Advanced Graphics

Beziers, B-splines, and NURBS

Alex Benton, University of Cambridge – A.Benton@damtp.cam.ac.uk

Supported in part by Google UK, Ltd

Bezier splines, B-Splines, and NURBS

Expensive products are sleek and smooth.

→ Expensive products are C2 continuous.

Shiny, but reflections are warped

Shiny, and reflections are perfect

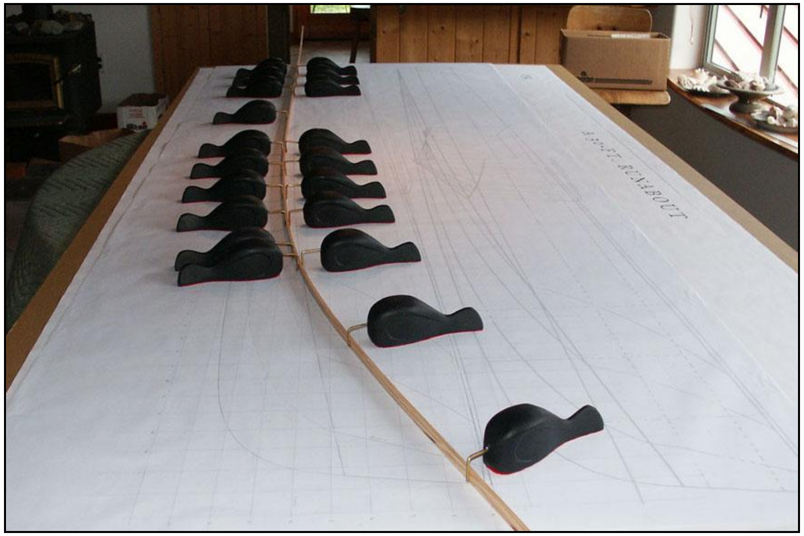
History

- *Continuity* (smooth curves) can be essential to the perception of *quality*.
- The automotive industry wanted to design cars which were aerodynamic, but also visibly of high quality.
- Bezier (Renault) and de Casteljau (Citroen) invented Bezier curves in the 1960s. de Boor (GM) generalized them to B-splines.

History

The term *spline* comes from the shipbuilding industry: long, thin strips of wood or metal would be bent and held in place by heavy ‘ducks’, lead weights which acted as control points of the curve.

Wooden splines can be described by C_n-continuous Hermite polynomials which interpolate n+1 control points.

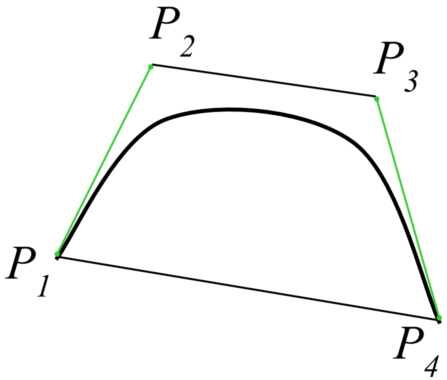


Top: Fig 3, P.7, Bray and Spectre, *Planking and Fastening*, Wooden Boat Pub (1996)

Bottom: http://www.pranos.com/boatsofwood/lofting%20ducks/lofting_ducks.htm

Beziers—a quick review

- A Bezier cubic is a function P(t) defined by four control points:
  - P_1 and P_4 are the endpoints of the curve
  - P_2 and P_3 define the other two corners of the bounding polygon.
- The curve fits entirely within the convex hull of $P_1 \ldots P_4$.
- A degree-*d* Bezier is infinitely continuous throughout its interior. However, when joining two Beziers, careful placement of the control points is required to ensure continuity.



Cubic: $P(t) = (1-t)^3 P_1 + 3t(1-t)^2 P_2 + 3t^2(1-t) P_3 + t^3 P_4$

Beziers

Cubics are just one example of Bezier splines:

- Linear: $P(t) = (1\text{-}t)P_1 + tP_2$
- Quadratic: $P(t) = (1\text{-}t)^2P_1 + 2t(1\text{-}t)P_2 + t^2P_3$
- Cubic: $P(t) = (1\text{-}t)^3P_1 + 3t(1\text{-}t)^2P_2 + 3t^2(1\text{-}t)P_3 + t^3P_4$

...

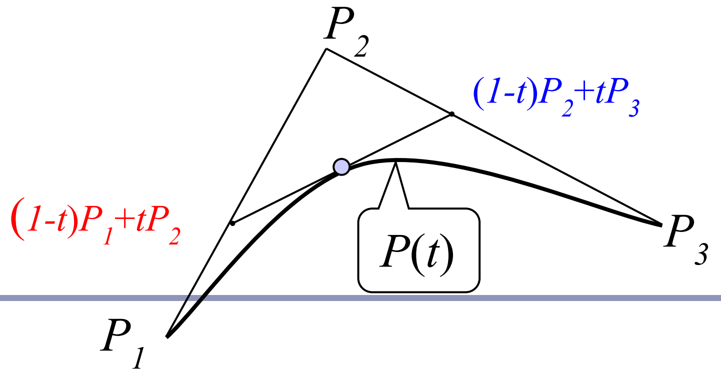
General:

"n choose i" = n! / i!(n-i)!

$$P(t) = \sum_{i=0}^{n} \binom{n}{i} (1-t)^{n-i} t^i P_i, \; 0 \leq t \leq 1$$

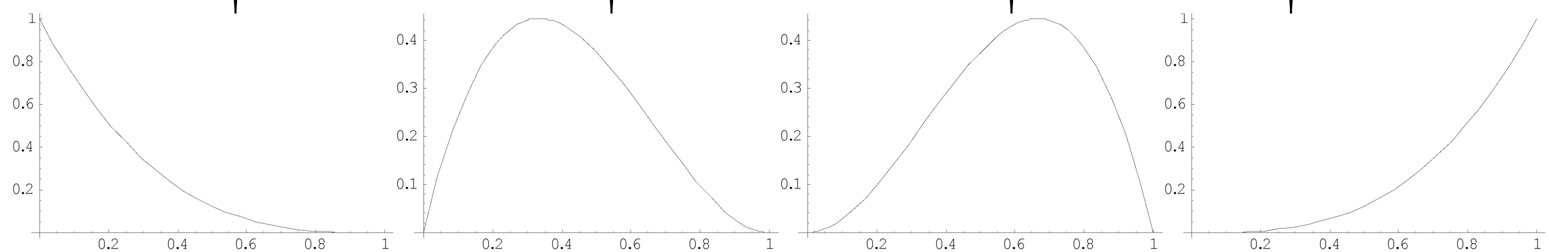
Beziers

- You can describe Beziers as *nested linear interpolations:*
  - The linear Bezier is a linear interpolation between two points:
    $P(t) = (1-t)\ (P_1) + (t)\ (P_2)$
  - The quadratic Bezier is a linear interpolation between two lines:
    $P(t) = (1-t)\ ((1-t)P_1+tP_2) + (t)\ ((1-t)P_2+tP_3)$
  - The cubic is a linear interpolation between linear interpolations between linear interpolations… etc.
- Another way to see Beziers is as a *weighted average* between the control points.



