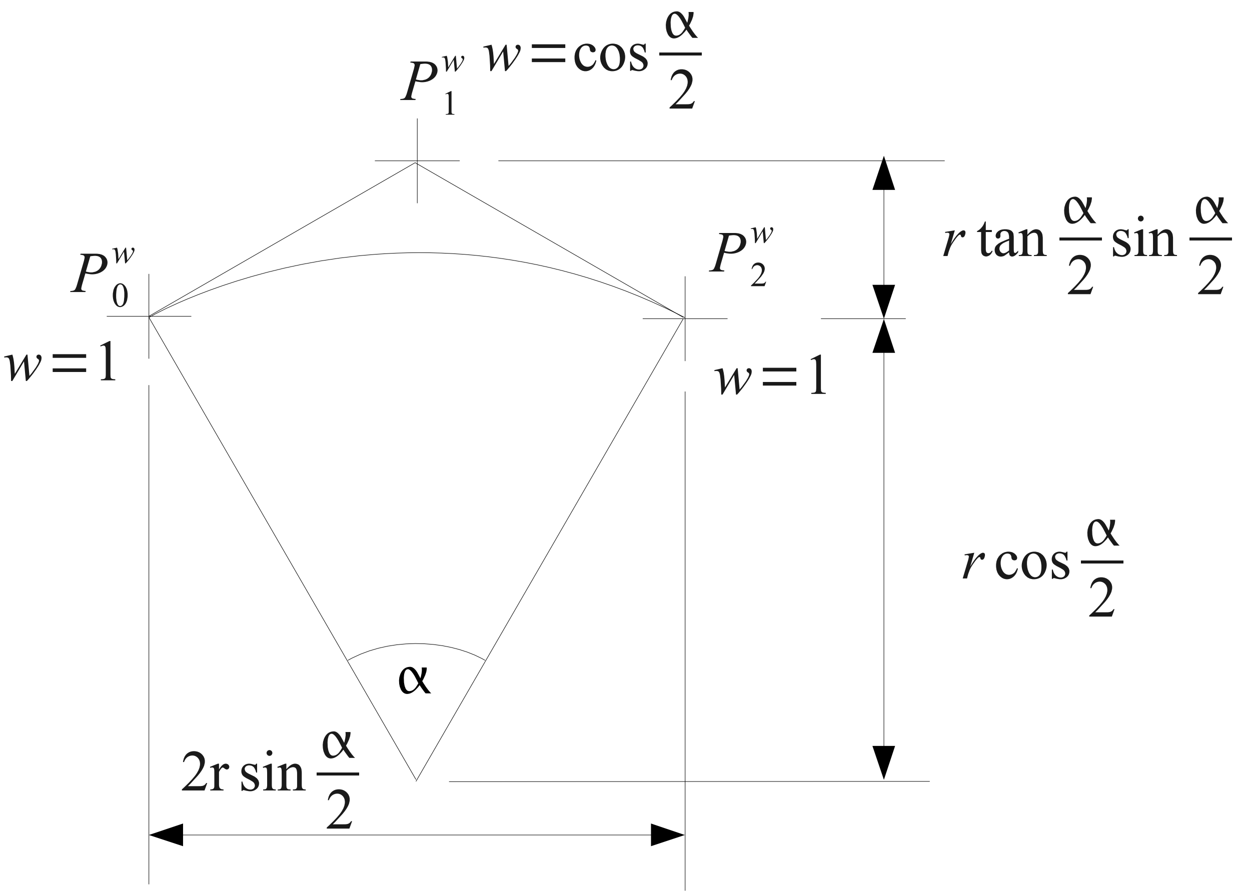
- Parametrization



$$U = [0, 0, 0, 1, 1, 2, 2, 3, 3, 3]$$

NURBS

- Generalization : Circular arc of angle $\alpha <= 2\pi/3$ (actually, $<\pi$)



