Bernried, Germany, February 29, 2016

Designing Curves and Surfaces :
Least square Bézier or B-spline curves

Christophe Rabut
INSA (IREM, IMT and MAIAA) ,Toulouse, France

Main idea of the talk

Bézier, B-spline curves, or surfaces :

For a given control polygon,

derive curves closer (or further) to the control polygon,

still being in the same vectorial space.

Context, Definitions and Notations

- **"B-curve" :**

"(generalized) Bézier curve" associated to "B-functions" $(B_i)_{i=0:n}$ and to the "control points" ("polygon") $(P_i)_{i=0:n} = (x_i, y_i)_{i=0:n}$

$$C(t) \;=\; \sum_{i=0:n} P_i\, B_i(t) \;=\; \begin{cases} \sum_{i=0:n} x_i\, B_i(t) \\ \sum_{i=0:n} y_i\, B_i(t) \end{cases}$$

- **"B-functions" :** Bernstein polynomials
  B-splines, NURBS
  ECC systems

Context, Definitions and Notations

- **"B-curve" :**

"(generalized) Bézier curve" associated to "B-functions" $(B_i)_{i=0:n}$ and to the "control points" ("polygon") $(P_i)_{i=0:n} = (x_i, y_i)_{i=0:n}$

$$C(t) = \sum_{i=0:n} P_i B_i(t) = \begin{cases} \sum_{i=0:n} x_i B_i(t) \\ \sum_{i=0:n} y_i B_i(t) \end{cases}$$

- **"B-functions" :** Bernstein polynomials
  B-splines, NURBS
  ECC systems
- **"(generalized) Bézier surface" :**

$$S(u,v) = \sum_{i=0:n} P_i B_i(u,v) = \begin{cases} \sum_{i=0:n} x_i B_i(u,v) \\ \sum_{i=0:n} y_i B_i(u,v) \\ \sum_{i=0:n} z_i B_i(u,v) \end{cases}$$

- B_i : – Tensorial product of univariate B-functions
  – "Bernstein polynomials" on triangles
  – Polyharmonic B-splines, ...

“The “shape“ of the curve is connected to its distance to the control points”

Usually we say :

- Bézier curves (Bernstein) $\Longrightarrow$ quite far from control polygon
- degree 2, 3, 4 splines $\Longrightarrow$ further and further the control points/control polygon, **and** $\mathcal{C}^1$, $\mathcal{C}^2$, $\mathcal{C}^3$ curves...
- intermediate : fractional splines (non polynomial), shifted splines, B-splines under tension...

“The “shape“ of the curve is connected to its distance to the control points”

Usually we say :

- Bézier curves (Bernstein) $\Longrightarrow$ quite far from control polygon
- degree 2, 3, 4 splines $\Longrightarrow$ further and further the control points/control polygon, **and** $\mathcal{C}^1$, $\mathcal{C}^2$, $\mathcal{C}^3$ curves...
- intermediate : fractional splines (non polynomial), shifted splines, B-splines under tension...
- Hyperbolic Chebyshevian splines : (pol. + sinh + cosh) closer to control polygon (also depends on the degree)
- Circular Chebyshevian splines : (pol. + cos + sin) further to control polygon (also depends on the degree)

"The "shape" of the curve is connected to its distance to the control points"

Usually we say :

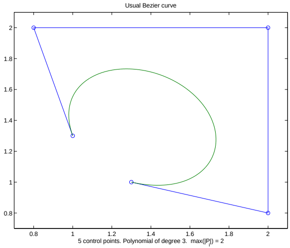
- Bézier curves (Bernstein) $\Longrightarrow$ quite far from control polygon
- degree 2, 3, 4 splines $\Longrightarrow$ further and further the control points/control polygon, **and** $\mathcal{C}^1$, $\mathcal{C}^2$, $\mathcal{C}^3$ curves...
- intermediate : fractional splines (non polynomial),
  shifted splines, B-splines under tension...
- Hyperbolic Chebyshevian splines : (pol. + sinh + cosh)
  closer to control polygon (also depends on the degree)
- Circular Chebyshevian splines : (pol. + cos + sin)
  further to control polygon (also depends on the degree)

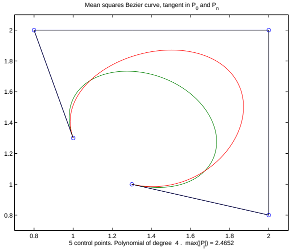
So the distance of a curve to the control polygon is very connected to the analytical properties (continuity, vectorial space...) of the curve.