Bernstein polynomials

$$P(t) = (1-t)^3 P_1 + 3t(1-t)^2 P_2 + 3t^2(1-t) P_3 + t^3 P_4$$

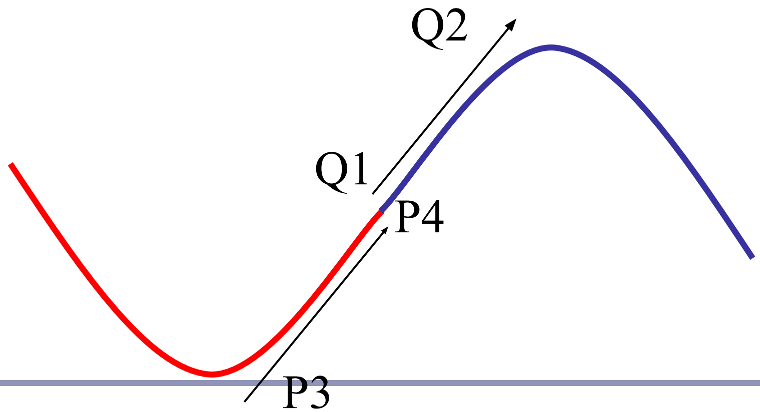


- The four control functions are the four *Bernstein polynomials* for $n=3$.
  - General form: $b_{v,n}(t) = \binom{n}{v} t^v (1-t)^{n-v}$
  - Bernstein polynomials in $0 \leq t \leq 1$ always sum to 1:

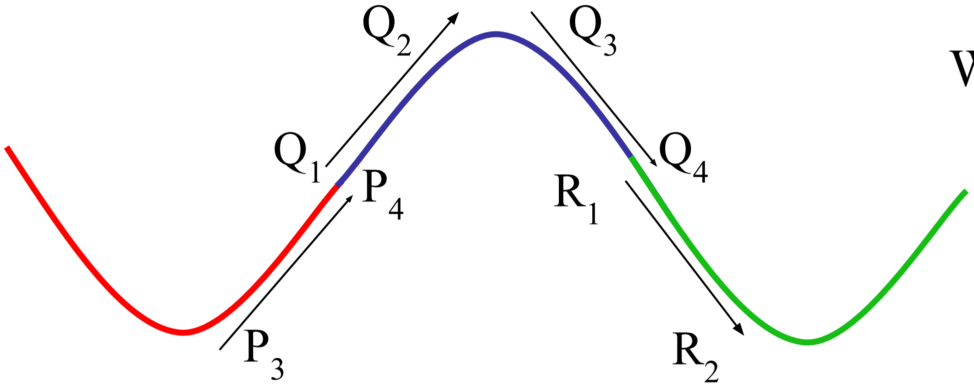
$$\sum_{v=1}^{n} \binom{n}{v} t^v (1-t)^{n-v} = (t + (1-t))^n = 1$$

Joining Bezier splines

- To join two Bezier splines with C0 continuity, set $P_4 = Q_1$.
- To join two Bezier splines with C1 continuity, require C0 and make the tangent vectors equal: set $P_4 = Q_1$ and $P_4 - P_3 = Q_2 - Q_1$.



What if we want to chain Beziers together?



We can parameterize this chain over *t* by saying that instead of going from 0 to 1, *t* moves smoothly through the intervals [0,1,2,3]

Consider a chain of splines with many control points…

$P = \{P_0, P_1, P_2, P_3\}$
$Q = \{Q_0, Q_1, Q_2, Q_3\}$
$R = \{R_0, R_1, R_2, R_3\}$

…with C1 continuity…

$P3=Q_0$, $P_2-P_3=Q_0-Q_1$
$Q3=R_0$, $Q_2-Q_3=R_0-R_1$

The curve *C*(*t*) would be:

$$C(t) = P(t) \cdot ((0 \le t < 1)\ ?\ 1 : 0) + Q(t-1) \cdot ((1 \le t < 2)\ ?\ 1 : 0) + R(t-2) \cdot ((2 \le t < 3)\ ?\ 1 : 0)$$

[0,1,2,3] is a type of *knot vector*. 0, 1, 2, and 3 are the *knots*.

NURBS

- *NURBS* ("*Non-Uniform Rational B-Splines*") are a generalization of Beziers.
  - NU: *Non-Uniform*. The knots in the knot vector are not required to be uniformly spaced.
  - R: *Rational*. The spline may be defined by rational polynomials (homogeneous coordinates.)
  - BS: *B-Spline*. A generalization of Bezier splines with controllable degree.

B-Splines

- A Bezier cubic is a polynomial of degree three: it must have four control points, it must begin at the first and end at the fourth, and it assumes that all four control points are equally important.

- *B-spline* curves are a piecewise parameterization of a series of splines, that supports an arbitrary number of control points and lets you specify the degree of the polynomial which interpolates them.

B-Splines

We'll build our definition of a B-spline from:

- d, the *degree* of the curve
- $k = d+1$, called the *parameter* of the curve
- $\{P_1 \ldots P_n\}$, a list of *n control points*
- $[t_1, \ldots, t_{k+n}]$, a *knot vector* of (k+n) parameter values

- $d = k\text{-}1$ is the degree of the curve, so k is the number of control points which influence a single interval.
  - Ex: a cubic (d=3) has four control points (k=4).
- There are $k+n$ knots, and $t_i \leq t_{i+1}$ for all t_i.
- Each B-spline is $C^{(k-2)}$ continuous: *continuity* is degree minus one, so a k=3 curve has d=2 and is C1.