$$P_0^w = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{pmatrix}$$

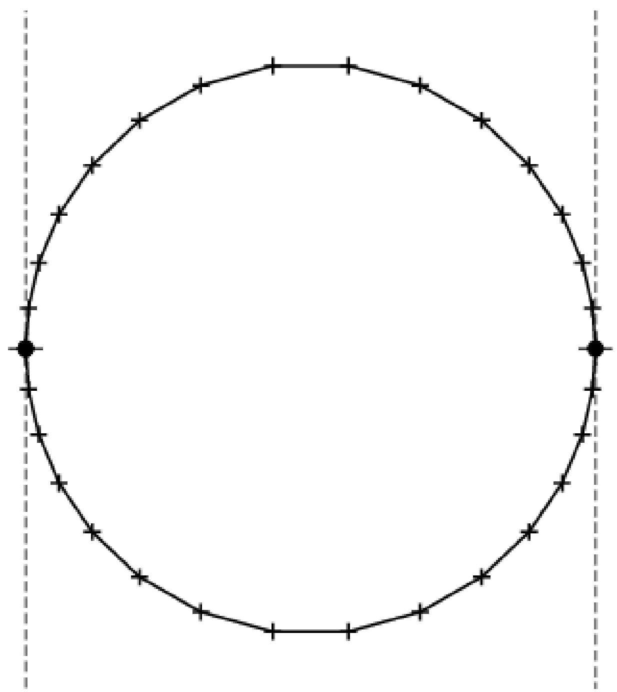
$$P_1^w = \begin{pmatrix} x_1 \cos\alpha/2 \\ y_1 \cos\alpha/2 \\ z_1 \cos\alpha/2 \\ \cos\alpha/2 \end{pmatrix}$$

$$P_2^w = \begin{pmatrix} x_2 \\ y_2 \\ z_2 \\ 1 \end{pmatrix}$$

Computer Aided Design

NURBS

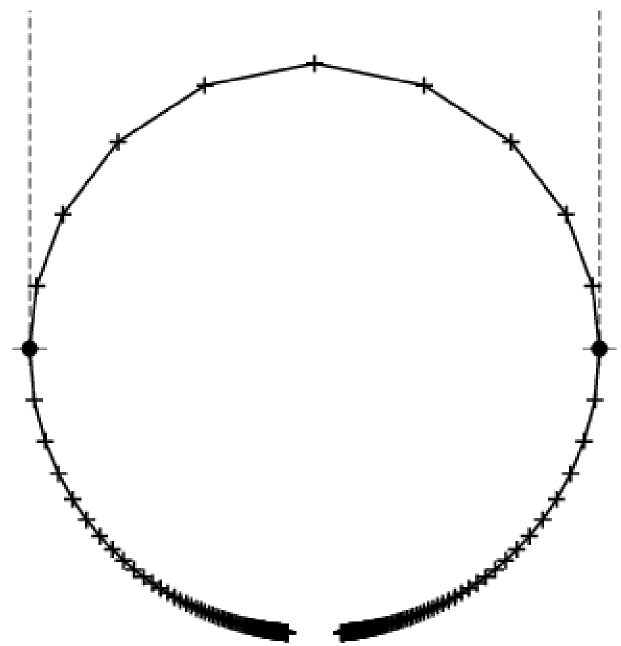
- Can we represent a circle with only 2 pieces ?
  - yes if we allow control points with w=0



$$P_0^w = P_4^w = \begin{bmatrix} -r \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad P_1^w = \begin{bmatrix} 0 \\ r \\ 0 \\ 0 \end{bmatrix} \quad P_2^w = \begin{bmatrix} r \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad P_3^w = \begin{bmatrix} 0 \\ -r \\ 0 \\ 0 \end{bmatrix}$$