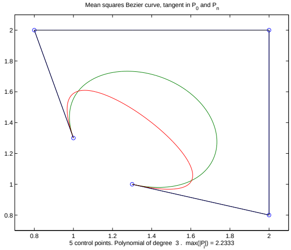
Could we disconnect these two important properties ??

A glance on some results

Usual Bézier curve Closer the control polygon Further the control polygon



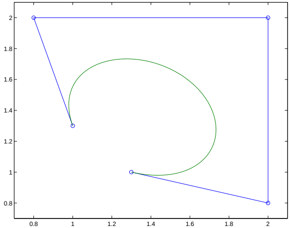




A glance on some results

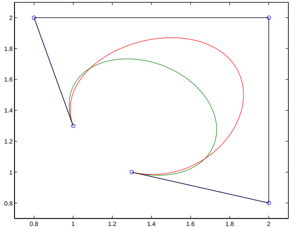
Usual Bézier curve Closer the control polygon Further the control polygon

Usual Bezier curve



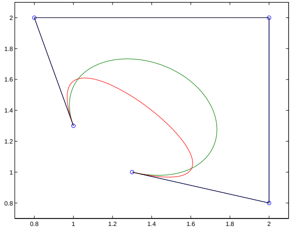
5 control points. Polynomial of degree 3. max(|P_i|) = 2

Mean squares Bezier curve, tangent in P_0 and P_n



5 control points. Polynomial of degree 4 . max(|P_i|) = 2.4652

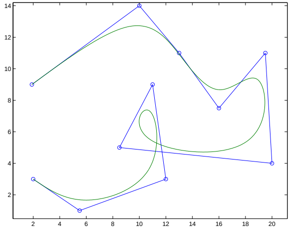
Mean squares Bezier curve, tangent in P_0 and P_n



5 control points. Polynomial of degree 3 . max(|P_i|) = 2.2333

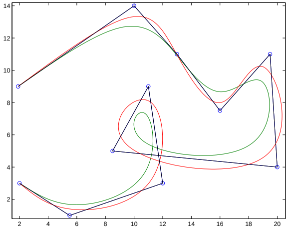
Usual B-spline curve Closer the control polyg Further the control polyg

Usual cubic B−spline curve



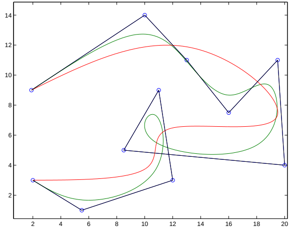
11 control points;   7 knots.   max(|P_i|) = 26.2344

Mean square cubic B−spline curve, interpolating P_0 and P_n



11 control points;   10 knots.   max(|P_i|) = 22.3772

Mean square cubic B−spline curve, interpolating P_0 and P_n



11 control points;   5 knots.   max(|P_i|) = 19.7067

Closer to the control polygon... and still in the same vectorial space

Notation : Control points : $(P_i)_{i=0:n}$

B-curve : $C(t) = \sum_{i=0:n} P_i\, B_i^n(t)$

"hat functions" : b_i^n. Control polygon : $CP(t) = \sum_{i=0:n} P_i\, b_i^n(t)$

First idea : $C(t) = \sum_{i=0:n} Q_i\, B_i^n(t)$. Minimize $\int \|C(t) - CP(t)\|^2\, dt$

$\min_{(Q_i)_{i=0:n}} \int \|\sum_{i=0:n} Q_i\, B_i^n(t) - P_i\, b_i^n(t)\|^2$

Two order $n+1$ linear systems to be solved.

