

Appunti di
Metodi Numerici per la Grafica
dalle lezioni di Paola Boito

Francesco Baldino

a.a. 22/23

CAGD

CAGD=Computer Aided Geometric Design.

Main goal: represent and manipulate curves and surfaces in three-dimensional space in a numerically efficient way.

Applications: computer graphics, industrial design, computer gaming, scientific visualization, medical imaging, geographic information systems...

Historically born from car and aircraft design.

Interaction with several areas of mathematics (geometry, analysis, numerical analysis...).

Many tools are widely used also outside CAGD (e.g., splines).

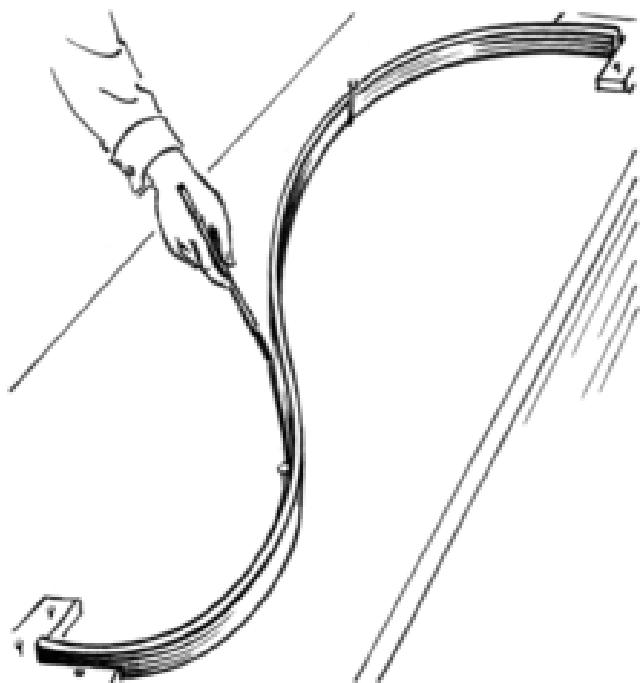
Before computers...



- ▶ use a template/mould,
- ▶ define shape by connecting circular arcs (Venetians, 13th-16th century),
- ▶ apply mechanical splines (earliest mention 1752, probably existed earlier).

Before computers...

Splines: wooden beams bent into “optimal” shapes and used to draw smooth curves.



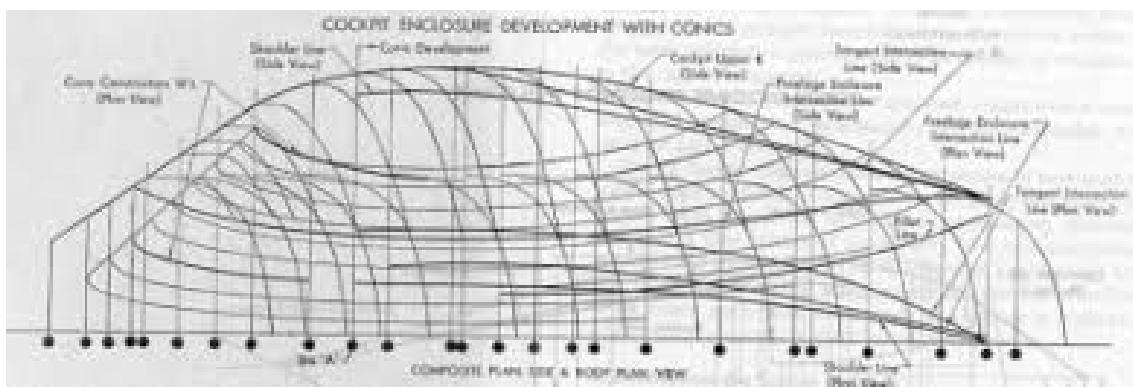
Before computers...

A Boeing engineer drafting with a spline:



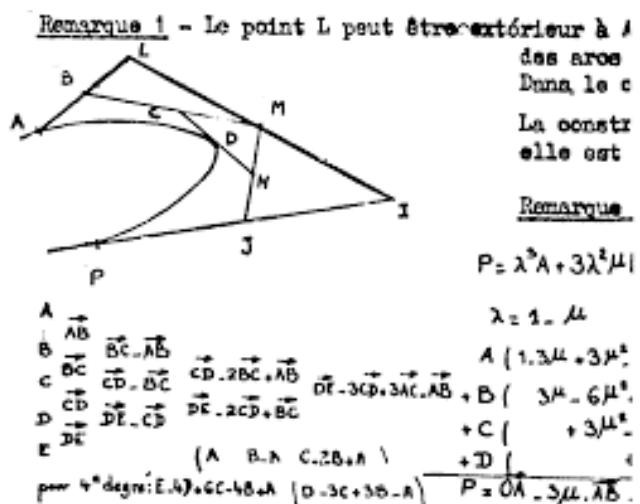
Before computers...

1944: Roy Liming combines classical drafting methods with computation techniques (*Practical Analytical Geometry with Application to Aircraft*, Macmillan).



A major breakthrough

Starting from 1959: **Paul de Casteljau** (working at Citroën) and later **Pierre Bézier** (at Renault) develop independently the theory of Bézier curves and surfaces. Bézier's software UNISURF was one of the first CAD systems.

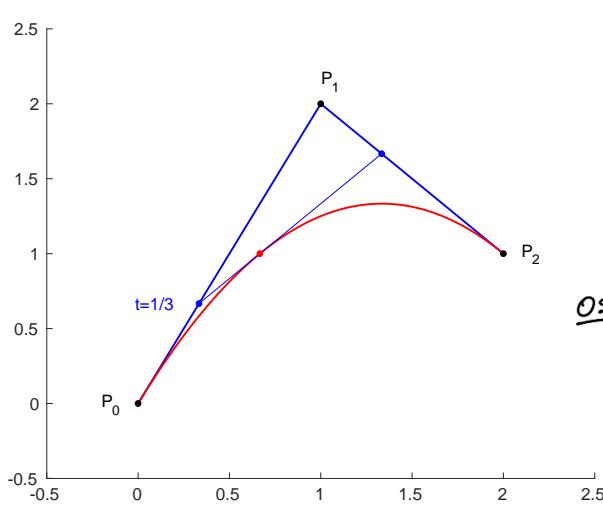


De Casteljau's algorithm

Let P_0, P_1, P_2 be three points in \mathbb{R}^2 . For $t \in [0, 1]$ define

$$\begin{aligned} C_0^1(t) &= (1-t)P_0 + tP_1 && \text{primo passo dell'algoritmo} \\ C_1^1(t) &= (1-t)P_1 + tP_2 && \text{primo punto dell'algoritmo} \\ C_0^2(t) &= (1-t)C_0^1(t) + tC_1^1(t) = (1-t)^2P_0 + 2t(1-t)P_1 + t^2P_2. \end{aligned}$$

We have constructed an arc of parabola.



con \mathbb{R}^3 intendiamo in realtà
il piano affine, utilizzeremo
solo combinazioni convesse
dei punti,

Oss C^2 passa per P_0 e P_2 ma non
 P_1 . Inoltre, è tangente a $\overrightarrow{P_0P_1}$
e $\overrightarrow{P_2P_1}$ ed è contenuta nel
triangolo $P_0P_1P_2$

De Casteljau's algorithm

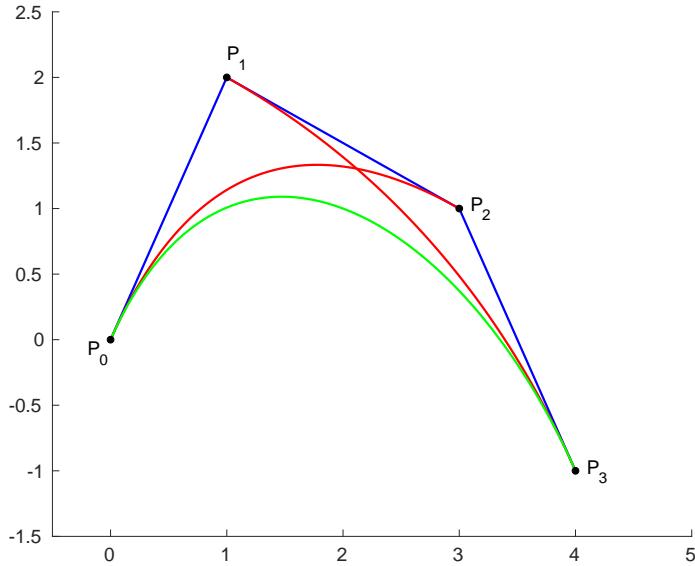
The curve C_0^2 has endpoints P_0 and P_2 . At these points it is tangent to the segments P_0P_1 and P_1P_2 . Also, C_0^2 is contained in the triangle defined by P_0, P_1 and P_2 .

What happens if we shift P_1 in the plane?

De Casteljau's algorithm

What can we do with four control points?

$$\begin{aligned}\mathcal{C}_0^1(t) &= (1-t)P_0 + tP_1, \quad \mathcal{C}_1^1(t) = (1-t)P_1 + tP_2, \quad \mathcal{C}_2^1(t) = (1-t)P_2 + tP_3 \\ \mathcal{C}_0^2(t) &= (1-t)\mathcal{C}_0^1(t) + t\mathcal{C}_1^1(t), \quad \mathcal{C}_1^2(t) = (1-t)\mathcal{C}_1^1(t) + t\mathcal{C}_2^1(t) \\ \mathcal{C}_0^3(t) &= (1-t)\mathcal{C}_0^2(t) + t\mathcal{C}_1^2(t)\end{aligned}$$



De Casteljau's algorithm

More generally, the algorithm works by repeated linear interpolation in \mathbb{R}^3 as follows:

- ▶ Given $P_0, P_1, \dots, P_n \in \mathbb{R}^3$ and $t \in [0, 1]$,
- ▶ set

$$\begin{aligned}\mathcal{C}_i^0(t) &= P_i, \\ \mathcal{C}_i^r(t) &= (1-t)\mathcal{C}_i^{r-1}(t) + t\mathcal{C}_{i+1}^{r-1}(t),\end{aligned}$$

for $r = 1, \dots, n$ and $i = 0, \dots, n-r$.

- ▶ Then $\mathcal{C}_0^n(t)$ is the point of parameter t on the **Bézier curve** defined by the **control points** P_0, P_1, \dots, P_n . The polygon of vertices P_0, P_1, \dots, P_n is called **control polygon**.

De Casteljau's algorithm

For a fixed t , we can write the coefficients $\mathcal{C}_i^r(t)$ in a triangular scheme, e.g.

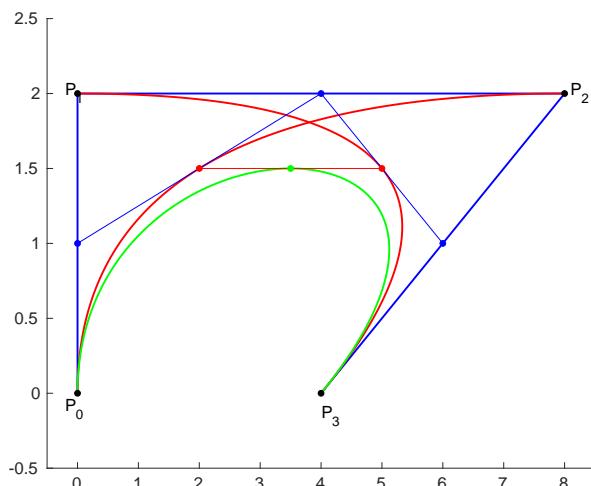
$$\begin{aligned}\mathcal{C}_0^0(t) &= P_0 \\ \mathcal{C}_1^0(t) &= P_1 \quad \mathcal{C}_0^1(t) \\ \mathcal{C}_2^0(t) &= P_2 \quad \mathcal{C}_1^1(t) \quad \mathcal{C}_0^2(t) \\ \mathcal{C}_3^0(t) &= P_3 \quad \mathcal{C}_2^1(t) \quad \mathcal{C}_1^2(t) \quad \mathcal{C}_0^3(t)\end{aligned}$$

- ▶ Computational cost: $n(n + 1)/2$ linear combinations in \mathbb{R}^d , storage $\mathcal{O}(n)$ (rewrite vector).
- ▶ Algorithm is numerically stable (because convex combinations).