B-Splines

- The equation for a B-spline curve is

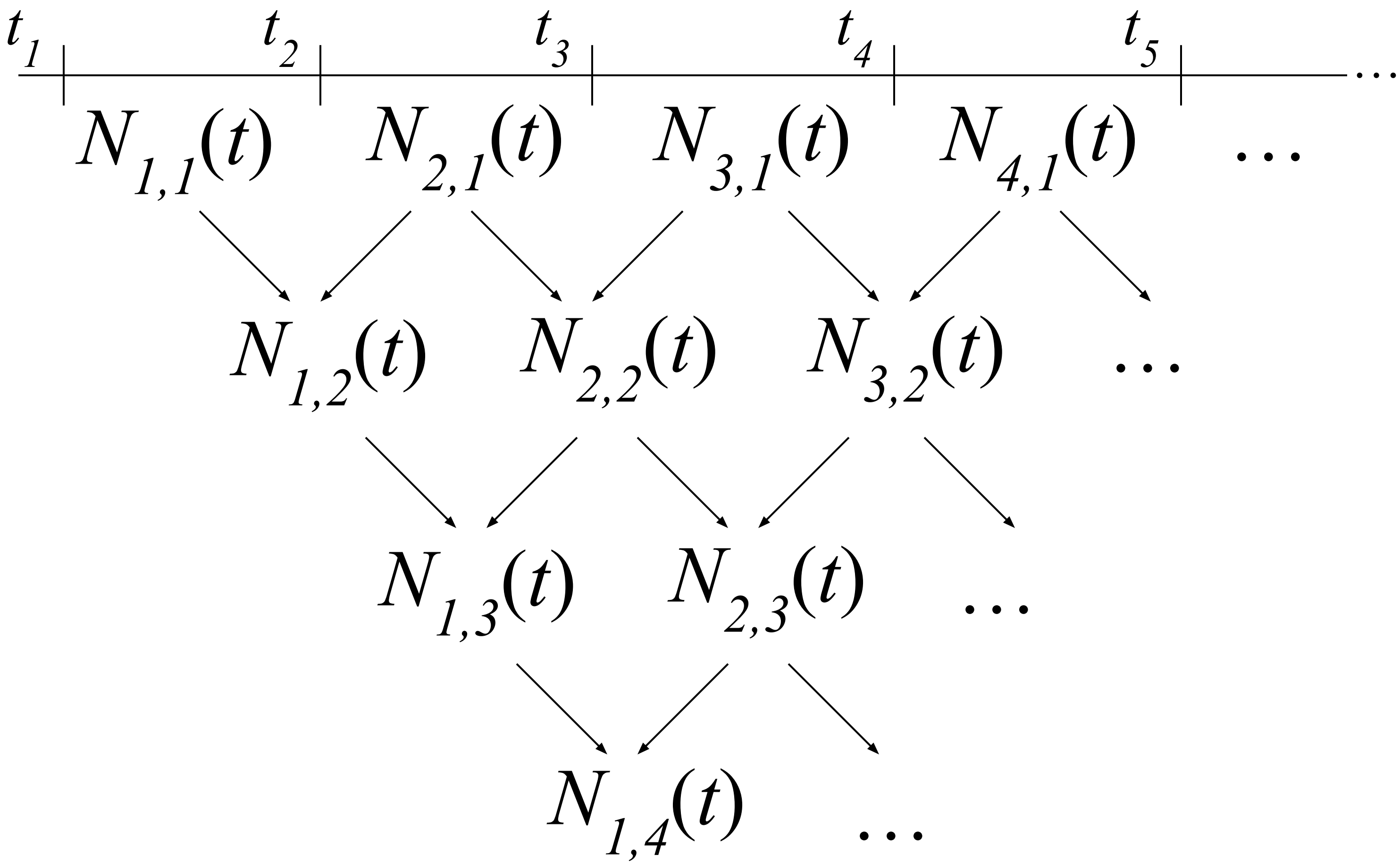
$$P(t) = \sum_{i=1}^{n} N_{i,k}(t) P_i, \; t_{min} \leq t < t_{max}$$

- $N_{i,k}(t)$ is the *basis function* of control point P_i for parameter k. $N_{i,k}(t)$ is defined recursively:

$$N_{i,1}(t) = \begin{cases} 1, & t_i \leq t < t_{i+1} \\ 0, & \text{otherwise} \end{cases}$$

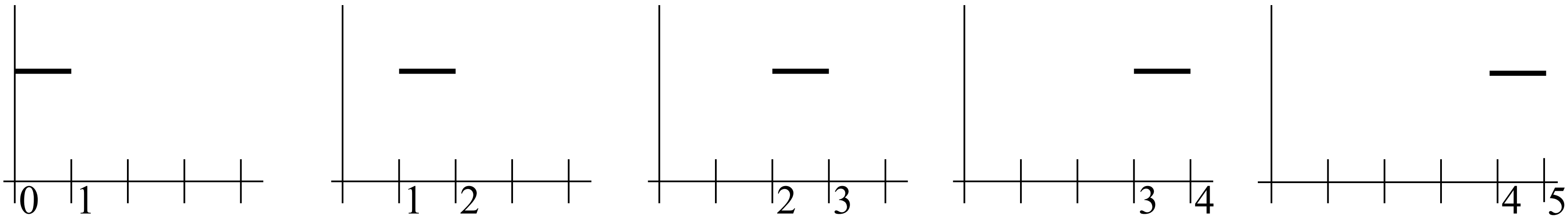
$$N_{i,k}(t) = \frac{t - t_i}{t_{i+k-1} - t_i} N_{i,k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} N_{i+1,k-1}(t)$$

B-Splines


t_1 t_2 t_3 t_4 t_5 …
$N_{1,1}(t)$ $N_{2,1}(t)$ $N_{3,1}(t)$ $N_{4,1}(t)$ …
$N_{1,2}(t)$ $N_{2,2}(t)$ $N_{3,2}(t)$ …
$N_{1,3}(t)$ $N_{2,3}(t)$ …
$N_{1,4}(t)$ …

B-Splines

$$N_{i,1}(t) = \begin{cases} 1, & t_i \le t < t_{i+1} \\ 0, & \text{otherwise} \end{cases}$$



$N_{1,1}(t)$ $N_{2,1}(t)$ $N_{3,1}(t)$ $N_{4,1}(t)$ $N_{5,1}(t)$

$t_1 = 0.0$
$t_2 = 1.0$
$t_3 = 2.0$
$t_4 = 3.0$
$t_5 = 4.0$
$t_6 = 5.0$

$N_{1,1}(t)=1,\ 0 \le t < 1$

$N_{3,1}(t)=1,\ 2 \le t < 3$

$N_{5,1}(t)=1,\ 4 \le t < 5$

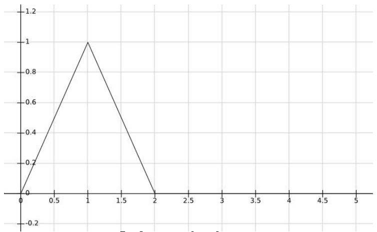
$N_{2,1}(t)=1,\ 1 \le t < 2$

$N_{4,1}(t)=1,\ 3 \le t < 4$

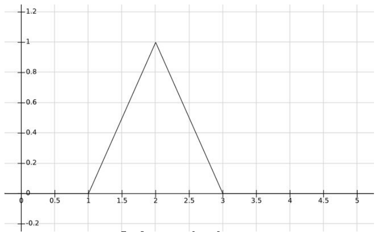
Knot vector = {0,1,2,3,4,5}, $k = 1 \rightarrow d = 0$ (degree = zero)