Closer to the control polygon... and still in the same vectorial space

Notation : Control points : $(P_i)_{i=0:n}$

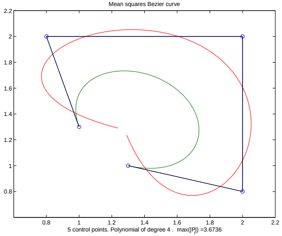
B-curve : $C(t) = \sum_{i=0:n} P_i\, B_i^n(t)$

"hat functions" : b_i^n. Control polygon : $CP(t) = \sum_{i=0:n} P_i\, b_i^n(t)$

First idea : $C(t) = \sum_{i=0:n} Q_i\, B_i^n(t)$. Minimize $\int \|C(t) - CP(t)\|^2\, dt$

$\min_{(Q_i)_{i=0:n}} \int \|\sum_{i=0:n} Q_i\, B_i^n(t) - P_i\, b_i^n(t)\|^2$

Two order $n+1$ linear systems to be solved.



Closer to the control polygon... and still in the same vectorial space

Notation : Control points : $(P_i)_{i=0:n}$

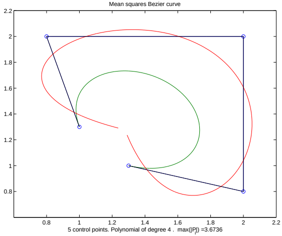
B-curve : $C(t) = \sum_{i=0:n} P_i\, B_i^n(t)$

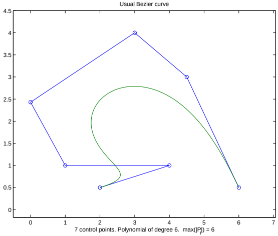
"hat functions" : b_i^n. Control polygon : $CP(t) = \sum_{i=0:n} P_i\, b_i^n(t)$

First idea : $C(t) = \sum_{i=0:n} Q_i\, B_i^n(t)$. Minimize $\int \|C(t) - CP(t)\|^2\, dt$

$\min_{(Q_i)_{i=0:n}} \int \|\sum_{i=0:n} Q_i\, B_i^n(t) - P_i\, b_i^n(t)\|^2$

Two order $n+1$ linear systems to be solved.





Closer to the control polygon...
and still in the same vectorial space

Notation : Control points : $(P_i)_{i=0:n}$

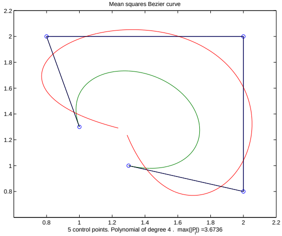
B-curve : $C(t) = \sum_{i=0:n} P_i \, B_i^n(t)$

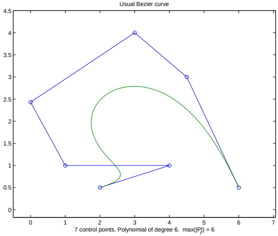
"hat functions" : b_i^n. Control polygon : $CP(t) = \sum_{i=0:n} P_i \, b_i^n(t)$

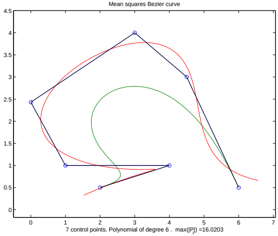
First idea : $C(t) = \sum_{i=0:n} Q_i \, B_i^n(t)$. Minimize $\int \|C(t) - CP(t)\|^2 \, dt$

$\min_{(Q_i)_{i=0:n}} \int \|\sum_{i=0:n} Q_i \, B_i^n(t) - P_i \, b_i^n(t)\|^2$

Two order $n+1$ linear systems to be solved.