$$U = [0,0,0,1,1,2,2,2]$$

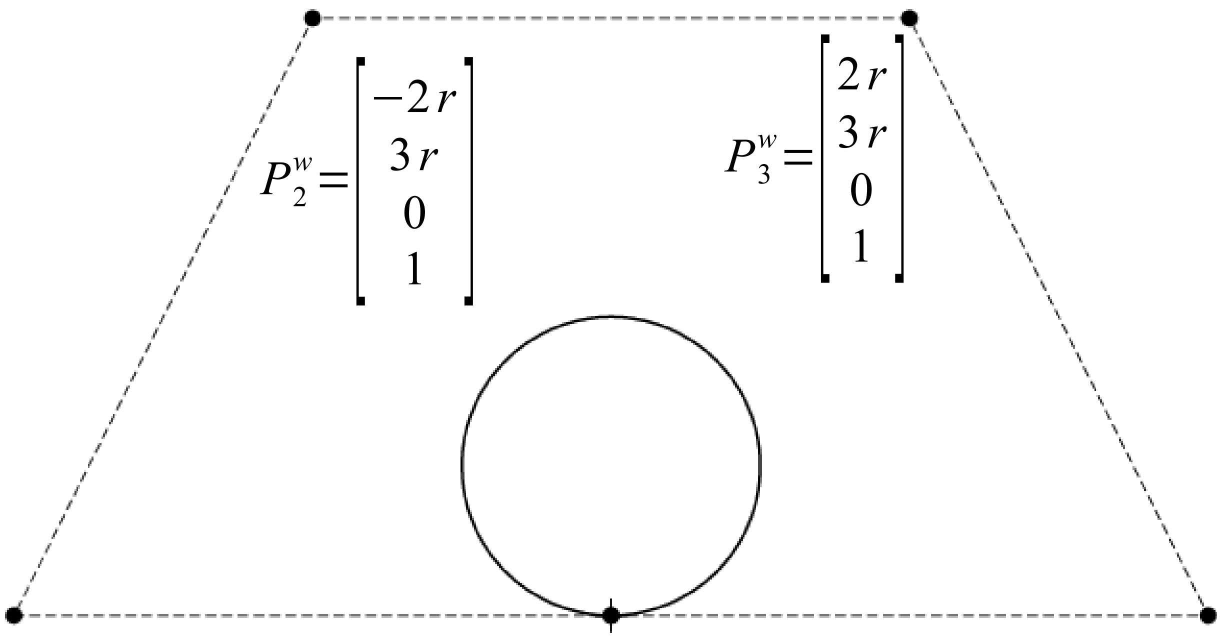
- Can we represent a circle as a single piece ?
  No (if we remain with degree 2)



- Interval for the parameter necessary to cover the whole circle : $\mathbb{R}$
- The nodal sequence is therefore undefined

NURBS

- Can we represent a circle as a single piece ?
  Yes : if we accept to increase the degree (here 5)



$$P_2^w = \begin{bmatrix} -2r \\ 3r \\ 0 \\ 1 \end{bmatrix} \qquad P_3^w = \begin{bmatrix} 2r \\ 3r \\ 0 \\ 1 \end{bmatrix}$$

$$P_1^w = \begin{bmatrix} -4r \\ -r \\ 0 \\ 1 \end{bmatrix} \qquad P_0^w = P_5^w = \begin{bmatrix} 0 \\ -5r \\ 0 \\ 5 \end{bmatrix} \qquad P_4^w = \begin{bmatrix} 4r \\ -r \\ 0 \\ 1 \end{bmatrix}$$

NURBS

- Properties of the NURBS basis functions

$$P(u) = \frac{\sum_i N_i^d(u) P_i w_i}{\sum_j N_j^d(u) w_j} = \frac{P^w(u)}{w(u)} = \sum_i R_i^d(u) P_i$$

$$\text{with} \quad R_i^d(u) = \frac{N_i^d(u) w_i}{\sum_j N_j^d(u) w_j}$$

NURBS

- Partition of the unity: $\sum_i R_i^d(u) = 1$ with $R_i^d(u) = \frac{N_i^d(u) w_i}{\sum_j N_j^d(u) w_j}$

$$\frac{\sum_i N_i^d(u) w_i}{\sum_j N_j^d(u) w_j} \equiv 1 \text{ iff } \sum_j N_j^d(u) w_j \neq 0$$

- In fact, there is always partition of unity with this definition. The original shape functions (N) must however be all linearly independent, and not all equal to zero at any point...

NURBS

- Properties of NURBS basis functions

$$R_i^d(u) = \frac{N_i^d(u)\, w_i}{\sum_j N_j^d(u)\, w_j}$$

  - Non-negativity $R_i^d(u) \geq 0$
  - Partition of the unity $\sum_i R_i^d(u) = 1$
  - If the nodal sequence is not periodic (d+1 repetitions) one have $R_0^d(0) = R_n^d(1) = 1$
  - $R_i^d(u)$ reaches exactly one maximum on the definition interval (for $d>0$)
  - Compact support : $R_i^d(u) = 0 \;\; \text{for} \;\; u \notin [u_i, u_{i+d+1}[$
  - Every derivative exist inside a nodal interval, at one node, SFs are differentiable at least d-k times (k is the multiplicity of a node)
  - If w_i=cst, we have classical non rational curves.