B-Splines

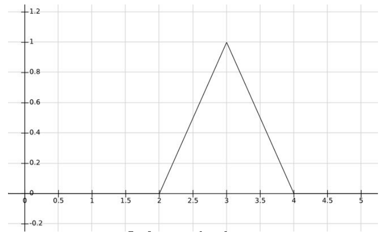
$$N_{i,k}(t) = \frac{t - t_i}{t_{i+k-1} - t_i} N_{i,k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} N_{i+1,k-1}(t)$$



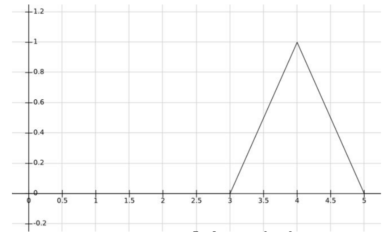
$N_{1,2}(t)$



$N_{2,2}(t)$



$N_{3,2}(t)$



$N_{4,2}(t)$

$$N_{1,2}(t) = \frac{t-0}{1-0} N_{1,1}(t) + \frac{2-t}{2-1} N_{2,1}(t) = \begin{cases} t & 0 \le t < 1 \\ 2-t & 1 \le t < 2 \end{cases}$$

$$N_{2,2}(t) = \frac{t-1}{2-1} N_{2,1}(t) + \frac{3-t}{3-2} N_{3,1}(t) = \begin{cases} t-1 & 1 \le t < 2 \\ 3-t & 2 \le t < 3 \end{cases}$$

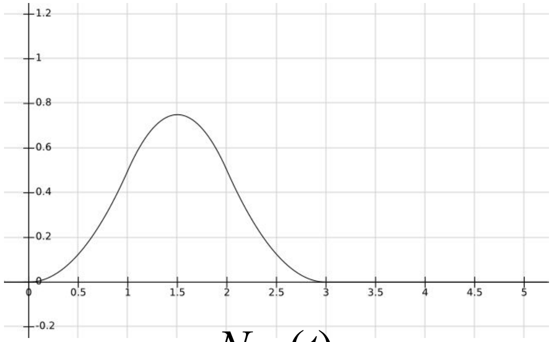
$$N_{3,2}(t) = \frac{t-2}{3-2} N_{3,1}(t) + \frac{4-t}{4-3} N_{4,1}(t) = \begin{cases} t-2 & 2 \le t < 3 \\ 4-t & 3 \le t < 4 \end{cases}$$

$$N_{4,2}(t) = \frac{t-3}{4-3} N_{4,1}(t) + \frac{5-t}{5-4} N_{5,1}(t) = \begin{cases} t-3 & 3 \le t < 4 \\ 5-t & 4 \le t < 5 \end{cases}$$

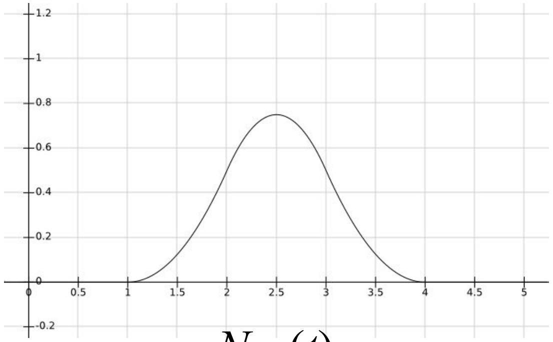
Knot vector = {0,1,2,3,4,5}, $k = 2 \rightarrow d = 1$ (degree = one)

B-Splines

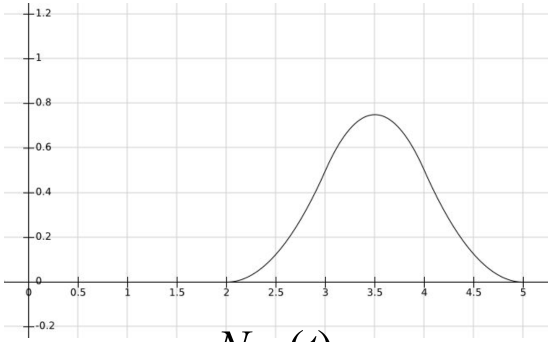
$$N_{i,k}(t) = \frac{t - t_i}{t_{i+k-1} - t_i} N_{i,k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} N_{i+1,k-1}(t)$$



$N_{1,3}(t)$



$N_{2,3}(t)$



$N_{3,3}(t)$

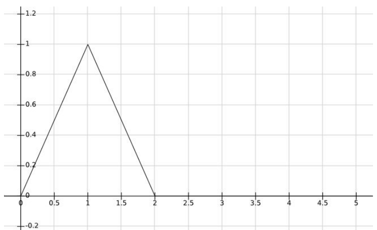
$$N_{1,3}(t) = \frac{t-0}{2-0} N_{1,2}(t) + \frac{3-t}{3-1} N_{2,2}(t) = \begin{cases} t^2/2 & 0 \le t < 1 \\ -t^2 + 3t - 3/2 & 1 \le t < 2 \\ (3-t)^2/2 & 2 \le t < 3 \end{cases}$$

$$N_{2,3}(t) = \frac{t-1}{3-1} N_{2,2}(t) + \frac{4-t}{4-2} N_{3,2}(t) = \begin{cases} (t-1)^2/2 & 1 \le t < 2 \\ -t^2 + 5t - 11/2 & 2 \le t < 3 \\ (4-t)^2/2 & 3 \le t < 4 \end{cases}$$

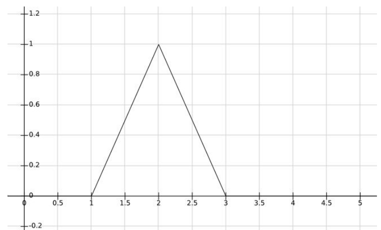
$$N_{3,3}(t) = \frac{t-2}{4-2} N_{3,2}(t) + \frac{5-t}{5-3} N_{4,2}(t) = \begin{cases} (t-2)^2/2 & 2 \le t < 3 \\ -t^2 + 7t - 23/2 & 3 \le t < 4 \\ (5-t)^2/2 & 4 \le t < 5 \end{cases}$$