De Casteljau's algorithm

Example: a planar cubic, $t = 1/2$:

$$\left[\begin{array}{c} 0 \\ 0 \\ 0 \\ 2 \\ 8 \\ 2 \\ 4 \\ 0 \end{array} \right] \quad \left[\begin{array}{c} 0 \\ 1 \\ 4 \\ 2 \\ 6 \\ 1 \end{array} \right] \quad \left[\begin{array}{c} 2 \\ 3/2 \\ 5 \\ 3/2 \end{array} \right] \quad \left[\begin{array}{c} 7/2 \\ 3/2 \end{array} \right]$$



Properties of Bézier curves

$$\text{invarianza nel senso che commuta} \\ (\lambda(\mathcal{C}(P_0, \dots, P_p)) = \mathcal{C}(\lambda(P_0), \dots, \lambda(P_p))$$

Affine invariance

- ▶ Bézier curves are invariant under affine maps (because linear interpolations are affinely invariant).
- ▶ In practice: suppose we plot a cubic Bézier curve by evaluating it at 100 values of t and plotting the computed points. If we want to apply a rotation to the curve, we can either
 - ▶ apply the rotation to each of the 100 evaluated points, or
 - ▶ apply the rotation to the 4 control points, then evaluate the new curve at 100 points.

The second approach is obviously more efficient (4 rotations versus 100).

Properties of Bézier curves

Invariance under affine parameter transformations

- ▶ So far we have taken $t \in [0, 1]$, but in fact we could work over any real interval $[a, b]$. Just use the (affine) transformation $t = (u - a)/(b - a)$.
- ▶ The generalized de Casteljau algorithm takes the form

$$\mathcal{C}_i^r(u) = \frac{b - u}{b - a} \mathcal{C}_i^{r-1}(u) + \frac{u - a}{b - a} \mathcal{C}_{i+1}^{r-1}(u).$$

Convex hull property

- ▶ For $t \in [0, 1]$, any Bézier curve lies in the convex hull of its control polygon. This is because each $\mathcal{C}_i^r(t)$ is computed as a convex combination of previous $\mathcal{C}_j^{r-1}(t)$.
- ▶ A simple consequence: planar control polygons produce planar Bézier curves.
- ▶ An application: interference checking.

Properties of Bézier curves

Endpoint interpolation

- ▶ The Bézier curve defined by control points P_0, \dots, P_n passes through P_0 and P_n .
- ▶ Easily verified: set $t = 0$ and then $t = 1$ in de Casteljau's algorithm.

Designing with Bézier curves

- ▶ A Bézier curve tends to “mimic” the associated Bézier polygon. For this reason Bézier curves provide a handy tool for interactive design.

Polinomi di Bernstein

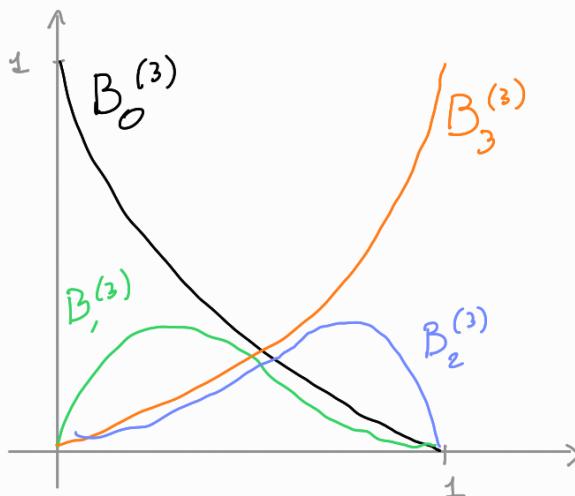
Def Dato $p \in \mathbb{N}$, $p > 0$; i POLINOMI DI BERNSTEIN di

grado p sono $B_i^{(p)}(t) = \binom{p}{i} t^i (1-t)^{p-i}$, $i=0, \dots, p$

Ese Per $p=3$ sono

$$B_0^{(3)}(t) = (1-t)^3 \quad B_2^{(3)}(t) = 3t^2(1-t)$$

$$B_1^{(3)}(t) = 3t(1-t)^2 \quad B_3^{(3)}(t) = t^3$$



Proprietà

- Non negatività: $B_i^{(p)} \geq 0 \quad \forall i=0, \dots, p \quad \forall t \in [0, 1]$
- Partizione dell'unità:

$$\sum_{i=0}^p B_i^{(p)}(t) = \sum_{i=0}^p \binom{p}{i} t^i (1-t)^{p-i} = [t + (1-t)]^p = 1$$

- Ricorrenza: $B_i^{(p)}(t) = (1-t)B_i^{(p-1)}(t) + tB_{i-1}^{(p-1)}(t), \quad i=0, \dots, p$
(dove per convenzione $i < 0 \vee i > p \Rightarrow B_i^{(p)} = 0$)

Dim per induzione su p

$$(1-t)B_i^{(p-1)}(t) + tB_{i-1}^{(p-1)}(t) = \\ = (1-t)\binom{p-1}{i} t^i (1-t)^{p-1-i} + t\binom{p-1}{i-1} t^{i-1} (1-t)^{p-1-i} = \\ \vdots$$

$$\begin{aligned}
&= \left[\binom{P-i}{i} + \binom{P-i}{i-1} \right] t^i (1-t)^{P-i} = \\
&= \binom{P}{i} t^i (1-t)^{P-i} = \\
&= B_i^{(P)}(t)
\end{aligned}$$

□

- $\{B_i^{(P)}(t)\}_{i=0}^P$ è una base dello spazio dei polinomi di grado $\leq P$, che indichiamo con \mathbb{P}_P

Dim Consideriamo la matrice $A \in \mathbb{R}^{(P+1) \times (P+1)}$ tale che

$$\begin{bmatrix} B_0^{(P)}(t) \\ \vdots \\ B_P^{(P)}(t) \end{bmatrix} = A \begin{bmatrix} 1 \\ t \\ \vdots \\ t^P \end{bmatrix}$$

Se riusciamo a mostrare che A è invertibile, poiché $\{t^i\}_{i=0}^P$ è una base di \mathbb{P}_P abbiamo vinto

Sviluppando la definizione abbiamo

$$\begin{aligned}
B_i^{(P)}(t) &= \binom{P}{i} t^i (1-t)^{P-i} = \\
&= \binom{P}{i} t^i \sum_{j=0}^{P-i} (-1)^{P-i-j} t^{P-i-j} \binom{P-i}{j} = \\
&= \binom{P}{i} \sum_{j=0}^{P-i} \binom{P-i}{j} (-1)^{P-j-i} t^{P-j}
\end{aligned}$$

Quindi $B_i^{(P)}(t)$ è combinazione solo di t^i, \dots, t^P , cioè A è triangolare superiore

Inoltre, gli elementi diagonali sono $A_{ii} = \binom{P}{i} \neq 0$ quindi A è invertibile e $\{B_i^{(P)}(t)\}_{i=0}^P$ è base di \mathbb{P}_P

□

- Elevamento di grado

$$\begin{aligned}
 B_i^{(p)}(t) &= [t + (1-t)] B_i^{(p)}(t) = \\
 &= t B_i^{(p)}(t) + (1-t) B_i^{(p)}(t) = \\
 &= \binom{p}{i} \underbrace{t^{i+1} (1-t)^{p-i}}_{\substack{B_{i+1}^{p+1} \text{ senza} \\ \text{binomiale}}} + \underbrace{\binom{p}{i} t^i (1-t)^{p+1-i}}_{\substack{B_i^{p+1} \text{ senza} \\ \text{binomiale}}} = \\
 &= \frac{\binom{p}{i}}{\binom{p+1}{i+1}} B_{i+1}^{(p+1)}(t) + \frac{\binom{p}{i}}{\binom{p+1}{i}} B_i^{(p+1)}(t) = \\
 &= \frac{i+1}{p+1} B_{i+1}^{(p+1)}(t) + \frac{p-i+1}{p+1} B_i^{(p+1)}(t)
 \end{aligned}$$

- Derivata

$$\begin{aligned}
 \frac{d}{dt} B_i^{(p)}(t) &= \binom{p}{i} \left[i t^{i-1} (1-t)^{p-i} + (p-i) t^i (1-t)^{p-i-1} \right] = \\
 &\quad \text{sempre sviluppando il binomiale} \\
 &= p \left[B_{i-1}^{(p-1)}(t) - B_i^{(p-1)}(t) \right]
 \end{aligned}$$

30/09

Abbiamo visto che per i polinomi di Bernstein vale

$$\frac{d}{dt} B_i^{(p)}(t) = p \left[B_{i-1}^{(p-1)}(t) - B_i^{(p-1)}(t) \right]$$

Per le derivate successive vale

$$\frac{d^r}{dt^r} B_i^{(p)}(0) = 0 \quad \text{per } r = 0, \dots, i-1 \quad (\text{fattore } t^i)$$

$$\frac{d^r}{dt^r} B_i^{(p)}(1) = 0 \quad \text{per } r = 0, \dots, p-i-1 \quad (\text{fattore } t^{p-i})$$

L'equazione $\frac{d}{dt} B_i^{(p)}(t) = 0 \Big|_{t \in (0,1)}$ con $1 \leq i \leq p-1$ ammette

come unica soluzione $t = \frac{i}{p}$. Si dimostra che $t = \frac{i}{p}$ è
max per $B_i^{(p)}(t)$, $i=0, \dots, p$.

Il valore $t = \frac{i}{p}$ viene chiamato **ASCISSA DI GREVILLE**

$$\text{Oss} \quad \int_0^1 B_i^{(p)}(t) dt = \frac{1}{p+1} \quad \forall i=0, \dots, p$$

Dim Cominciamo mostrando che tutti i polinomi hanno lo stesso integrale

$$\text{per } 1 \leq i \leq p \text{ vale } 0 = B_i^{(p+1)}(1) - B_i^{(p+1)}(0) = \\ = \int_0^1 \frac{d}{dt} B_i^{(p+1)}(t) dt = \\ = p \int_0^1 B_{i-1}^{(p)}(t) dt - p \int_0^1 B_i^{(p)}(t) dt$$

$$\text{Quindi: } \int_0^1 B_i^{(p)}(t) dt = \int_0^1 B_{i-1}^{(p)}(t) dt \quad \forall i=1, \dots, p \quad \text{cioè}$$

$$\forall i=0, \dots, p, \quad \int_0^1 B_i^{(p)}(t) dt = \int_0^1 B_p^{(p)}(t) dt = \int_0^1 t^p dt = \frac{1}{p+1}$$

Abbiamo visto che $\begin{bmatrix} B_0^{(p)}(t) \\ \vdots \\ B_p^{(p)}(t) \end{bmatrix} = A \begin{bmatrix} 1 \\ t \\ \vdots \\ t^p \end{bmatrix}$ con A triangolare
superiore invertibile \square

Consideriamo $M = A^{-1}$, che è non negativa e descritta
da $t^j = \sum_{k=j}^p \frac{\binom{k}{j}}{\binom{p}{j}} B_k^{(p)}(t)$

Idea d: Dim parti da $t^j = t^j \cdot [t + (1-t)]^{P-j}$ e sviluppa

□

$$\text{In particolare } t = \sum_{k=1}^P \frac{k}{P} B_k^{(P)}(t) = \sum_{k=0}^P \frac{k}{P} B_k^{(P)}(t)$$

ascisse d:
Greville !