Closer to the control polygon...
and still in the same vectorial space

Notation : Control points : $(P_i)_{i=0:n}$

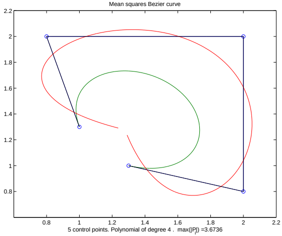
B-curve : $C(t) = \sum_{i=0:n} P_i\, B_i^n(t)$

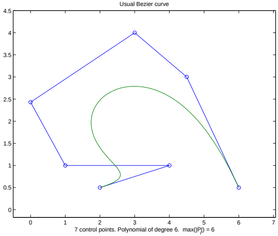
"hat functions" : b_i^n. Control polygon : $CP(t) = \sum_{i=0:n} P_i\, b_i^n(t)$

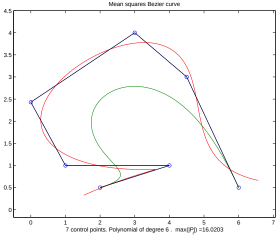
First idea : $C(t) = \sum_{i=0:n} Q_i\, B_i^n(t)$. Minimize $\int \|C(t) - CP(t)\|^2\, dt$

$\min_{(Q_i)_{i=0:n}} \int \|\sum_{i=0:n} Q_i\, B_i^n(t) - P_i\, b_i^n(t)\|^2$

Two order $n+1$ linear systems to be solved.







What about the curve at first and last control point ?

Now with interpolating end conditions :

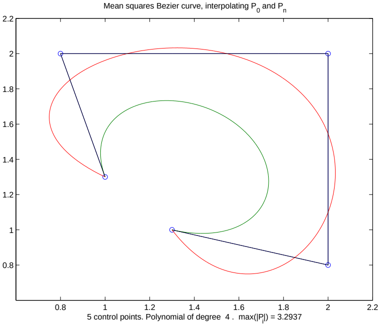
Easy !

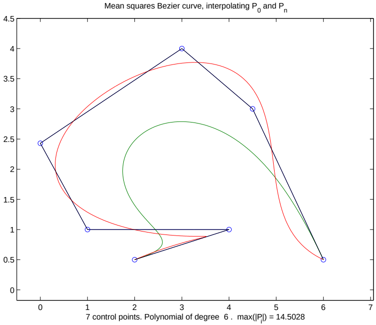
$Q_0 = P_0$ and $Q_n = P_n$,
$\min_{(Q_i)_{i=1:n-1}} \int \|\sum_{i=0:n} Q_i\, B_i^n(t) - P_i\, b_i^n(t)\|^2$

Now with interpolating end conditions :

Easy !

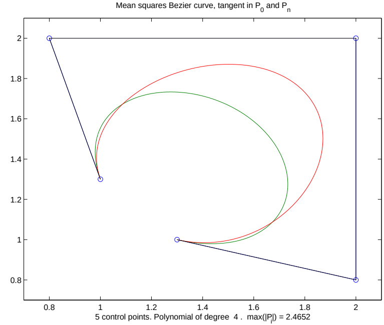
$Q_0 = P_0$ and $Q_n = P_n$,
$\min_{(Q_i)_{i=1:n-1}} \int \|\sum_{i=0:n} Q_i\, B_i^n(t) - P_i\, b_i^n(t)\|^2$

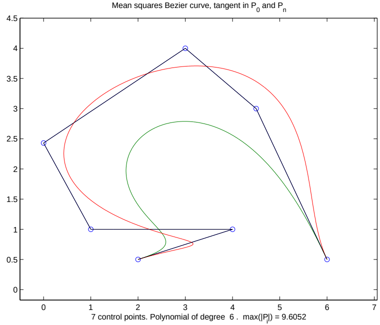




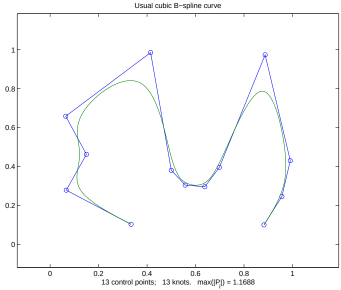
Now with tangent end conditions :

$$Q_0 = P_0,\ Q_1 = P_1,$$
$$Q_{n-1} = P_{n-1} \text{ and } Q_n = P_n,$$
$$\min_{(Q_i)_{i=2:n-2}} \int \left\| \sum_{i=0:n} Q_i\, B_i^n(t) - P_i\, b_i^n(t) \right\|^2$$