Knot vector = {0,1,2,3,4,5}, $k = 3 \rightarrow d = 2$ (degree = two)

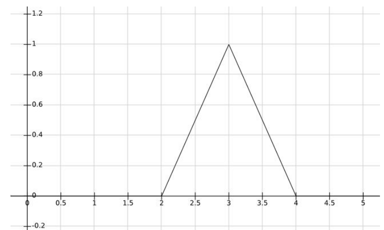
Basis functions really sum to one (k=2)



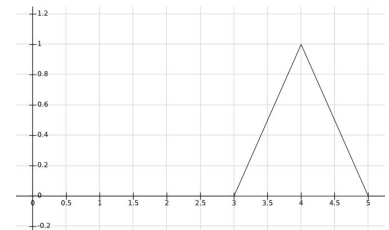
$N_{1,2}(t)$



$N_{2,2}(t)$

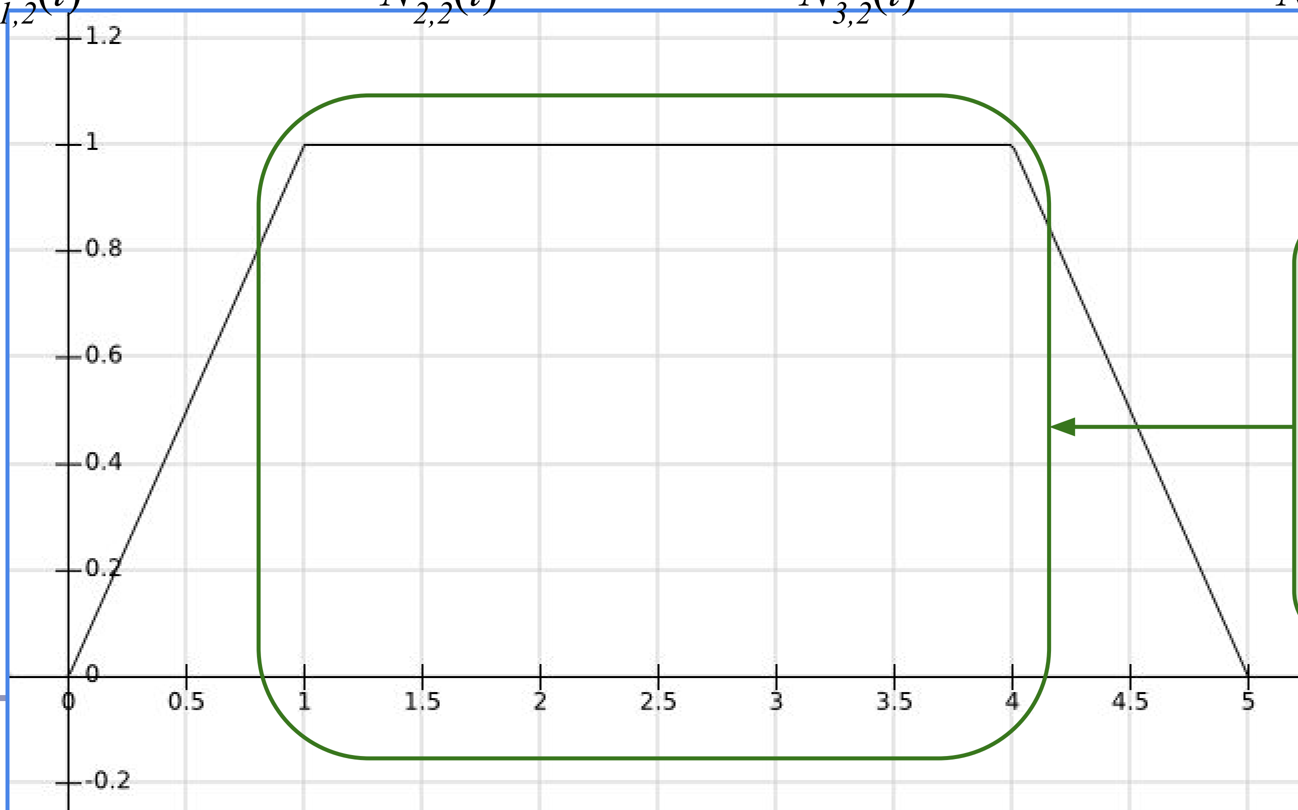


$N_{3,2}(t)$



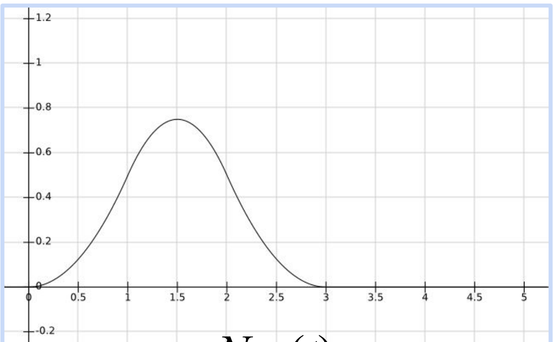
$N_{4,2}(t)$

=



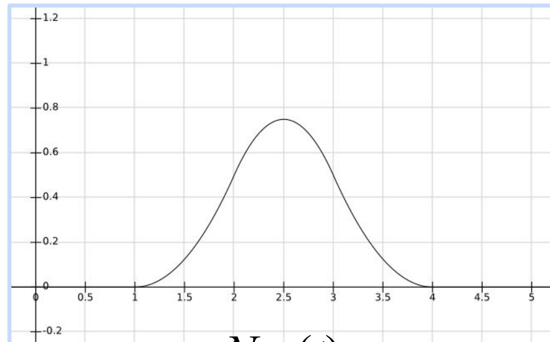
The sum of the four basis functions is fully defined (sums to one) between t_2 (t=1.0) and t_5 (t=4.0).