Curve d: Bezier

Def Siano $P_0, \dots, P_p \in \mathbb{R}^d$ (spazio affine)

La CURVA DI BEZIER associata è

$$C(t) = \sum_{i=0}^P P_i B_i^{(P)}(t) \quad \text{con } t \in [0, 1]$$

Proprietà

- Invarianza affine: sia $A: \mathbb{R}^d \rightarrow \mathbb{R}^d$ affine

$$\begin{aligned} \text{vale } A(C(t)) &= A \sum_{i=0}^P P_i B_i^{(P)}(t) + q = \\ &= A \sum_{i=0}^P P_i B_i^{(P)}(t) + q \underbrace{\sum_{i=0}^P P_i B_i^{(P)}(t)}_{=1} = \\ &= \sum_{i=0}^P (AP_i + q) B_i^{(P)}(t) = \\ &= \sum_{i=0}^P A(P_i) B_i^{(P)}(t) \end{aligned}$$

- Inviluppo convesso: $C(t)$ è contenuto nell'inviluppo convesso di P_0, \dots, P_p grazie a partizione dell'unità e non negatività de: $B_i^{(P)}(t)$

- Precisione lineare: Se i P_i sono allineati la curva di Bezier corrispondente è un segmento
- Algoritmo di Casteljau:

$$C(t) = \sum_{i=0}^P P_i B_i^{(P)}(t) = \sum_{i=0}^P P_i \left((1-t) B_i^{(P-1)}(t) + t B_{i-1}^{(P-1)}(t) \right) =$$

$$= \sum_{i=0}^P (1-t) P_i B_i^{(P-1)}(t) + \sum_{i=0}^P t P_i B_{i-1}^{(P-1)}(t) =$$

per la prima sommatoria il termine $i=P$ è nullo poiché $B_P^{(P-1)}(t) \equiv 0$, e per la seconda il termine $i=0$ è nullo poiché $B_{-1}^{(P-1)}(t) \equiv 0$. Inoltre shiftiamo l'indice della seconda sommatoria

$$= \sum_{i=0}^{P-1} (1-t) P_i B_i^{(P-1)}(t) + \sum_{i=0}^{P-1} t P_{i+1} B_i^{(P)}(t) =$$

$$= \sum_{i=0}^{P-1} \underbrace{\left[(1-t) P_i + t P_{i+1} \right]}_{P_i^{(1)}} B_i^{(P-1)}(t) =$$

$$= \sum_{i=0}^{P-1} P_i^{(1)} B_i^{(P-1)}(t)$$

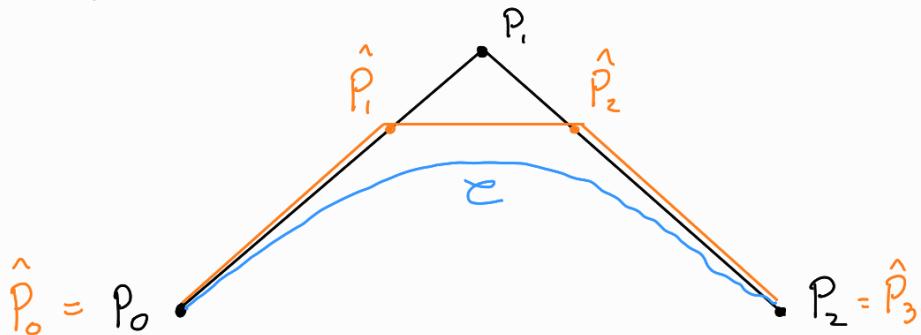
- Interpolazione agli estremi: $C(0) = \sum_{i=0}^P P_i B_i^{(P)}(0) = P_0$

$$C(1) = \sum_{i=0}^P P_i B_i^{(P)}(1) = P_P$$

- Elevamento di grado:

Vogliamo riscrivere $C(t)$ parametrizzandola con $P+2$ punti invece di $P+1$

$$\begin{aligned}
C(t) &= \sum_{i=0}^p P_i B_i^{(p)}(t) = \sum_{i=0}^p P_i \left(\frac{i+1}{p+1} B_{i+1}^{(p+1)}(t) + \frac{p+1-i}{p+1} B_i^{(p+1)}(t) \right) = \\
&= \sum_{i=0}^p P_i \frac{i+1}{p+1} B_{i+1}^{(p+1)}(t) + \sum_{i=0}^p P_i \frac{p+1-i}{p+1} B_i^{(p+1)}(t) = \\
&= \sum_{i=0}^{p+1} P_{i-1} \frac{i}{p+1} B_i^{(p+1)}(t) + \sum_{i=0}^{p+1} P_i \frac{p+1-i}{p+1} B_i^{(p+1)}(t) = \\
&= \sum_{i=0}^{p+1} \left(\frac{i}{p+1} P_{i-1} + \frac{p+1-i}{p+1} P_i \right) B_i^{(p+1)}(t)
\end{aligned}$$



dove $\hat{P}_1 = \frac{1}{3} P_0 + \frac{2}{3} P_1$ e $\hat{P}_2 = \frac{2}{3} P_1 + \frac{1}{3} P_2$

Abbiamo eseguito un processo di "corner cutting"

- Derivata

$$\begin{aligned}
\frac{d}{dt} C(t) &= \sum_{i=0}^p P_i \frac{d}{dt} B_i^{(p)}(t) = p \sum_{i=0}^p P_i \left(B_{i-1}^{(p-1)}(t) - B_i^{(p-1)}(t) \right) = \\
&= p \sum_{i=0}^p P_i B_{i-1}^{(p-1)}(t) - p \sum_{i=0}^p P_i B_i^{(p-1)}(t) = \\
&= p \sum_{i=0}^{p-1} P_{i+1} B_i^{(p-1)}(t) - p \sum_{i=0}^{p-1} P_i B_i^{(p-1)}(t) = \\
&= p \sum_{i=0}^{p-1} (P_{i+1} - P_i) B_i^{(p-1)}(t)
\end{aligned}$$

- Tangenza agli estremi:

$$\frac{d}{dt} C(0) = p(P_1 - P_0) \quad \frac{d}{dt} C(1) = p(P_p - P_{p-1})$$

- Controllo pseudolocale

Vediamo cosa succede perturbando il punto j-esimo

Definiamo:

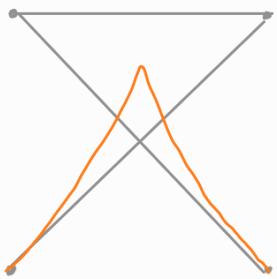
$$\tilde{P}_i = \begin{cases} P_i & \text{se } i \neq j \\ P_j + \Delta p & \text{se } i = j \end{cases}$$

Vale $\tilde{\mathcal{C}}(t) - \mathcal{C}(t) = \Delta p B_j^{(p)}(t) \neq 0 \quad \forall t \in (0, 1)$

Quindi ogni punto della curva è perturbato (quindi non è una perturbazione locale) ma la perturbazione è limitata e sempre in direzione Δp

- Variation Diminishing: per ogni retta in posizione generica il numero di intersezioni tra la retta e la curva è sempre minore o uguale al numero di punti di intersezione tra la retta e il poligono di controllo

Ese [decasteljjeu-bezier.m su elearning]

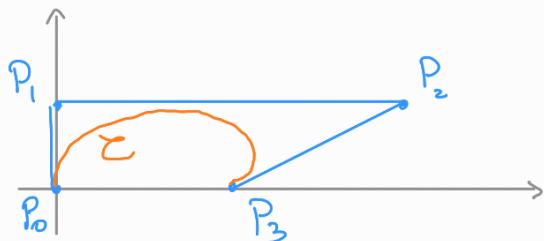


è un esempio di poligono di controllo con auto intersezione, che genera una curva

dotata di cuspidi

Ese (delle slide)

$$P_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad P_1 = \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \quad P_2 = \begin{bmatrix} 8 \\ 2 \end{bmatrix}, \quad P_3 = \begin{bmatrix} 4 \\ 0 \end{bmatrix}$$



Teoricamente sappiamo che se $C(t) = \sum_{i=0}^p P_i B_i^{(p)}(t)$
allora $\frac{d}{dt} C(t) = p \sum_{i=0}^{p-1} (P_{i+1} - P_i) B_i^{(p-1)}(t)$

Possiamo chiedere quanto fa $\frac{d}{dt} C(t)|_{t=1/2} = ?$

$$P_1 - P_0 = \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \quad P_2 - P_1 = \begin{bmatrix} 8 \\ 0 \end{bmatrix}, \quad P_3 - P_2 = \begin{bmatrix} -4 \\ -2 \end{bmatrix}$$

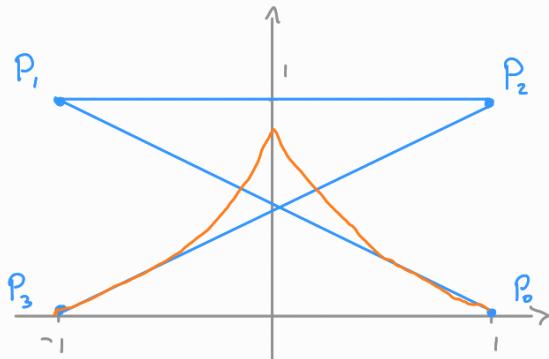
$t = 1/2$ | $t = 1/2$
 $\begin{bmatrix} 4 \\ 1 \end{bmatrix}$ | $\begin{bmatrix} 2 \\ -1 \end{bmatrix}$
 $\underbrace{\qquad\qquad\qquad}_{t = 1/2} \qquad\qquad \underbrace{\qquad\qquad\qquad}_{t = 1/2}$
 $\begin{bmatrix} 3 \\ 0 \end{bmatrix}$

Quindi: $\frac{d}{dt} \mathcal{E}\left(\frac{1}{2}\right) = P \begin{bmatrix} 3 \\ 0 \end{bmatrix} = \begin{bmatrix} g \\ 0 \end{bmatrix}$ dove ci siamo calcolati il valore della sommatoria con l'algoritmo di de casteljau

Esempio prendiamo l'esempio della cuspide e verifichiamo che sia effettivamente una cuspide

$$\begin{aligned} P_0 &= \begin{bmatrix} 1 \\ 0 \end{bmatrix} & P_1 - P_0 &= \begin{bmatrix} -2 \\ 1 \end{bmatrix} \\ P_1 &= \begin{bmatrix} -1 \\ 1 \end{bmatrix} & P_2 - P_1 &= \begin{bmatrix} 2 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 0 \\ 1/2 \end{bmatrix} \\ P_2 &= \begin{bmatrix} 1 \\ 1 \end{bmatrix} & P_3 - P_2 &= \begin{bmatrix} 0 \\ -1 \end{bmatrix} \quad \begin{bmatrix} 0 \\ -1/2 \end{bmatrix} \\ P_3 &= \begin{bmatrix} -1 \\ 0 \end{bmatrix} & & \end{aligned}$$

differenze media per $t = 1/2$



Quindi $t = 1/2$ è effettivamente un punto critico (non basta a mostrare che sia una cuspide ma abbastanza ai fini dell'esercizio)

Oss noi abbiamo prima derivato e poi applicato de casteljau, ma potevamo anche fare la derivata alla fine o in un passo qualsiasi: d: de casteljau invece che all'inizio, per esempio:

$$\begin{aligned} P_0 &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} & P_0^{(1)} &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} & P_0^{(2)} &= \begin{bmatrix} 2 \\ 3/2 \end{bmatrix} \\ P_1 &= \begin{bmatrix} 0 \\ 2 \end{bmatrix} & P_1^{(1)} &= \begin{bmatrix} 4 \\ 2 \end{bmatrix} & P_1^{(2)} &= \begin{bmatrix} 5 \\ 3/2 \end{bmatrix} \\ P_2 &= \begin{bmatrix} 8 \\ 2 \end{bmatrix} & P_2^{(1)} &= \begin{bmatrix} 6 \\ 1 \end{bmatrix} & P_2^{(2)} &= \begin{bmatrix} 3 \\ 0 \end{bmatrix} \quad \text{come sopra} \\ P_3 &= \begin{bmatrix} 4 \\ 0 \end{bmatrix} & & & & \end{aligned}$$

questo perché le combinazioni affini e le derivate commutano

In questo esempio il poligono di controllo si intersecava e così anche la curva (nella cuspide)

Modificando i punti $\tilde{P}_1 = \begin{bmatrix} -0.5 \\ 2 \end{bmatrix}$, $\tilde{P}_2 = \begin{bmatrix} 0.5 \\ 2 \end{bmatrix}$ il poligono si interseca ma la curva no

Prendendo invece un poligono "spiralante" il poligono non si interseca ma la curva sì

Abbiamo visto una cubica planare con cuspide.

Prob è possibile avere una cubica non planare con cuspide?

Risposta NO.