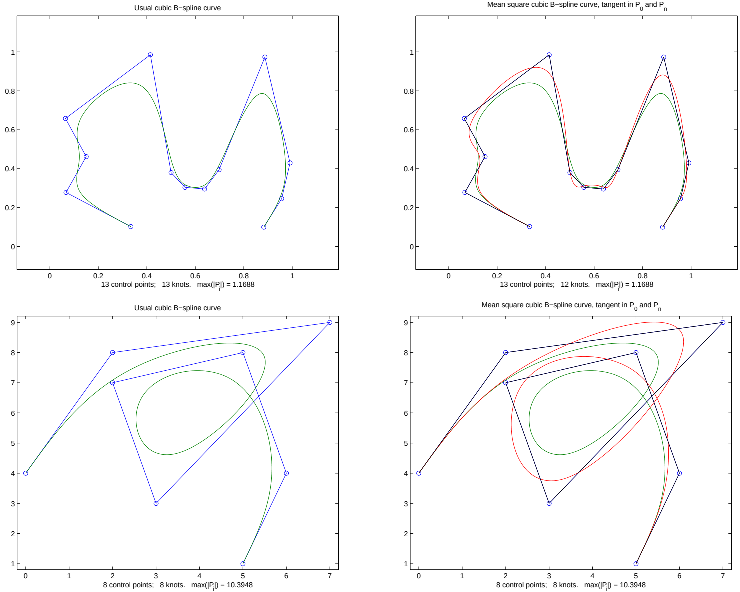




Some curves with cubic B-splines



Some curves with cubic B-splines



Even closer to the control polygon !

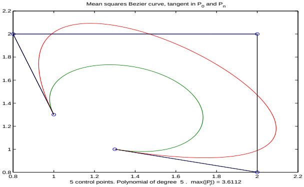
Even closer to the control polygon !

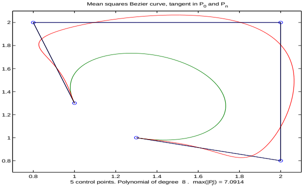
Increase the degree of the polynomial (keeping same data P) :

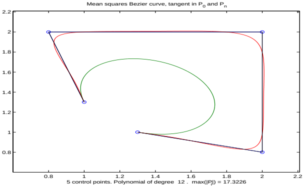
$$N > n.\ Q_0 = P_0 \quad Q_1 = P_0 + \frac{n}{N}(P_1 - P_0),$$

$$Q_n = P_n \text{ and } Q_{n-1} = P_n - \frac{n}{N}(P_n - P_{n-1})$$

$$\min_{(Q_i)_{i=2:N-2}} \int \left\| \sum_{i=0:N} Q_i\, B_i^N(t) - \sum_{j=0:n} P_j\, b_j^n(t) \right\|^2$$







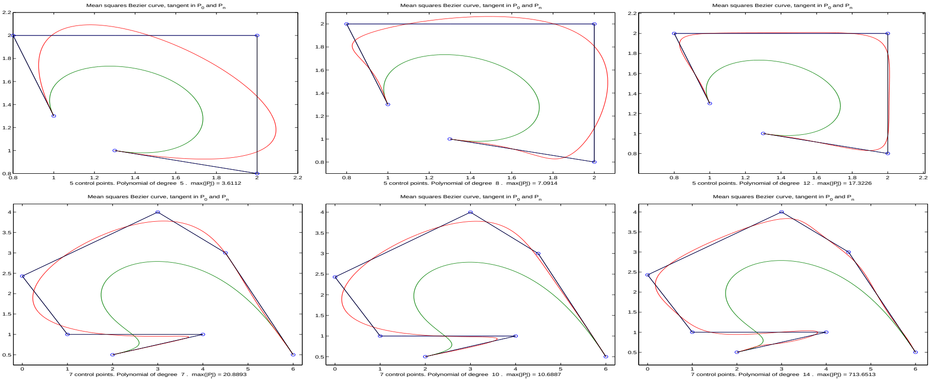
Even closer to the control polygon !