Basis functions really sum to one (k=3)



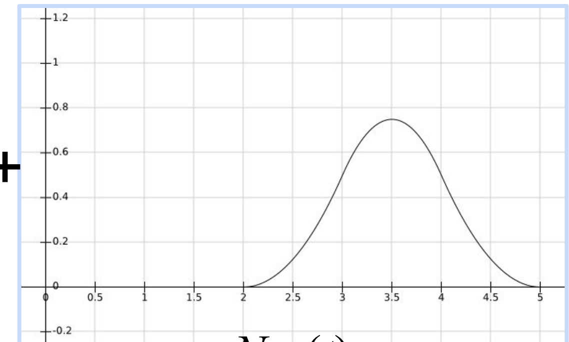
$N_{1,3}(t)$

+



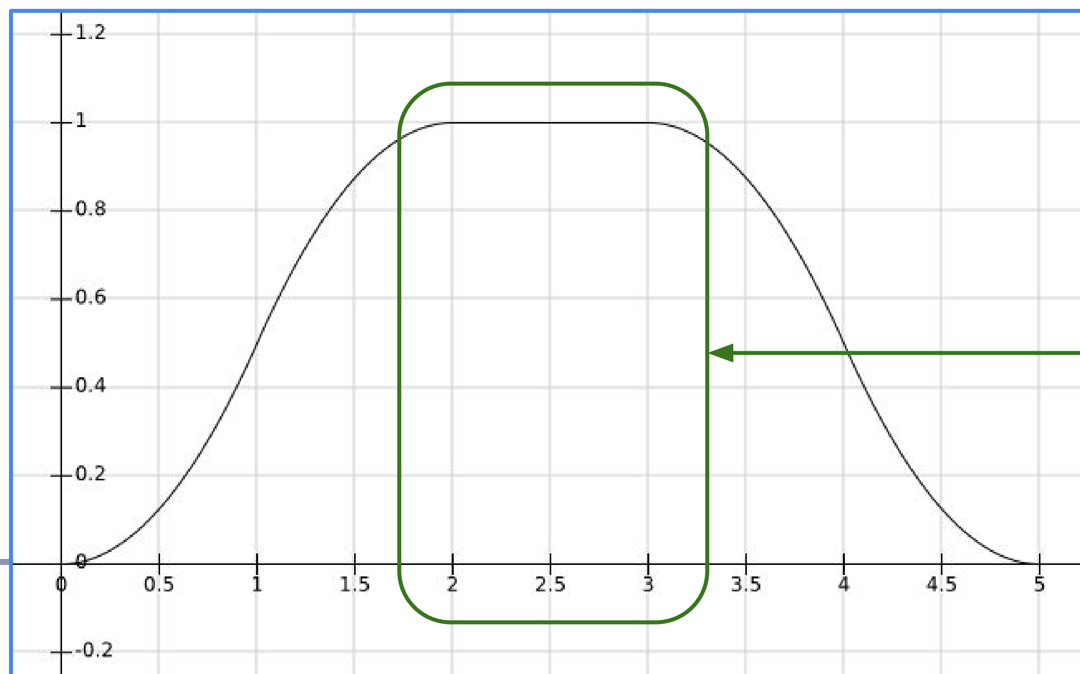
$N_{2,3}(t)$

+



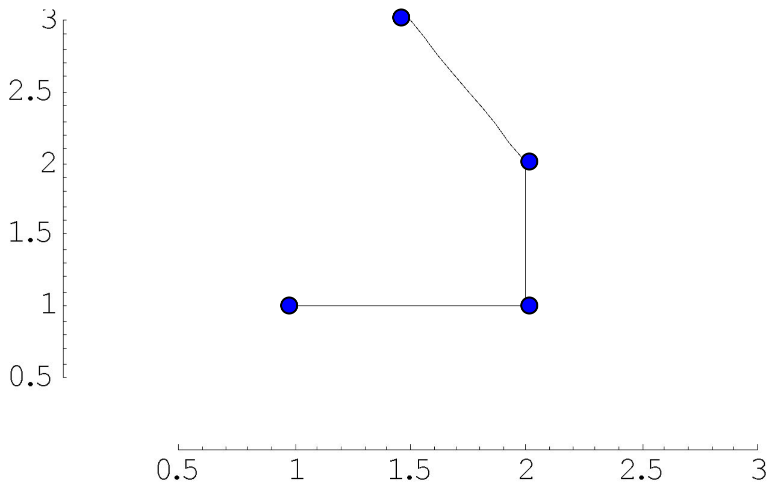
$N_{3,3}(t)$

=

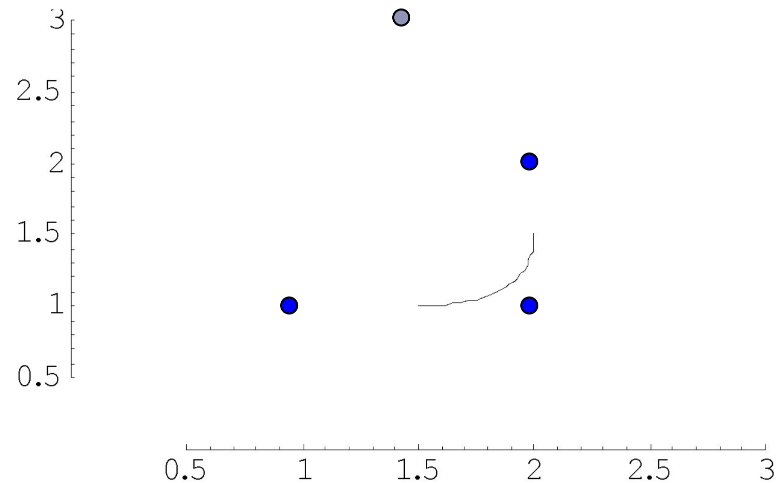


The sum of the three functions is fully defined (sums to one) between t_3 (t=2.0) and t_4 (t=3.0).

B-Splines



At k=2 the function is piecewise linear, depends on P_1,P_2,P_3,P_4, and is fully defined on $[t_2, t_5)$.



At k=3 the function is piecewise quadratic, depends on P_1,P_2,P_3, and is fully defined on $[t_3, t_4)$.

Each parameter-k basis function depends on $k+1$ knot values; $N_{i,k}$ depends on t_i through t_{i+k}, inclusive. So six knots → five discontinuous functions → four piecewise linear interpolations → three quadratics, interpolating three control points. n=3 control points, d=2 degree, k=3 parameter, n+k=6 knots.