NURBS

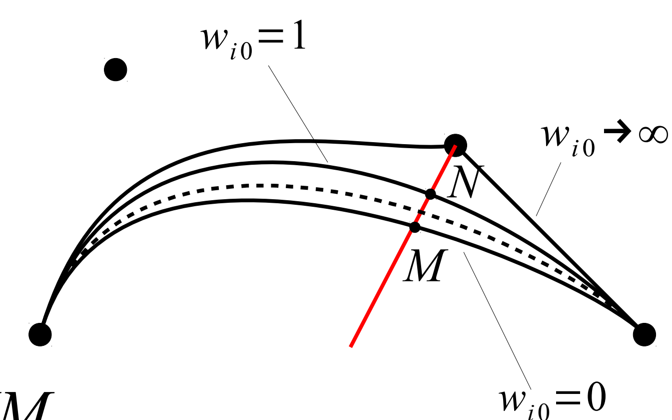
- These properties have the same consequences on the curve as in the classical B-Spline case
  - Beware : the w_i must remain positive. They may vanish iff $w(u)$ remain always **strictly** positive.
- Affine invariance
- Convex hull
- Local control of a given control point
- etc...

NURBS

- Modification of the weight of a control point
  - Does not change the 3D position of the control point (only the distance to the perspective plane in the central projection)
  - Modify the « attraction » of the control point with respect to the curve
  - Means to modify the shape of the curve without moving control points
    - No changes in the geometric continuity of the curve
    - Interesting at the junction of independent curves : the continuity of tangents at the interface is not disturbed – even if the modification is not local (ex: for Bézier curves)

- Location of a given point when modifying a weight...
  - We modify the weight w_{i0} associated to the point P_{i0}
  - Set $N = P(u)\big|_{w_{i0}=1}$

    $M = P(u)\big|_{w_{i0}=0}$



  - The point $P = P(u),\ \ w_{i0} \geq 0$
    is on the straight line joining NM
  - The straight line NM passes by P_{i0} (limit case $w_{i0}\rightarrow\infty$)

    (Also also true for rational surfaces as seen later...)

- Proof

$$P(u)=\frac{\sum_{i\neq i0} N_i^p(u)P_i w_i + N_{i0}^p(u)P_{i0}w_{i0}}{\sum_{k\neq i0} N_k^p(u)w_k + N_{i0}^p(u)w_{i0}}$$

$$P(u)=\frac{P+nP_{i0}w_{i0}}{w+nw_{i0}} \longrightarrow N=\frac{P+nP_{i0}}{w+n} \qquad M=\frac{P}{w} \qquad P=Mw$$

$$N=\frac{Mw+nP_{i0}}{w+n} \qquad P_{i0}=\frac{N(w+n)-Mw}{n}$$

$$P(u)=\frac{P+nP_{i0}w_{i0}}{w+nw_{i0}}=\frac{Mw+(N(w+n)-Mw)w_{i0}}{w+nw_{i0}}=$$

$$N\frac{(w+n)w_{i0}}{w+nw_{i0}}+M\frac{(w-ww_{i0})}{w+nw_{i0}}=Nf(w_{i0})+Mg(w_{i0})$$

- Calculating NURBS derivatives
  - Presence of the rational term : rules of derivation are more complex.
  - Set

$$A(u) = w(u) \cdot P(u)$$

$$P(u) = \frac{A(u)}{w(u)}$$

3 first components of the homogeneous coordinates

last component of the homogeneous coordinates

So :

$$P'(u) = \frac{w(u)\,A'(u) - w'(u)\,A(u)}{w(u)^2} = \frac{w(u)\,A'(u) - w'(u)\,w(u)\,P(u)}{w(u)^2}$$

$$= \frac{A'(u) - w'(u)\,P(u)}{w(u)}$$

NURBS

- Derivatives (general case at the order k)

$$A(u) = w(u) \cdot P(u)$$

$$A^{(k)}(u) = \sum_{i=0}^{k} \binom{k}{i} w^{(i)}(u) \cdot P^{(k-i)}(u)$$

$$= w(u)\, P^{(k)}(u) + \sum_{i=1}^{k} \binom{k}{i} w^{(i)}(u) \cdot P^{(k-i)}(u)$$

$$P^{(k)}(u) = \frac{A^{(k)}(u) - \sum_{i=1}^{k} \binom{k}{i} w^{(i)}(u) \cdot P^{(k-i)}(u)}{w(u)}$$