Increase the degree of the polynomial (keeping same data P) :

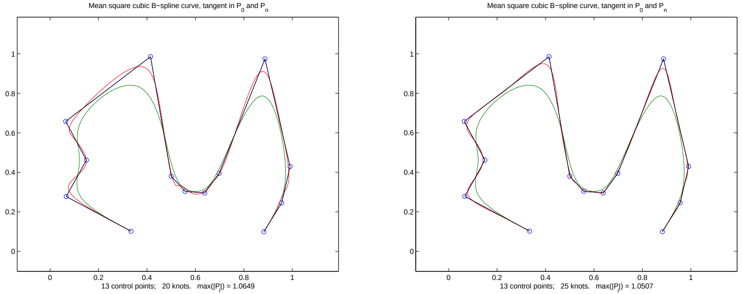
$$N > n.\ Q_0 = P_0 \quad Q_1 = P_0 + \frac{n}{N}(P_1 - P_0),$$

$$Q_n = P_n \text{ and } Q_{n-1} = P_n - \frac{n}{N}(P_n - P_{n-1})$$

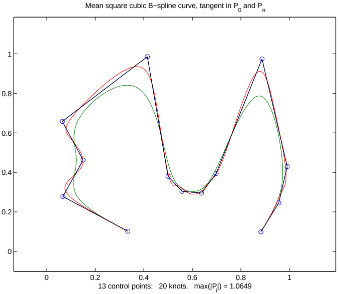
$$\min_{(Q_i)_{i=2:N-2}} \int \left\| \sum_{i=0:N} Q_i\, B_i^N(t) - \sum_{j=0:n} P_j\, b_j^n(t) \right\|^2$$

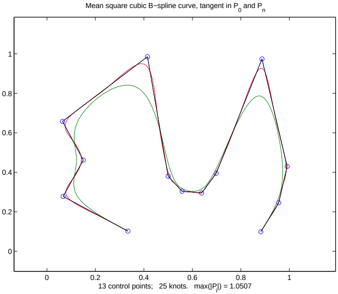


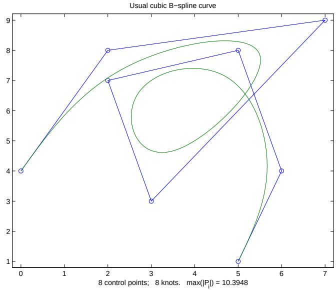
... or the number of knots of the spline :

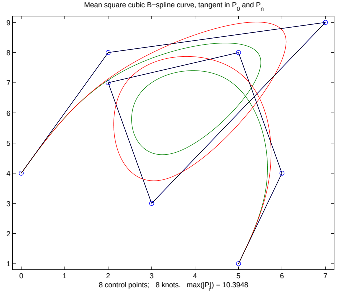


... or the number of knots of the spline :







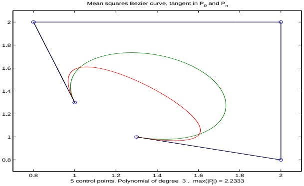


Now a “flatter” curve

Decrease the degree of the polynomial, or the number of knots of the B-spline $\quad N < n, \quad Q_0 = P_0,\ Q_1 = P_0 + \frac{n}{N}(P_1 - P_0),$

$$Q_{n-1} = P_n - \frac{n}{N}(P_n - P_{n-1}) \text{ and } Q_n = P_n,$$

$$\min_{(Q_i)_{i=2:N-2}} \int \left\| \sum_{i=0:N} Q_i\, B_i^N(t) - \sum_{j=0:n} P_j\, b_j^n(t) \right\|^2$$

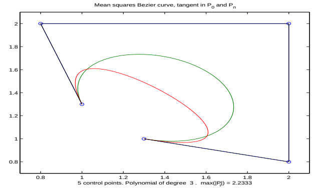


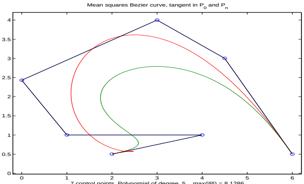
Now a “flatter” curve

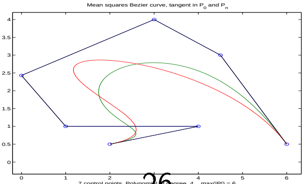
Decrease the degree of the polynomial, or the number of knots of the B-spline $\quad N < n, \quad Q_0 = P_0,\ Q_1 = P_0 + \frac{n}{N}(P_1 - P_0),$