Giustifichiamo questo no. Per avere una cuspide si deve almeno annullare la derivata

$$\text{data } C(t) = \sum_{i=0}^3 P_i B_i^{(3)}(t), \quad \frac{d}{dt} C(t) = 3 \sum_{i=0}^2 (P_{i+1} - P_i) B_i^{(2)}(t)$$

Supponiamo che ci sia una cuspide in $t=\alpha$, allora $\frac{d}{dt} C(\alpha) = 0$, cioè

$$3 \sum_{i=0}^2 (P_{i+1} - P_i) B_i^{(2)}(\alpha) = 0$$

quindi i vettori $P_1 - P_0$, $P_2 - P_1$ e $P_3 - P_2$ sono linearmente dipendenti quindi i punti P_0, \dots, P_3 sono copiani e quindi lo è anche la curva

Oss (grafico di un polinomio)

Il grafico di un polinomio può essere espresso come curva di Bezier.

Dim I polinomi di Bernstein formano una base, quindi

dato un polinomio $q(t)$ d: grado $\deg(q) = n$, posso

scrivere $q(t) = \sum_{i=0}^n q_i B_i^{(n)}(t)$ dove i q_i sono i coefficienti

d: q rispetto alla base d: Bernstein

Il grafico d: q ha equazioni parametriche $\begin{cases} x(t) = t \\ y(t) = q(t) \end{cases}$

Vogliamo quindi scrivere $\begin{bmatrix} t \\ q(t) \end{bmatrix} = \sum_{i=0}^n P_i B_i^{(n)}(t)$ dove $P_i = \begin{bmatrix} P_{i,x} \\ P_{i,y} \end{bmatrix}$

quindi si scomponete in due equazioni

$$\begin{cases} t = \sum_{i=0}^n P_{i,x} B_i^{(n)}(t) \rightarrow \frac{1}{n} \text{ (ascisse d: Greville già viste)} \\ q(t) = \sum_{i=0}^n P_{i,y} B_i^{(n)}(t) \rightarrow P_{i,y} = q_i \end{cases}$$

Quindi il grafico di $q(t)$ si può esprimere come la curva

d: Bezier

$$C(t) = \sum_{i=0}^n \begin{bmatrix} \frac{1}{n} \\ q_i \end{bmatrix} B_i^{(n)}(t)$$

□

Condizionamento della valutazione d: un polinomio

Fissato un grado n , fissata una base $\Phi = (\varphi_0(t), \dots, \varphi_n(t))$

d: \mathbb{P}_n (spazio dei polinomi di grado al più n)

$$\text{Sia } p(t) = \sum_{i=0}^n c_i \varphi_i(t)$$

C: chiediamo cosa si può dire del condizionamento della valutazione

d: $p(t)$ rispetto a questa base (per un t fissato in un certo intervallo $[a, b]$)

Introduciamo una perturbazione sui coefficienti:

$$\text{Sia } \tilde{p}(t) = \sum_{i=0}^n \tilde{c}_i \varphi_i(t) \text{ polinomio perturbato con } \tilde{c}_i = (1 + \varepsilon_i) c_i, |\varepsilon_i| < \varepsilon \forall i$$

$$\text{Allora } |\tilde{p}(t) - p(t)| = \left| \sum_{i=0}^n (\tilde{c}_i - c_i) \varphi_i(t) \right| \leq \sum_{i=0}^n |\tilde{c}_i - c_i| \cdot |\varphi_i(t)| =$$

!

$$= \sum_{i=0}^n |\varepsilon_i c_i| \cdot \varphi_i(t) \leq \varepsilon \underbrace{\sum_{i=0}^n |c_i| |\varphi_i(t)|}_{K(\phi, t)}$$

dove $K_\phi(p, t)$ è detto il **NUMERO DI CONDIZIONAMENTO** del problema

Oss Se è una base non negativa su $[a, b]$ posso scrivere

$$K_\phi(p, t) = \sum_{i=0}^n |c_i| \varphi_i(t)$$

Prop Sono ϕ, ψ basi d: P_n non negative su $[a, b]$ con

$$\phi = \begin{bmatrix} \varphi_0(t) \\ \vdots \\ \varphi_n(t) \end{bmatrix}, \quad \psi = \begin{bmatrix} \psi_0(t) \\ \vdots \\ \psi_n(t) \end{bmatrix}$$

E supponiamo $\psi = F\phi$ dove $F \in \mathbb{R}^{(n+1) \times (n+1)}$ è invertibile e non negativa. Allora

$$K_\phi(p, t) \leq K_\psi(p, t) \quad \forall p \in P_n \quad \forall t \in [a, b]$$

Dim Fisso $p \in P_n$, $t \in [a, b]$ e scrivo

$$\begin{aligned} p(t) &= \sum_{i=0}^n c_i \varphi_i(t) = \sum_{i=0}^n a_i \psi_i(t) \\ &= c^T \phi(t) = a^T \psi(t) \quad \text{con } c = \begin{bmatrix} c_0 \\ \vdots \\ c_n \end{bmatrix}, \quad a = \begin{bmatrix} a_0 \\ \vdots \\ a_n \end{bmatrix} \end{aligned}$$

Allora $c^T \phi(t) = a^T F \phi(t)$ da cui $c = F^T a$, quindi: elemento per elemento si ha $c_i = \sum_{j=0}^n F_{ji} a_j$ e passando ai valori assoluti si ha $|c_i| \leq \sum_{j=0}^n |F_{ji}| |a_j|$ quindi: componente per componente $|c| \leq F^T |a|$

$$\text{Allora } K_\phi(p, t) = |c|^T \phi(t) \leq |a|^T F \phi(t) = |a|^T \psi(t) = K_\psi(p, t)$$

□

Abbiamo visto che possiamo ottenere la base monomiale dalla base d: Bernstein con una matrice non negativa

Oss Sia $\phi(t)$ base d: Bernstein, $t \in [0,1]$
 $\psi(t)$ base monomiale

Sappiamo che $\exists M$ matrice non negativa t.c. $\psi(t) = M\phi(t)$

Allora $K_\phi(p,t) \leq K_\psi(p,t) \quad \forall p \in P_n \quad \forall t \in [0,1]$

Più in generale: Siano ϕ, ψ basi non negative d: P_n su $[0,1]$
definisca la relazione d'ordine parziale

$\phi \preceq \psi \Leftrightarrow K_\phi(p,t) \leq K_\psi(p,t) \quad \forall p \in P_n \quad \forall t \in [0,1]$

Prop La base d: Bernstein è minima rispetto a \preceq

Inoltre, è la sola base minima strettamente positiva su $(0,1)$

Ref R.T. Farouki, T.N.T. Goodman

"On the optimal stability of the Bernstein basis"

Math. comp. 65 (1996), 1553-1566

7/10

Raccordo d: Curve

Date due curve parametriche $C_1(t) \quad t \in [a,b]$
 $C_2(s) \quad s \in [c,d]$

con C_1 e C_2 sufficientemente regolari

vogliamo raccorderle in una curva $\tilde{C}(u) \quad u \in [a,b] \cup [c,d]$

tale che $\tilde{C}(u) = \begin{cases} C_1(u) & \text{se } u \in [a,b] \\ C_2(u) & \text{se } u \in [c,d] \end{cases}$

Condizione di continuità $C^0: C_1(b) = C_2(c)$

Regolarità C^1 (analitica): $\frac{d}{dt} \mathcal{C}_1(b) = \frac{d}{ds} \mathcal{C}_2(c)$

ma poiché non ci interessa davvero la legge oraria ma solo la traiettoria diamo una richiesta più debole

Regolarità G^1 (geometrica):

Sia $w: [c, d] \rightarrow [e, f]$ strettamente crescente e sia

$\hat{\mathcal{C}}_2(w)$ la curva \mathcal{C}_2 ri-parametrizzata

chiedo allora che $\frac{d}{dt} \mathcal{C}_1(b) = \frac{d}{dw} \hat{\mathcal{C}}_2(e)$

Idem per regolarità di grado più elevato

Consideriamo adesso curve di Bezier

$$\mathcal{C}_1(t) = \sum_{i=0}^{n_1} P_i B_i(t) \quad t \in [0, 1]$$

$$\mathcal{C}_2(s) = \sum_{i=0}^{n_2} Q_i B_i(s) \quad s \in [0, 1]$$

Per proprietà di elevamento di grado posso supporre senza perdita di generalità che $n_1 = n_2 = n$

Vediamo come cambiano le richieste precedenti:

Condizione di continuità C^0 :

$$\mathcal{C}_1(1) = \mathcal{C}_2(0) \Leftrightarrow P_n = Q_0$$

Condizione di regolarità C^1 :

$$\frac{d}{dt} \mathcal{C}_1(1) = \frac{d}{ds} \mathcal{C}_2(0) \Leftrightarrow P_n - P_{n-1} = Q_1 - Q_0$$

Condizione di regolarità G^2 :

$$\frac{d}{dt} \mathcal{C}_1(t) = \alpha \frac{d}{ds} \mathcal{C}_2(s) \Leftrightarrow P_n - P_{n-1} = \alpha (Q_1 - Q_0) \quad \alpha > 0$$

$$\Leftrightarrow P_n - P_{n-1} \parallel Q_1 - Q_0$$

Proprietà di suddivisione

Riprendiamo l'algoritmo di de Casteljau

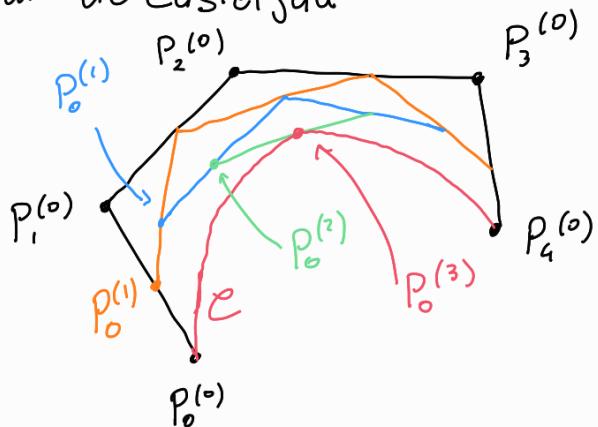
Considero una curva

definita da 5 punti

e applico de Casteljau

per $t = \frac{1}{2}$ per ottenere

$$P_0^{(3)} = \mathcal{C}\left(\frac{1}{2}\right)$$



Oss: $\mathcal{C}[0, \frac{1}{2}]$ sembra una curva di Bézier definita da $P_0^{(0)}, P_0^{(1)}, P_0^{(2)}$ e $P_0^{(3)}$.

Prop. Sia $\mathcal{C}(t)$ curva di Bézier con punti P_0, \dots, P_n .