Knot vector = {0,1,2,3,4,5}

Non-Uniform B-Splines

- The knot vector {0,1,2,3,4,5} is *uniform*:

  $$t_{i+1} - t_i = t_{i+2} - t_{i+1} \quad \forall\, t_i.$$

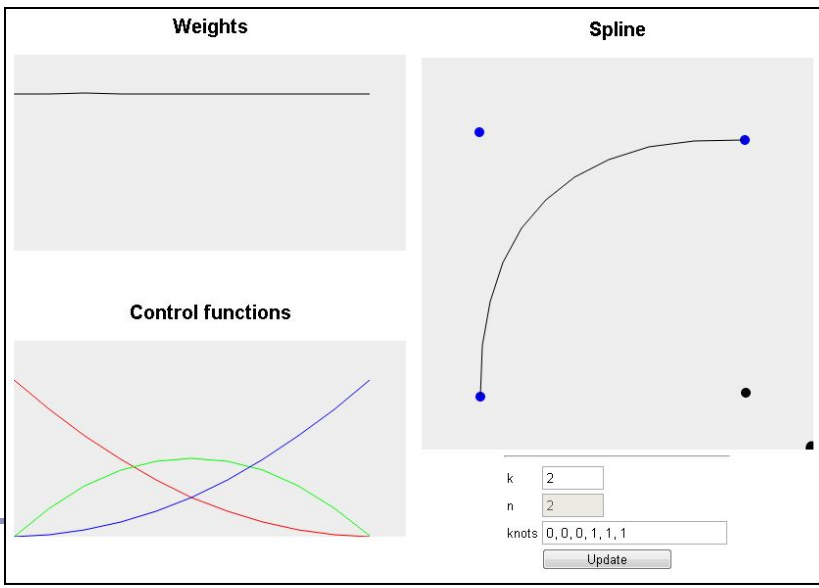
- Varying the size of an interval changes the parametric-space distribution of the weights assigned to the control functions.
- Repeating a knot value reduces the continuity of the curve in the affected span by one degree.
- Repeating a knot *k* times will lead to a control function being influenced only by that knot value; the spline will pass through the corresponding control point with C0 continuity.

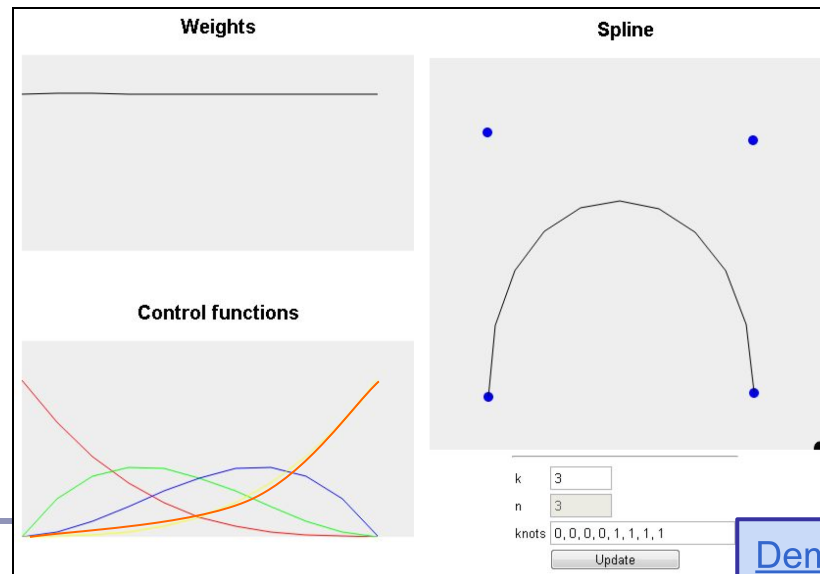
Open vs *Closed*

- A knot vector which repeats its first and last knot values k times is called *open*, otherwise *closed*.
  - Repeating the knots k times is the only way to force the curve to pass through the first or last control point.
  - Without this, the functions $N_{1,k}$ and $N_{n,k}$ which weight P_1 and P_n would still be 'ramping up' and not yet equal to one at the first and last t_i.

Open vs Closed

- Two examples you may recognize:
  - k=3, n=3 control points, knots={0,0,0,1,1,1}
  - k=4, n=4 control points, knots={0,0,0,0,1,1,1,1}





Demo

Non-Uniform *Rational* B-Splines

- Repeating knot values is a clumsy way to control the curve's proximity to the control point.
  - We want to be able to slide the curve nearer or farther without losing continuity or introducing new control points.
  - The solution: *homogeneous coordinates*.
  - Associate a 'weight' with each control point: ω_i.

Non-Uniform Rational B-Splines

- Recall: $[x, y, z, \omega]_H \rightarrow [x / \omega, y / \omega, z / \omega]$
  - Or: $[x, y, z, 1] \rightarrow [x\omega, y\omega, z\omega, \omega]_H$
- The control point

  $P_i = (x_i, y_i, z_i)$

  becomes the homogeneous control point

  $P_{iH} = (x_i\omega_i, y_i\omega_i, z_i\omega_i)$
- A NURBS in homogeneous coordinates is:

$$P_H(t) = \sum_{i=1}^{n} N_{i,k}(t) P_{iH}, \; t_{min} \le t < t_{max}$$

Non-Uniform Rational B-Splines

- To convert from homogeneous coords to normal coordinates:

$$
\begin{aligned}
x_H(t) &= \sum_{i=1}^{n}(x_i\omega_i)(N_{i,k}(t)) \\
y_H(t) &= \sum_{i=1}^{n}(y_i\omega_i)(N_{i,k}(t)) \\
z_H(t) &= \sum_{i=1}^{n}(z_i\omega_i)(N_{i,k}(t)) \\
\omega(t) &= \sum_{i=1}^{n}(\omega_i)(N_{i,k}(t))
\end{aligned}
$$

$$
\begin{aligned}
x(t) &= x_H(t)/\omega(t) \\
y(t) &= y_H(t)/\omega(t) \\
z(t) &= z_H(t)/\omega(t)
\end{aligned}
$$

Non-Uniform Rational B-Splines

- A piecewise rational curve is thus defined by:

$$P(t) = \sum_{i=1}^{n} R_{i,k}(t) P_i, \; t_{min} t < t_{max}$$

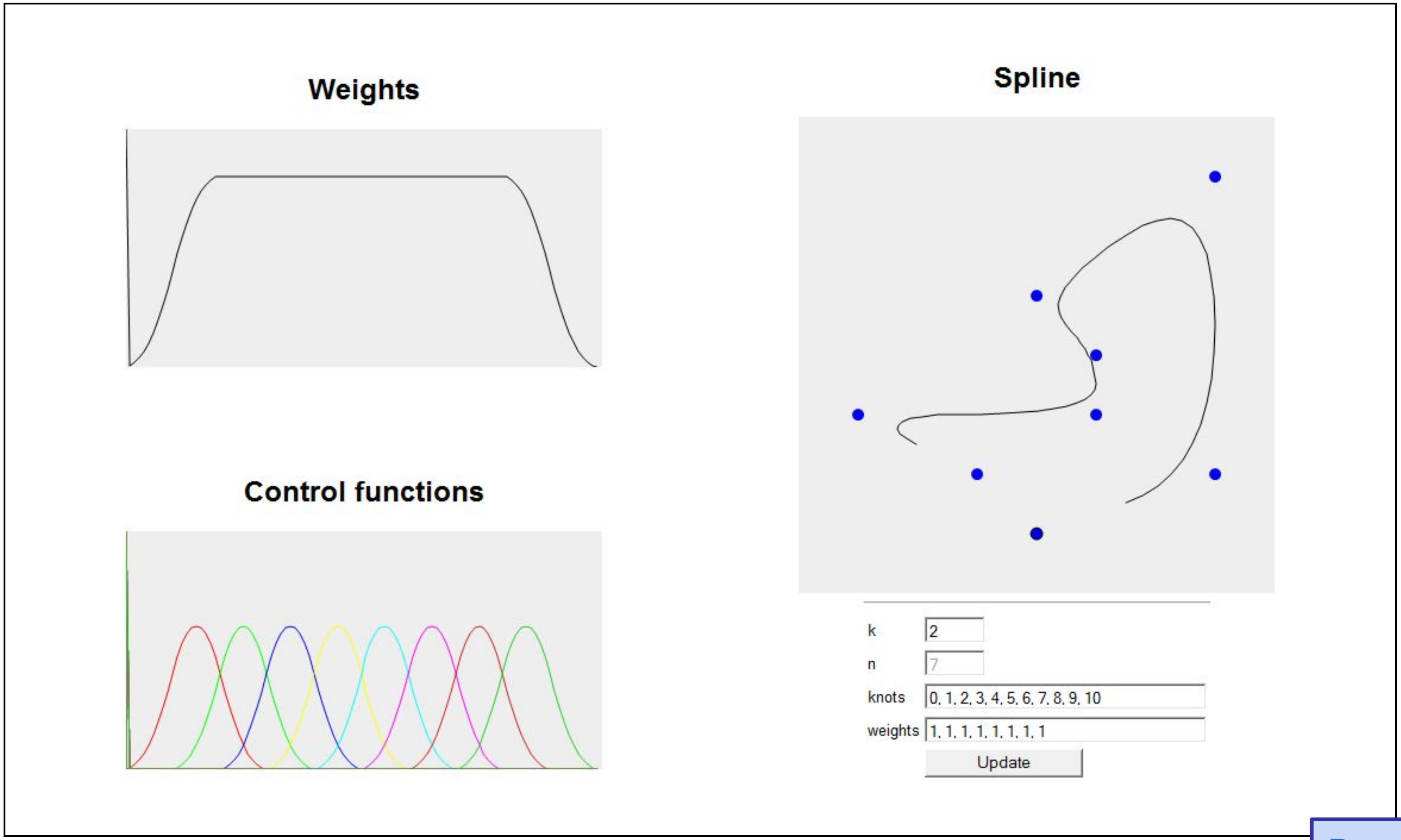
with supporting *rational basis functions*:

$$R_{i,k}(t) = \frac{\omega_i N_{i,k}(t)}{\sum_{j=1}^{n} \omega_j N_{j,k}(t)}$$

This is essentially an average re-weighted by the ω's.

- Such a curve can be made to pass arbitrarily far or near to a control point by changing the corresponding weight.

Non-Uniform Rational B-Splines in action



Demo

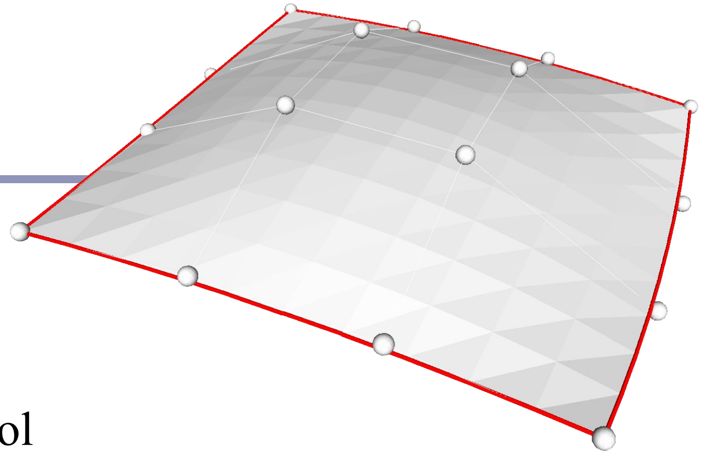
Tensor product

- The *tensor product* of two vectors is a matrix.

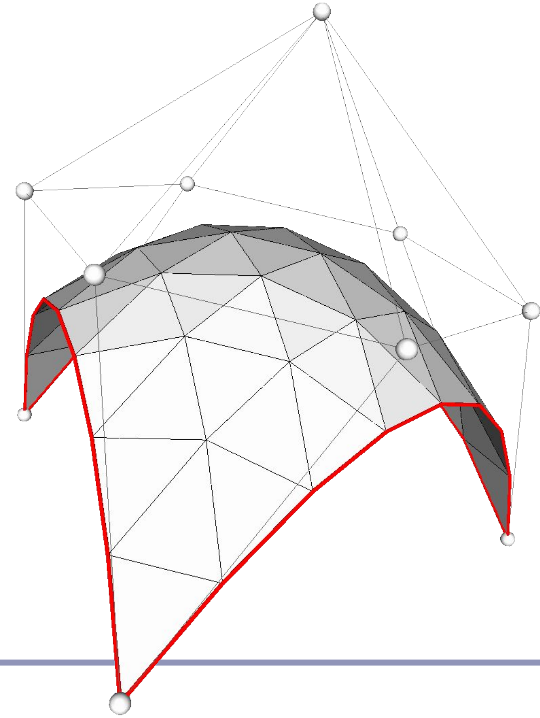
$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} \otimes \begin{bmatrix} d \\ e \\ f \end{bmatrix} = \begin{bmatrix} ad & ae & af \\ bd & be & bf \\ cd & ce & cf \end{bmatrix}$$

- Can also take the tensor of two polynomials.
  - Each coefficient represents a piece of each of the two original expressions, to the cumulative polynomial represents both original polynomials completely.

NURBS patches



- The tensor product of the polynomial coefficients of two NURBS splines is a matrix of polynomial coefficients.
  - If curve A has parameter *k* and *n* control points and curve B has parameter *j* and *m* control points then A⊗B is an $(n) \times (m)$ matrix of polynomials of parameter *max* (*j*,*k*).
- Multiply this matrix against an $(n) \times (m)$ matrix of control points and sum them all up and you've got a bivariate expression for a rectangular surface patch, in 3D
- This approach generalizes to triangles and arbitrary *n*-gons.



References

- Les Piegl and Wayne Tiller, *The NURBS Book*, Springer (1997)
- Alan Watt, *3D Computer Graphics*, Addison Wesley (2000)
- G. Farin, J. Hoschek, M.-S. Kim, *Handbook of Computer Aided Geometric Design*, North-Holland (2002)