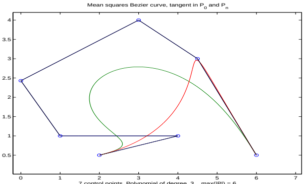
$$Q_{n-1} = P_n - \frac{n}{N}(P_n - P_{n-1}) \text{ and } Q_n = P_n,$$

$$\min_{(Q_i)_{i=2:N-2}} \int \left\| \sum_{i=0:N} Q_i\, B_i^N(t) - \sum_{j=0:n} P_j\, b_j^n(t) \right\|^2$$

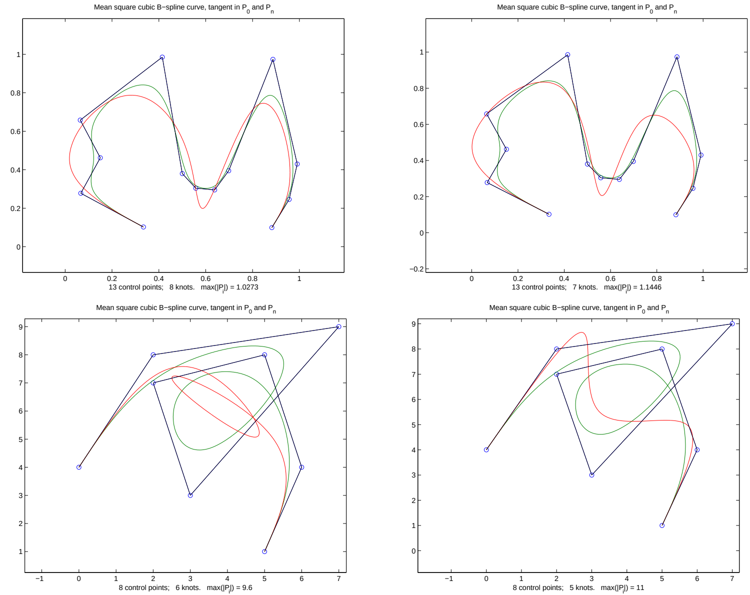








Less knots for cubic B-spline curves



“Least square B-functions”

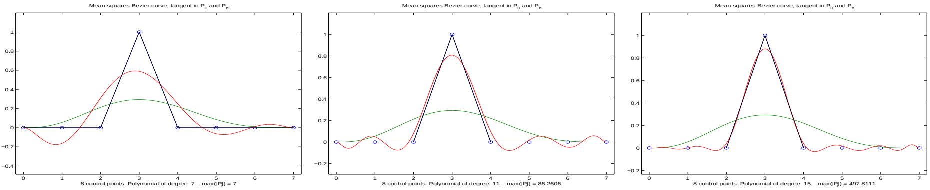
Since the “hat functions” generate the control polygon,
just apply the criterion (closer/further) to the hat functions in order to derive new B-functions :

Let N be $N = n$, or $N > n$, or $N < n$

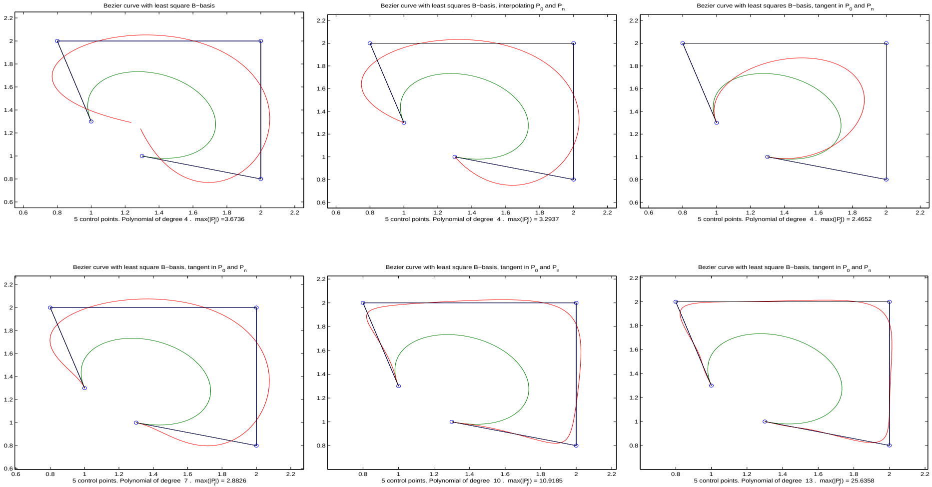
Let C_i^N be the i^{th} “new” B-function.