Fissato $\tau \in (0, 1)$, si ha

$$\mathcal{C}_{P_0, \dots, P_n}(\alpha \tau) = \mathcal{C}_{P_0^{(0)}, \dots, P_0^{(n)}}(\alpha), \quad \alpha \in [0, 1]$$

Lemma 1 Fissati j, n e $\alpha, t \in [0, 1]$ si ha

$$B_j^{(n)}(\alpha t) = \sum_{k=j}^n B_k^{(n)}(\alpha) B_j^{(k)}(t)$$

$$\text{Dim Lemma 1} \quad \sum_{k=j}^n B_k^{(n)}(\alpha) B_j^{(k)}(t) = \sum_{k=j}^n \binom{n}{k} \alpha^k (1-\alpha)^{n-k} \binom{k}{j} t^j (1-t)^{k-j} =$$

$$= \sum_{k=j}^n \underbrace{\binom{n}{k} \binom{k}{j}}_{= \binom{n}{j} \binom{n-j}{k-j}} \alpha^k (1-\alpha)^{n-k} t^j (1-t)^{k-j}$$

$$h = k-j, K = h+j$$

$$= \sum_{h=0}^{n-j} \binom{n}{j} \binom{n-j}{h} \alpha^j \alpha^h (1-\alpha)^{n-j-h} t^j (1-t)^h =$$

$$= \sum_{h=0}^{n-j} \binom{n}{j} \binom{n-j}{h} \alpha^j t^j \alpha^h (1-\alpha)^{n-j-h} (1-t)^h =$$

$$= \binom{n}{j} (\alpha t)^j \sum_{h=0}^{n-j} \binom{n-j}{h} [\alpha(1-t)]^h (1-\alpha)^{n-j-h} =$$

$$= \binom{n}{j} (\alpha t)^j [\alpha(1-t) + 1 - \alpha]^{n-j} =$$

$$= \binom{n}{j} (\alpha t)^j (1 - \alpha t)^{n-j} = B_j^{(n)}(\alpha t)$$

□

Lemma 2 $P_o^{(K)} = \sum_{j=0}^K P_j B_j^{(K)}(\tau)$

Dim Lemma 2 Ex per Δ

Dim $P_{o,p}$

$$C_{P_o^{(0)}, \dots, P_o^{(n)}}(\alpha) = \sum_{K=0}^n B_K^{(n)}(\alpha) P_o^{(K)} = \quad (\text{per i lemma 2})$$

$$= \sum_{K=0}^n \sum_{j=0}^K B_K^{(n)}(\alpha) P_j B_j^{(K)}(\tau) = \quad (\text{cambio d: indic: non banale})$$

$$= \sum_{j=0}^n \left(\sum_{K=j}^n B_K^{(n)}(\alpha) B_j^{(K)}(\tau) \right) P_j = \quad (\text{per i lemma 1})$$

$$= \sum_{j=0}^n P_j B_j^{(n)}(\alpha \tau) = \\ = C_{P_0, \dots, P_n}(\alpha \tau)$$

□

Def Un **MULTIINDICE** di $\{1, \dots, n\}$ è una tupla ordinata $\alpha = (\alpha_1, \dots, \alpha_k)$ tale che $\{\alpha_1, \dots, \alpha_k\} \subseteq \{1, \dots, n\}$ e $\alpha_1 < \alpha_2 < \dots < \alpha_k$ con $k \in \{1, \dots, n\}$

Def Sia $A = (a_{ij})$ matrice $m \times n$

A si dice **TOTALMENTE POSITIVA** (TP) se $\forall \alpha \in \{1, \dots, m\}, \beta \in \{1, \dots, n\}$ multiindice di lunghezza k vale $\det A[\alpha|\beta] > 0$

Analogamente si dice **TOTALMENTE NON NEGATIVA** (TN) se $\forall \alpha, \beta$ come sopra si ha $\det A[\alpha|\beta] \geq 0$

Def Una matrice $A = (a_{ij})$ si dice **STOCASTICA** se $A \geq 0$ e $\forall i: \sum_j a_{ij} = 1$ (ovvero tutte le righe sommano a 1)

Teo (**CAUCHY-BINET**)

Date due matrici A, B di taglia compatibile e α, β due multiindici di lunghezza k allora

$$\det(AB)[\alpha|\beta] = \sum_{\gamma} \det A[\alpha|\gamma] \cdot \det B[\gamma|\beta]$$

dove γ varia tra tutti i multiindici di lunghezza k

Oss Una conseguenza d: Binet è che prodotto d: TP è TP e prodotto d: TN è TN

Un importante esempio d: matrici TP sono le matrici d: Vandermonde a parametri positivi crescenti:

Prop Siano $0 < x_1 < \dots < x_n$ e sia V la matrice d: Vandermonde d: parametri x_1, \dots, x_n

$$V = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^{n-1} \end{bmatrix} \in \mathbb{R}^{n \times n}$$

Allora V è TP

Oss Sia $c \in \mathbb{R}^n$ allora $Vc = \begin{pmatrix} p(x_1) \\ \vdots \\ p(x_n) \end{pmatrix}$ dove $p(x) = \sum_{i=0}^{n-1} c_i x^i$
serve per l'interpolazione

Dim Cominciamo mostrando che ogni sottomatrice d: una matrice d: Vandermonde generica ha determinante non nullo

Sia $A = V(y_1, \dots, y_n) [\alpha | \beta]$ $K \times K$ e consideriamo il sistema lineare $Ac = 0$

Chiamiamo $f(x) = \sum_{i=1}^K c_i x^{\beta_i}$, allora $Ac = \begin{pmatrix} f(y_{\alpha_1}) \\ \vdots \\ f(y_{\alpha_K}) \end{pmatrix}$

Imporre $Ac = 0$ implica che $f(x)$ deve avere almeno K radici distinte, ma poiché ha solo K termini, per la regola di Cartesio ciò è possibile solo se $f \equiv 0$

ovvero solo se $c=0$, quindi A è invertibile
e $\det A \neq 0$

Dimostriamo ora la tesi per induzione su n

$n=1$ Per $n=1$: determinanti sono le entrate della matrice
 V che per ipotesi sono tutti positivi:

$n>1$ Sia $A = V[\alpha | \beta]$ sotto matrice $n \times n$ e sviluppiano
il suo determinante sull'ultima riga, indicando con
 d_j il sotto determinante j -esimo

$$\begin{aligned}\det A &= x_{\alpha_n}^{B_n} d_n - x_{\alpha_n}^{B_{n-1}} d_{n-1} + \dots \\ &= g(x_{\alpha_n})\end{aligned}$$

dove $g(x)$ è un polinomio il cui termine di testa ha
coefficiente positivo per ipotesi induttiva. Segue
che a meno di prendere \bar{x} sufficientemente grande,
esiste \tilde{x} t.c. $g(\tilde{x}) > 0$

Se fosse $\det A = g(x_{\alpha_n}) < 0$ allora deve esistere
 \tilde{x} t.c. $g(\tilde{x}) = 0$ e sarebbe quindi possibile,
sostituendo \tilde{x} a x_{α_n} , costruire una matrice di
Vandermonde la cui sottomatrice di righe α e
colonne β abbia determinante nullo che abbiamo
mostrato essere impossibile

Quindi: $\det A > 0$

Def Una matrice A è **BIDIAGONALE**...

- **SUPERIORE** se $A_{ij} \neq 0 \Rightarrow j=i \vee j=i+1$
- **INFERIORE** se $A_{ij} \neq 0 \Rightarrow j=i \vee j=i-1$



Teo (Cryer 1976)

Ogni matrice TN si fattorizza nel prodotto di matrici bidiagonali: TN aventi al massimo un elemento non nullo esclusa la diagonale principale

Alternativamente: ogni matrice TN è prodotto di matrici bidiagonali, TN

Es (fattorizzazione bidiagonale tramite eliminazione di Neville, matrici di Pascal)

$$P_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 3 & 6 & 10 \\ 1 & 4 & 10 & 20 \end{pmatrix}, \text{ in generale } P_{sj} = P_{j1} = 1, \quad j=1, \dots, n$$

$$P_{i+1,j+1} = P_{i+1,j} + P_{i,j+1}$$

P_n è TP quindi in particolare è TN

Applico Neville:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 3 & 6 & 10 \\ 1 & 4 & 10 & 20 \end{bmatrix} \xrightarrow{R_4 = R_4 - R_3} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 3 & 6 & 10 \\ 0 & 1 & 4 & 10 \end{bmatrix} \xrightarrow{R_3 = R_3 - R_2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 0 & 1 & 3 & 6 \\ 0 & 1 & 4 & 10 \end{bmatrix} \quad \dots$$

moltiplicazione
per $\begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & -1 & 1 \end{bmatrix}$ $\begin{bmatrix} 1 & & & \\ & 1 & & \\ & & -1 & 1 & \\ & & & & 1 \end{bmatrix}$

$$\underline{R_2 = R_2 - R_1} \rightarrow \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \\ 0 & 1 & 3 & 6 \\ 0 & 1 & 4 & 10 \end{bmatrix} \quad \underline{R_4 = R_4 - R_3} \rightarrow \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \\ 0 & 1 & 3 & 6 \\ 0 & 0 & 1 & 4 \end{bmatrix} \quad \underline{R_3 = R_3 - R_2} \rightarrow$$

$$\begin{bmatrix} 1 & & & \\ -1 & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & -1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & & & \\ & 1 & & \\ & & -1 & 1 \\ & & & 1 \end{bmatrix}$$

$$\sim \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 1 & 4 \end{bmatrix} \quad \underline{R_4 = R_4 - R_3} \rightarrow \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & -1 & 1 \end{bmatrix}$$

Ese eseguire lo stesso procedimento dall'altro lato per eliminare le righe

Si arriva ad una forma diagonale, cioè in generale

dette $E_n^{(k)}(\alpha) = \begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & \alpha & \\ & & & \ddots \\ & & & & 1 \end{bmatrix}$

(k-1)-esima colonna

k-esima riga

se λ_k

$$\left[\begin{array}{c|ccccc} 1 & 0 & \dots & 0 \\ \hline 0 & P_{n-1} \\ \vdots & & & & & \\ 0 & & & & & \end{array} \right] = E_n^{(2)}(-1) E_n^{(3)}(-1) \dots E_n^{(n)}(-1) \cdot P_n \cdot E_n^{(n)}(-1)^T - E_n^{(3)}(-1)^T E_n^{(2)}(-1)^T$$

$$= \left(\prod_{k=2}^n E_n^{(k)}(-1) \right) P_n \left(\prod_{k=2}^n E_n^{(k)}(-1) \right)^T$$

Ovvero $P_n = E_n^{(n)}(1) - E_n^{(2)}(1) \left[\begin{array}{c|ccccc} 1 & 0 & \dots & 0 \\ \hline 0 & P_{n-1} \\ \vdots & & & & & \\ 0 & & & & & \end{array} \right] E_n^{(2)}(1)^T - E_n^{(n)}(1)^T$

e iterando abbiamo che P_n è fattorizzata in matrici bidiagonali elementari

Osserviamo che volendo scrivere la fattorizzazione non in matrici bidagonali elementari ma solo bidagonali,

Allora $E_n^{(n)}(1) - E_n^{(2)}(1) = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 & 1 \end{bmatrix}$

Notazione Dato $x \in \mathbb{R}^n$ indichiamo con $V(x)$ il numero d: cambiamenti di segno di x (ignorando gli zeri)

Prop. Sia $B \in \mathbb{R}^{n \times n}$ bidagonale TN

Allora $\forall x \in \mathbb{R}^n \quad V(Bx) \leq V(x)$

Dim Consideriamo il caso B bidagonale superiore (il caso bidagonale inferiore è simmetrico)

$$(Bx)_i = B_{ii}x_i + B_{i,i+1}x_{i+1} \quad (\text{ponendo a 0 gli elementi non definiti})$$

Definisco $w = [x_1, (Bx)_1, x_2, (Bx)_2, \dots, x_n, (Bx)_n]^T$

Poiché $B_{ij} \geq 0 \quad \forall i,j$, allora se x_i e x_{i+1} hanno lo stesso segno anche x_i e $(Bx)_i$ hanno lo stesso segno

quindi: $V(w) \leq V(x)$

Inoltre Bx è una sotto successione d: w quindi:

$$V(Bx) \leq V(w), \quad \text{da cui la tesi}$$

□

Cor Poiché ogni matrice TN si fattorizza in bidagonali TN, data A matrice TN, $\forall x \in \mathbb{R}^n \quad V(Ax) \leq V(x)$

Def Sia $\varphi(t) = (\varphi_0(t), \dots, \varphi_n(t))$ sistema di funzioni con

$$\varphi_i : I \subseteq \mathbb{R} \rightarrow \mathbb{R}$$

Diciamo che il sistema è **TN** se per ogni scelta di $t_0 < t_1 < \dots < t_r$, $t_i \in I$, la matrice di collocazione

$$M_{\varphi_0, \dots, \varphi_n}^{\varphi_0, \dots, \varphi_n}_{t_0, \dots, t_r} \begin{pmatrix} \varphi_0(t_0) & \dots & \varphi_n(t_0) \\ \vdots & & \vdots \\ \varphi_0(t_r) & \dots & \varphi_n(t_r) \end{pmatrix} \in \mathbb{R}^{(r+1) \times (n+1)}$$

è TN

Analogamente per definire un sistema **TP**

$$\text{Oss} \quad \text{Sia } c = \begin{pmatrix} c_0 \\ \vdots \\ c_n \end{pmatrix} \in \mathbb{R}^{n+1} \text{ allora } M_{\varphi_0, \dots, \varphi_n}^{\varphi_0, \dots, \varphi_n}_{t_0, \dots, t_r} \cdot c = \begin{pmatrix} \sum_{j=0}^n c_j \varphi_j(t_0) \\ \vdots \\ \sum_{j=0}^n c_j \varphi_j(t_r) \end{pmatrix}$$

Prop. Sia $(\varphi_0, \dots, \varphi_n)$ sistema di funzioni TN su $I \subseteq \mathbb{R}$

Allora:

(1) Se $h: J \rightarrow I$ è strettamente crescente, allora

$$(\varphi_0 \circ h, \dots, \varphi_n \circ h) \text{ è TN su } J$$

(2) Se $g: I \rightarrow \mathbb{R}$ è una funzione non negativa allora

$$(g \cdot \varphi_0, \dots, g \cdot \varphi_n) \text{ è TN su } I$$

(3) Se $A \in \mathbb{R}^{(n+1) \times (n+1)}$ è TN, allora il sistema

$$(\psi_0, \dots, \psi_n) \text{ definito da } \psi_i(t) = \sum_{j=0}^n A_{ij} \varphi_j(t)$$

è TN

$$\text{moralmente } \begin{pmatrix} \psi_0 \\ \vdots \\ \psi_n \end{pmatrix} = A \begin{pmatrix} \varphi_0 \\ \vdots \\ \varphi_n \end{pmatrix}$$

Dim

(1) Siano $u_0 < \dots < u_r$, $u_i \in J$

$$M_{\varphi_0 \circ h, \dots, \varphi_n \circ h} = \begin{bmatrix} \varphi_0 \circ h(u_0) & \dots & \varphi_n \circ h(u_0) \\ \vdots & & \vdots \\ \varphi_0 \circ h(u_r) & \dots & \varphi_n \circ h(u_r) \end{bmatrix} = M_{\varphi_0, \dots, \varphi_n} \begin{bmatrix} h(u_0), \dots, h(u_r) \end{bmatrix}$$

e poiché h è strettamente positiva $M_{\varphi_0, \dots, \varphi_n} \begin{bmatrix} h(u_0), \dots, h(u_r) \end{bmatrix}$ è TN

(2) Siano $t_0 < \dots < t_r$, $t_i \in I$

$$M_{g \cdot \varphi_0, \dots, g \cdot \varphi_n} = \begin{bmatrix} g(t_0) \varphi_0(t_0) & \dots & g(t_0) \varphi_n(t_0) \\ \vdots & & \vdots \\ g(t_r) \varphi_0(t_r) & \dots & g(t_r) \varphi_n(t_r) \end{bmatrix} =$$

$$= \underbrace{\text{diag}(g(t_0), \dots, g(t_r))}_{TN \text{ perche' } g \text{ non negativa}} M_{\varphi_0, \dots, \varphi_n} \begin{bmatrix} t_0, \dots, t_r \end{bmatrix}$$

Quindi è TN

(3) Siano $t_0 < \dots < t_r \in I$

$$\left(M_{\varphi_0, \dots, \varphi_n} \right)^T = A \cdot \left(M_{\varphi_0, \dots, \varphi_n} \right)^T \Rightarrow M_{\varphi_0, \dots, \varphi_n} = M_{\varphi_0, \dots, \varphi_n} \begin{bmatrix} t_0, \dots, t_r \end{bmatrix} \cdot A^T$$

è TN perché prodotto di TN

□

Teo Sia $(\varphi_0, \dots, \varphi_n)$ TN su I , sia $c \in \mathbb{R}^{n+1}$, $c = \begin{pmatrix} c_0 \\ \vdots \\ c_n \end{pmatrix}$

Sia $f(t) = \sum_{i=0}^n c_i \varphi_i(t)$. Allora il numero di cambiamenti

d: segno di f su I è \leq del numero d: cambiamenti d: segno d: c

$$\underline{\text{Dim}} \quad \forall t_0 < \dots < t_r \in I \quad \text{ho} \quad \begin{pmatrix} f(t_0) \\ \vdots \\ f(t_r) \end{pmatrix} = M_{\varphi_0, \dots, \varphi_n} \cdot c$$

$$\text{quindi per il corollario d: pr.: ma } V \begin{pmatrix} f(t_0) \\ \vdots \\ f(t_r) \end{pmatrix} \leq V \begin{pmatrix} c_0 \\ \vdots \\ c_r \end{pmatrix}$$

Dall'arbitrarietà della scelta dei t_i ho la tesi:

□

14/10

Applichiamo la teoria appena vista alle curve di Bezier

Preso $(\varphi_0, \dots, \varphi_n)$ sistema $\underbrace{\text{NTN}}_{\text{cioè } \sum \varphi_j = 1}$ su $I \subseteq \mathbb{R}$ e scelti $c_0, \dots, c_n \in \mathbb{R}^d$
definisca la curva $C(t) = \sum_{j=0}^n c_j \varphi_j(t)$

Consideriamo il caso $d=2$

$$c_j = \begin{pmatrix} C_{jx} \\ C_{jy} \end{pmatrix} \quad C(t) = \begin{pmatrix} C_x(t) \\ C_y(t) \end{pmatrix}$$

Scegliamo una retta affine $\ell: ax + by + c = 0$ nel piano
se osserviamo $aC_x(t) + bC_y(t) + c$, in prossimità di intersezioni
(trasversali, non tangenti) tra la curva e la retta questa quantità
cambierà di segno

Analogamente, se il segmento $\overline{C_i C_{i+1}}$ interseca internamente la
retta, $aC_{ix} + bC_{iy} + c$ avrà segno opposto a $aC_{(i+1)x} + bC_{(i+1)y} + c$

Quindi, il numero di intersezioni (trasversali) fra $C(t)$ e ℓ è
dato dal numero di cambiamenti di segno di $aC_x(t) + bC_y(t) + c$
e vale

$$\begin{aligned}
 aC_{jx}(t) + bC_{jy}(t) + c &= a \sum_{j=0}^n C_{jx} \varphi_j(t) + b \sum_{j=0}^n C_{jy} \varphi_j(t) + c = \\
 &= a \sum_{j=0}^n C_{jx} \varphi_j(t) + b \sum_{j=0}^n C_{jy} \varphi_j(t) + c \sum_{j=0}^n 1 \cdot \varphi_j(t) = \\
 &= \sum_{j=0}^n (aC_{jx} + bC_{jy} + c) \varphi_j(t)
 \end{aligned}$$

che per il teorema visto precedentemente è minore o uguale al numero d: variazioni d: segno d: $aC_{jx} + bC_{jy} + c$, $j=0, \dots, n$

Teo (VARIATION DIMINISHING)

Sia $(\varphi_0, \dots, \varphi_n)$ sistema NTN su $I \subseteq \mathbb{R}$, $C_0, \dots, C_n \in \mathbb{R}^2$ e $\mathcal{C}(t) = \sum_{j=0}^n C_j \varphi_j(t)$

Allora per ogni retta ℓ nel piano, il numero d: punti di intersezione (trasversale) fra ℓ e \mathcal{C} è minore o uguale al numero d: intersezioni fra ℓ e il poligono di controllo

Es (di basi TN d: P_n)

- base monomiale su $[0, +\infty)$

Abbiamo mostrato che $\forall 0 < t_0 < \dots < t_r$ vale

$$V = \begin{pmatrix} 1 & t_0 & \dots & t_0^{n-1} \\ 1 & t_1 & \dots & t_1^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & t_r & \dots & t_r^{n-1} \end{pmatrix} \text{ è TP}$$

quindi la base monomiale è TN su $(0, +\infty)$ e per continuità anche su $[0, +\infty)$

- base di Bernstein su $[0, 1]$

Sia $\beta(t): [0, 1] \rightarrow [0, +\infty)$, $\beta(t) = \frac{t}{1-t}$ strettamente crescente

$$e \quad g(t) : [0, 1] \rightarrow \mathbb{R}, \quad g(t) = (1-t)^n$$

Usiamo queste due funzioni per trasformare la base monomiale

$$u^i \xrightarrow{\circ f} \left(\frac{t}{1-t} \right)^i \xrightarrow{\circ g} (1-t)^n \frac{t^i}{(1-t)^i} = t^i (1-t)^{n-i}$$

e queste funzioni, per $i=0, \dots, n$, formano un sistema TN su $[0, 1]$

$$\underbrace{\begin{pmatrix} \binom{n}{0} & \binom{n}{1} & \cdots & \binom{n}{n} \end{pmatrix}}_{TN} \begin{pmatrix} (1-t)^n \\ t(1-t)^{n-1} \\ \vdots \\ t^n \end{pmatrix} = \begin{pmatrix} B_0^{(n)}(t) \\ B_1^{(n)}(t) \\ \vdots \\ B_n^{(n)}(t) \end{pmatrix}$$

Quindi: la base d: Bernstein è TN su $[0, 1]$ e per continuità anche su $[0, 1]$

In particolare è NTN

Ottimalità geometrica

Siano $(\varphi_0, \dots, \varphi_n)$ e (v_0, \dots, v_n) basi NTN e sia K la matrice d: cambiamento d: base, $(\varphi_0, \dots, \varphi_n) = (v_0, \dots, v_n)K$

Oss Se φ è una base normalizzata allora φK è base normalizzata se e solo se K è non singolare e stocastica (dove K è la matrice d: cambiamento d: base)

Dim per esercizio

Teo Sia $A \in \mathbb{R}^{n \times n}$ matrice TN stocastica

Allora A si fattorizza come prodotto di matrici TN bidiagonali stocastiche della forma

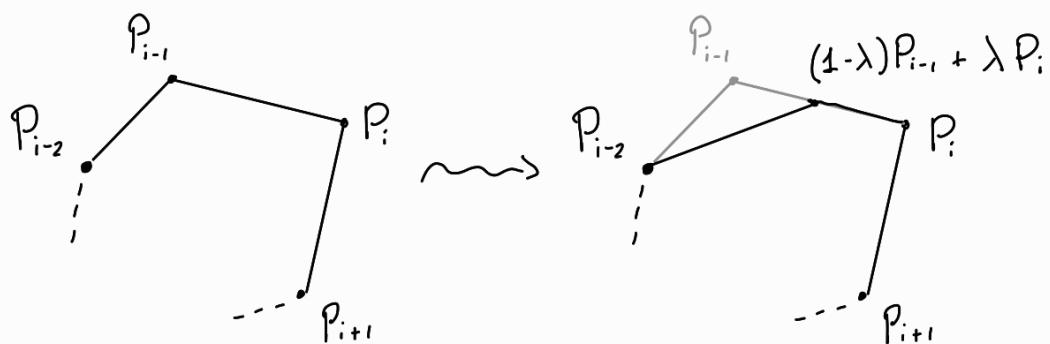
$$\begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & \mu & 1-\mu \\ & & & 1 \end{bmatrix}, \begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & 1-\lambda & \lambda \\ & & & 1 \end{bmatrix} \quad \text{per } \mu, \lambda \in [0, 1]$$

Oss Siano P_0, \dots, P_{n-1} punti di controllo, allora

$$i - \begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & 1-\lambda & \lambda \\ & & & 1 \end{bmatrix} \begin{pmatrix} P_0 \\ \vdots \\ P_{i-1} \\ \vdots \\ P_{n-1} \end{pmatrix} = \begin{pmatrix} P_0 \\ P_i \\ \vdots \\ (1-\lambda)P_{i-1} + \lambda P_i \\ P_i \\ \vdots \\ P_{n-1} \end{pmatrix}$$

e analogamente prendendo l'altro tipo di matrice

Visivamente



Ovvero abbiamo attuato un processo di corner cutting

Tornando alle nostre due basi, sia $C(t) = \sum_{j=0}^n P_j \varphi_j(t)$ una curva scritta in base $(\varphi_0, \dots, \varphi_n)$

Vale $\varphi_i(t) = \sum_{k=0}^n \varphi_k(t) K_{kj}$, quindi possiamo riscrivere $C(t)$ come

$$\begin{aligned}
 C(t) &= \sum_{j=0}^n P_j \varphi_j(t) = \sum_{j=0}^n P_j \sum_{k=0}^n \vartheta_k(t) K_{kj} = \\
 &= \sum_{k=0}^n \vartheta_k(t) \underbrace{\sum_{j=0}^n K_{kj} P_j}_{Q_k} = \sum_{k=0}^n \vartheta_k(t) Q_k
 \end{aligned}$$

dove $\begin{pmatrix} Q_0 \\ \vdots \\ Q_n \end{pmatrix} = K \begin{pmatrix} P_0 \\ \vdots \\ P_n \end{pmatrix}$

quindi per l'osservazione precedente e per il teorema, i punti del poligono di controllo associato alla base ϑ si ottengono dal poligono di controllo associato alla base φ tramite successive operazioni di corner cutting.