Define C_i^N as being the function approximating the i^{th} order n hat function, and let $C(t) = \sum_{i=0:N} P_i\, C_i^N$.

We obtain :



We so obtain...

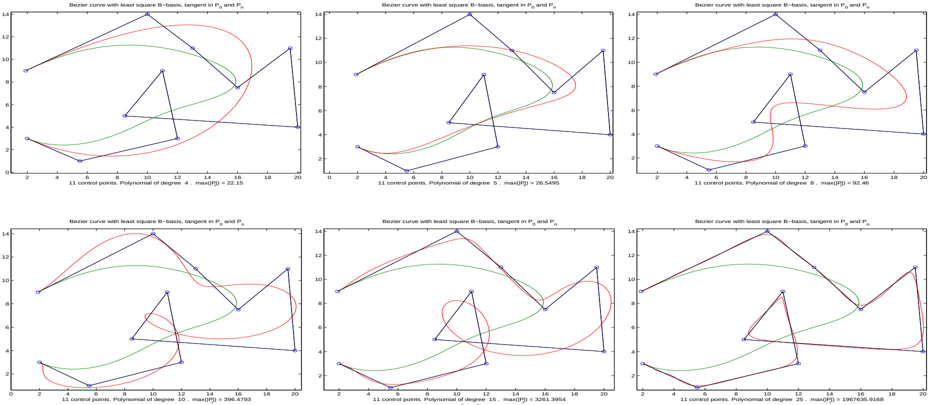


...the same curves !

Convergence towards control polygon

Since polynomials and splines are dense on the space of continuous functions, the B-curve converges towards the control polygon when N tends to infinity.

However be careful with the condition number for polynomials !



Other criteria, other optimizations

In addition to the "classical" B-curve ($Q_i = P_i$), we may be interested in some mix of minimizing

- Distance to the control polygon :
  $E_1 = \int_0^1 (\sum_{i=0:n} Q_i \, B_i^n(t) - \sum_{j=0:k} P_j \, b_j^n(t))^2 \, dt$

- Distance to the control points :
  $E2 = \left(\sum_{j=0:k} Q_i \, B_i^n(j/n) - P_j \right)^2$

- "Smooth curve" : $E3 = \int \left\| \sum_{j=0:k} Q_i \, (B_i^n)''(t) \right\|^2 \, dt$

- Choosing $k \neq n$,k+1 being the dimension of the "B-space" (degree of polynomial, number of knots of the spline function...).

In order to "mix" these criteria, minimize $E = \rho_1 \, E_1 + \rho_2 \, E_2 + \rho_3 \, E_3$...

How to design curves or surfaces

- Choose a vectorial space and associated B-functions

  Polynomials (Bernstein)

  Polynomial splines, NURBS, polyharmonic splines

  Fractional (polynomial or polyharmonic) splines

  ECC spaces (hyperbolic or circular,

  and polynomials, sum of monomials,...)

  ...

- Choose a control polygon (polyhedron for surfaces)
- Choose a level of distance to the control polygon
- Compute the associated B-curve
- Go back to any above item if necessary

What to remember from this talk

DISCONNECT the form of the curves (or surfaces) (ie the functional space in which they are) **from the distance of the curve (surface) to the control points**.

It is easy to **CHOOSE AND MINIMIZE A "DESIRED DISTANCE"** for given B-functions (space) and control polygon.

Doing a **global minimization** is equivalent to
using "**new B-functions**".

Thank you for your attention

Mathematics is beautiful !

Enjoy your mathematics !