Da un punto di vista geometrico, poiché le operazioni di corner cutting "avvicinano" il poligono di controllo alla curva, consideriamo ϑ una base "migliore" della base φ .

Teo Su P_n la base di Bernstein $(B_0^{(n)}, \dots, B_n^{(n)})$ ha la seguente proprietà:

\forall (altra) base NTN $(\varphi_0, \dots, \varphi_n)$ $\exists K \in \mathbb{R}^{(n+1) \times (n+1)}$ TN stocastica tale che $(\varphi_0, \dots, \varphi_n) = (B_0^{(n)}, \dots, B_n^{(n)}) K$

Free-form curves and blending functions

- We have seen that **Bézier curves** in \mathbb{R}^d can be written in the form

$$\mathcal{C}(t) = \sum_{i=0}^n B_i^{(n)}(t) P_i, \quad t \in [0, 1],$$

where $P_0, \dots, P_n \in \mathbb{R}^d$ are the **control points** and $B_0^{(n)}(t), \dots, B_n^{(n)}(t)$ are the **Bernstein polynomials** of degree n .

- What happens if we replace the Bernstein polynomials with other functions? Given n , we could choose some other set of functions $f_0(t), \dots, f_n(t)$ and define the curve

$$\mathcal{K}(t) = \sum_{i=0}^n f_i(t) P_i, \quad t \in [0, 1].$$

- In CAGD jargon, $f_0(t), \dots, f_n(t)$ are called **blending functions**.

Free-form curves and blending functions

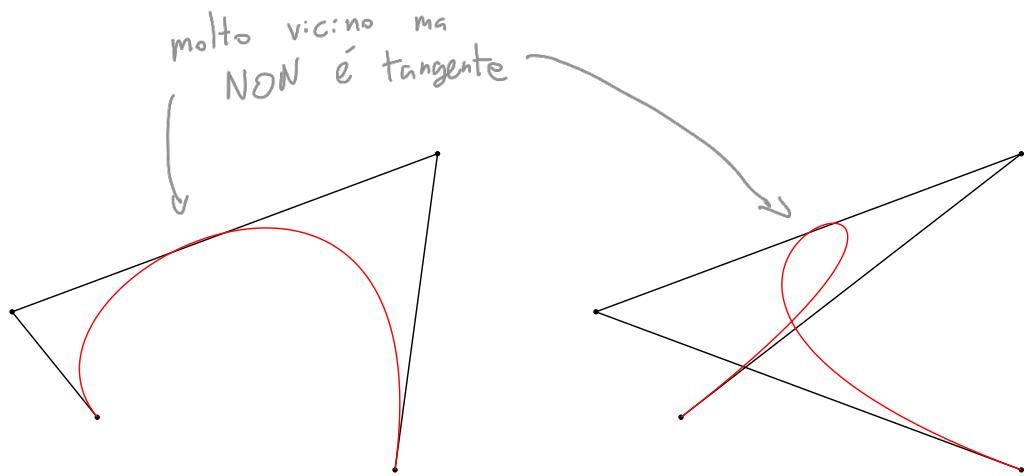
- Although there are good reasons to choose the Bernstein polynomials as blending functions, other functions have sometimes been used. Let's take a look at some examples for $n = 3$.
- These examples share two important properties:
 - **Affine invariance**: $f_0(t), f_1(t), f_2(t), f_3(t)$ form a partition of unity. This means that the resulting curve is coordinate system independent.
 - **Symmetry**: for $i = 0, \dots, 3$ it holds $f_i(1 - t) = f_{3-i}(t)$. This means that the curve does not change if the order of the control points is reversed.

Timmer's parametric cubic

- ▶ Proposed by Harry Timmer of McDonnell Douglas (Harry G. Timmer, *Alternative representation for parametric cubic curves and surfaces*, Computer-Aided Design, 12:25–28, 1980).
- ▶ Like Bézier curves, it interpolates the endpoints of the control polygon and it is tangent to the control polygon in those points. Moreover, it goes through the midpoint of the segment P_1P_2 .
- ▶ The blending functions are

$$\begin{aligned}f_0(t) &= (1 - 2t)(1 - t)^2, \\f_1(t) &= 4t(1 - t)^2, \\f_2(t) &= 4t^2(1 - t), \\f_3(t) &= (2t - 1)t^2.\end{aligned}$$

Timmer's parametric cubic



The curve goes through the midpoint of the segment P_1P_2 , but it is not tangent to this segment.

In particolare esce dall'inviluppo convesso !

Ball's cubic

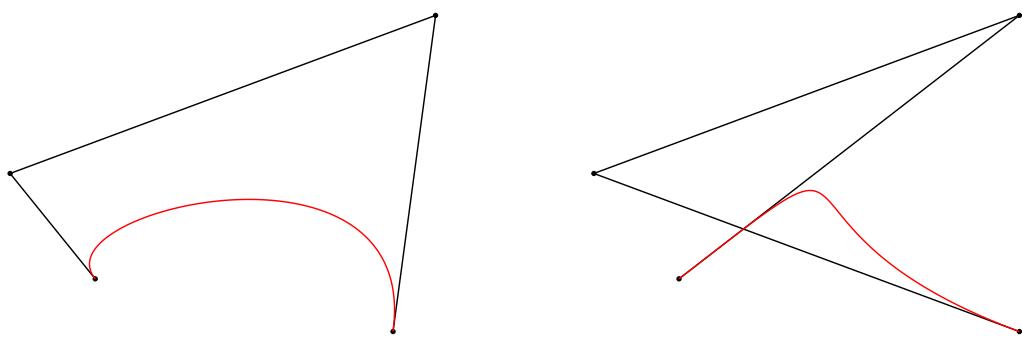
- ▶ Proposed by Alan Ball of the British Aircraft Corporation (Alan A. Ball, *Consurf. part 1: Introduction to the conic lofting tile*, Computer-Aided Design, 6:243–249, 1974).
 - ▶ Was used in BAC's in-house CAD system.
 - ▶ The blending functions are

$$\begin{aligned}
 & \text{quadratische} \quad \left\{ \begin{array}{l} f_0(t) = (1-t)^2, \\ f_1(t) = 2t(1-t)^2, \\ f_2(t) = 2t^2(1-t), \\ f_3(t) \equiv t^2. \end{array} \right. \quad \text{- cubische}
 \end{aligned}$$

- ▶ Note that, if $P_1 = P_2$, the curve becomes a quadratic Bézier curve:
$$P_2 t^2 (1-t) + P_1 t (1-t)^2$$

$$P_0(1-t)^2 + \overbrace{P_1 2t(1-t)^2 + P_1 2t^2(1-t)}^{P_1 2t(1-t)[t+(1-t)]} + P_3 t^2 = P_0(1-t)^2 + P_1 2t(1-t) + P_3 t^2.$$

Ball's cubic



Overhauser curves

- ▶ Developed and used at Ford Motor Company (A. W. Overhauser, *Analytic definition of curves and surfaces by parabolic blending*. Technical report, Ford Motor Company, May 1968).
- ▶ Also known as cubic Catmull-Rom splines.
- ▶ Interpolate all control points in a piecewise string of curve segments.
- ▶ The blending functions for a curve segment are

$$C(t) = \sum_{i=0}^n P_i f_i(t)$$

$$\Rightarrow C(0) = P_1, \quad C(1) = P_2$$

quindi la curva comincia in P_1
invece di P_0 e finisce in P_2
invece di P_3

P_0 e P_3 servono per controllare
le derivate agli estremi:

$$f_0(t) = -\frac{1}{2}t + t^2 - \frac{1}{2}t^3,$$

$$f_1(t) = 1 - \frac{5}{2}t^2 + \frac{3}{2}t^3,$$

$$f_2(t) = \frac{1}{2}t + 2t^2 - \frac{3}{2}t^3,$$

$$f_3(t) = -\frac{1}{2}t^2 + \frac{1}{2}t^3.$$

Overhauser curves

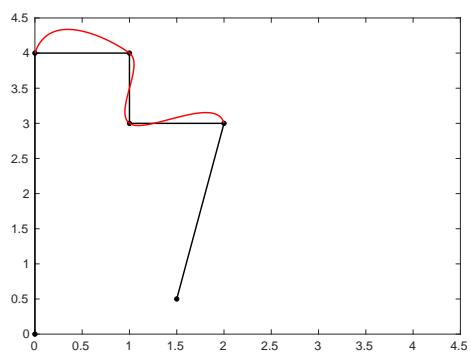
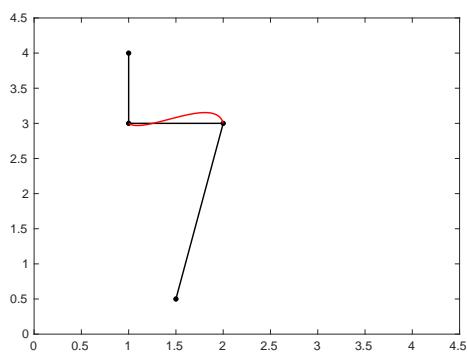
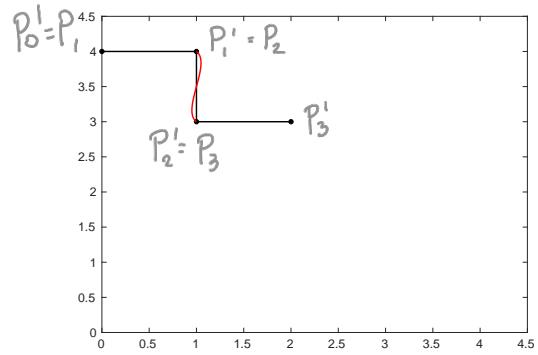
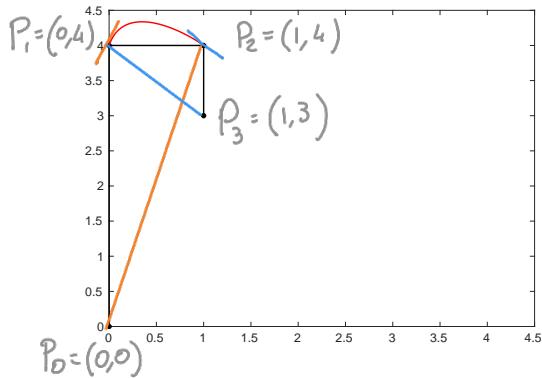
- ▶ A single Overhauser curve segment interpolates P_1 and P_2 .
- ▶ Slopes at control points:

$$K'(0) = \frac{1}{2}(P_2 - P_0),$$

$$K'(1) = \frac{1}{2}(P_3 - P_1).$$

- ▶ A second curve segment will be tangent to the first one if its first three control points are identical to the last three control points of the first curve.

Overhauser curves



Spline

Introduciamo il problema dell'interpolazione polinomiale

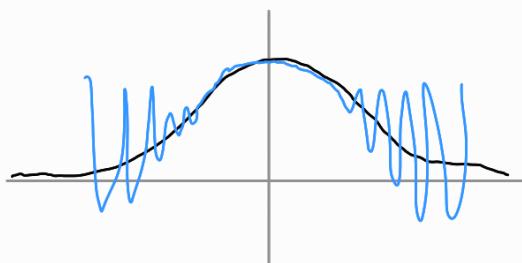
Fissati: nodi: $\{t_i\}$ con $a = t_0 < t_1 < \dots < t_n = b$ e una funzione $f: [a, b] \rightarrow \mathbb{R}$ continua, esiste ed è unico il polinomio di grado $\leq n$ $P_n(x)$ che interpola f nei nodi, ovvero $P_n(t_j) = f(t_j)$, $j=0, \dots, n$

Problema (dell'interpolazione base)

Dati: i punti $(t_0, f(t_0)), \dots, (t_n, f(t_n))$ determinare P_n
dove "determinare" = "saper valutare"

Oss Raffinare i nodi di valutazione $\{t_i\}$ non corrisponde necessariamente ad una migliore approssimazione polinomiale
di f

Ese (fenomeno di Runge) $f(x) = \frac{1}{1+25x^2}, x \in [-1, 1]$



Teo (di Faber)

Per ogni tavola di nodi (successione di n nodi, $n \in \mathbb{N}$) su $[a, b]$ esiste $f \in C[a, b]$ tale che la successione di polinomi di interpolazione non converge uniformemente a f

Tuttavia, se $[a, b] = [-1, 1]$, f assolutamente continua e $\{t_i\}$

nodi di Chebyshov, allora ho convergenza dei polinomi di interpolazione (vedi pacchetto CHEBFUN per MatLab)

La rappresentazione di una curva tramite polinomi di interpolazione (con un solo polinomio) ha l'inconveniente di essere non locale: modificare un solo nodo modifica tutta la curva

Per ovviare a questo inconveniente, usiamo l'interpolazione polinomiale a tratti, quindi parliamo di spline

Dati: $a = t_0 < t_1 < \dots < t_n = b$ e $f(t_0), \dots, f(t_n)$, voglio determinare una funzione $s(x): [a, b] \rightarrow \mathbb{R}$ tale che:

- $s(t_i) = f(t_i) \quad i=0, \dots, n$
- $s(t)|_{[t_i, t_{i+1}]} \text{ è un polinomio di grado } \leq d \text{ fisso, } i=0, \dots, n-1$
(spesso $d=3$)

eventualmente con condizioni aggiuntive.

Esempi classici:

- Spline cubiche di Hermite:

Dati anche $f'(t_i)$ per $i=0, \dots, n$, si impone che:

- $s(t_i) = f(t_i)$ per $i=0, \dots, n$
- $s(t)|_{[t_i, t_{i+1}]} \text{ sia un polinomio di grado } \leq 3, \quad i=0, \dots, n-1$
- $(s|_{[t_i, t_{i+1}]})'(t_i) = f'(t_i) \quad (s|_{[t_i, t_{i+1}]})'(t_{i+1}) = f'(t_{i+1})$
 $i=0, \dots, n-1$

Con queste condizioni s'è univocamente determinata e, per nodi equispaziati di ampiezza $h = \frac{b-a}{n}$, s'ha

che $\|f - s\| \xrightarrow{n \rightarrow +\infty} 0$ come $O(h^4)$

- Spline cubiche C^2

Dati: $\{t_i\}$, $\{f(t_i)\}$ come sopra, s : impone che:

- $s(t_i) = f(t_i)$ $i = 0, \dots, n$
- $s|_{[t_i, t_{i+1}]} \text{ è polinomio di grado } \leq 3, \quad i = 0, \dots, n-1$
- s è globalmente C^2

Detta $s_i(t) := s(t)|_{[t_i, t_{i+1}]}$, $s_i(t)$ è un polinomio al più cubico quindi è determinato da 4 coefficienti.

Le condizioni da impostare sono

- $s_i(t_i) = f(t_i), \quad s_i(t_{i+1}) = f(t_{i+1}), \quad i = 0, \dots, n-1$
- $s'_i(t_{i+1}) = s'_{i+1}(t_{i+1}) \quad \left. \right\} i = 0, \dots, n-2$
- $s''_i(t_{i+1}) = s''_{i+1}(t_{i+1}) \quad \left. \right\} i = 0, \dots, n-2$

Abbiamo quindi un sistema lineare $4n-2$ equazioni e $4n$ incognite, quindi in generale s non è univocamente determinata e s : hanno due grad: di libertà:

- se conosco $f'(a), f'(b)$ posso impostare
 $s'(a) = f'(a), \quad s'(b) = f'(b) \quad (\text{spline COMPLETE})$
- $s''(a) = s''(b) = 0 \quad (\text{spline NATURALI})$
- $s_0'''(t_1) = s_1'''(t_1) \text{ e } s_{n-2}'''(t_{n-1}) = s_{n-1}'''(t_{n-1}) \quad (\text{NOT-A-KNOT})$
- Se f è periodica si può impostare che anche s lo sia

Anche qui abbiamo risultati di convergenza del

tipo $\|f - s\| \xrightarrow{n \rightarrow +\infty} 0$ sotto opportune ipotesi. Per nodi:

equi-spaziat: d: ampiezza h si ha convergenza:

- come $O(h^4)$ per spline complete
- come $O(h^2)$ per spline naturali

In pratica, posso riscrivere le condizioni come un sistema lineare dei momenti, cioè le incognite sono $s''(t_i)$, e il sistema costruito è tipicamente tridiagonale, diagonale, ben condizionato. Si può quindi ottenere la soluzione numerica in $O(n)$

La valutazione d: s ha un costo logaritmico per l'individuazione dell'intervallo giusto e un costo fisso (basso) per la valutazione d: un polinomio cubico

Spline di Bézier

Idea: parametrizzare ciascun $s_i(t)$ come una curva d: Bézier

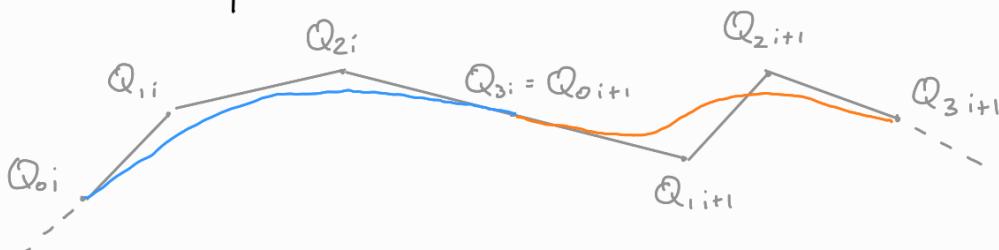
Def Siano $a = u_0 < \dots < u_n = b$, dico che $X: [a, b] \rightarrow \mathbb{R}^d$ è

una **SPLINE DI BÉZIER** d: grado m se

$X_i := X|_{[u_i, u_{i+1}]} \text{ è una curva d: Bézier d: grado } \leq m$ del

$$\text{tipo } X_i(u) = \sum_{j=0}^m Q_{ji} B_j^{(m)} \left(\frac{u - u_i}{u_{i+1} - u_i} \right), \quad u \in [u_i, u_{i+1}], \quad i = 0, \dots, n-1$$

Impongo raccordi: C^0, C^1, C^2, \dots usando le condizioni che conosciamo sui punti di controllo



- $C^0 \Rightarrow Q_{3i} = Q_{0,i+1}$
- $C^1 \Rightarrow \frac{Q_{3i} - Q_{2i}}{\Delta Q_{2i}} = \frac{Q_{1,i+1} - Q_{0,i+1}}{\Delta Q_{0,i+1}}$
- $C^2 \Rightarrow \frac{\Delta Q_{2i} - \Delta Q_{1,i}}{\Delta^2 Q_{1,i}} = \frac{\Delta Q_{1,i+1} - \Delta Q_{0,i+1}}{\Delta^2 Q_{0,i+1}}$

Rivediamo le tecniche d'interpolazione con spline d'Bezier

- Interpolazione d'Hermite con spline d'Bezier

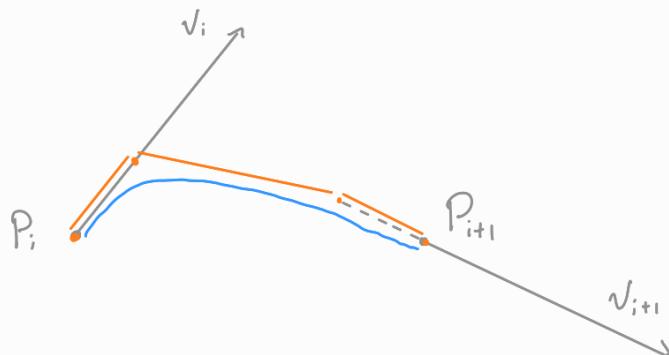
Sono dati $(P_i, v_i)_{i=0, \dots, n}$ $P_i \in \mathbb{R}^d$ (affine)
 $v_i \in \mathbb{R}^d$ (vettoriale)

Cerco una spline d'Bezier, cioè de' punti d'controllo

$(Q_{ji})_{\substack{i=0, \dots, 3 \\ j=0, \dots, n-1}}$ che interpolano i P_i e abbiano derivate v_i in P_i

Sicuramente $Q_{0i} = P_i$ e $Q_{3i} = P_{i+1}$ affinché X_i abbia estremi P_i e P_{i+1} . Affinché X_i abbia derivata v_i in P_i deve valere $v_i = 3(Q_{1i} - Q_{0i})$ quindi impongo $Q_{1i} = P_i + \frac{1}{3}v_i$ e analogamente $Q_{2i} = P_{i+1} - \frac{1}{3}v_{i+1}$

Quindi: ogn: curva d'Bezier è univocamente determinata



Poiché modificare P_i e v_i modifica solo Q_{0i} , Q_{1i} , Q_{2i} , e Q_{3i} , diciamo che la curva è locale

- Interpolazione d: Lagrange (C^2) con spline d: Bézier

Dati: $P_0, \dots, P_n \in \mathbb{R}^d$ cerchiamo una spline d: Bézier

$X(u)$ d: classe C^2 t.c. $X(u_i) = P_i$

Usiamo come Q_{ji} ($j=0, \dots, 3$, $i=0, \dots, n-1$) la rappresentazione data dall'interpolazione d: Hermite con i v_i incogniti.

Questo basta per assicurare C^0 e C^1

Imponiamo che valga anche C^2 (nei raccordi internamente è già garantito dalle curve d: Bézier).

I vincoli sono, per $i=1, \dots, n-1$

$$\text{dove } h_i = u_{i+1} - u_i$$

$$\frac{\Delta^2 Q_{1,i-1}}{h_{i-1}^2} = \frac{\Delta^2 Q_{0,i}}{h_i^2} \quad \Delta^2 Q_{1,i-1} = Q_{1,i-1} - 2Q_{2,i-1} + Q_{3,i-1}$$

$$\Delta^2 Q_{0,i} = Q_{0,i} - 2Q_{1,i} + Q_{2,i}$$

$$\Leftrightarrow \frac{1}{h_{i-1}^2} \left(P_{i-1} - \frac{v_{i-1}}{3} - 2P_i + 2\frac{v_i}{3} + P_i \right) = \\ = \frac{1}{h_i^2} \left(P_i - 2P_i - 2\frac{v_i}{3} + P_{i+1} - \frac{v_{i+1}}{3} \right)$$

$$\Leftrightarrow v_{i-1} \left(\frac{1}{3h_{i-1}^2} \right) + v_i \left(\frac{2}{3h_{i-1}^2} + \frac{2}{3h_i^2} \right) + v_{i+1} \left(\frac{1}{3h_i^2} \right) = \\ = -\frac{1}{h_{i-1}^2} P_{i-1} + \left(\frac{1}{h_{i-1}^2} - \frac{1}{h_i^2} \right) P_i + \frac{1}{h_i^2} P_{i+1}$$

$$\Leftrightarrow v_{i-1} \left(\frac{1}{3h_{i-1}^2} \right) + v_i \left(\frac{2}{3h_{i-1}^2} + \frac{2}{3h_i^2} \right) + v_{i+1} \left(\frac{1}{3h_i^2} \right) = \\ = \frac{P_i - P_{i-1}}{h_{i-1}^2} + \frac{P_{i+1} - P_i}{h_i^2}$$

Quindi i v_i sono soluzioni d: un sistema lineare tridiagonale di $n-1$ equazioni e $n+1$ incognite. Possiamo riscrivere

il vincolo ponendo $\mu_i = \frac{h_i}{h_i + h_{i-1}}$ e ottienendo

$$\mu_i v_{i-1} + 2v_i + (1-\mu_i) v_{i+1} = 3 \left(\mu_i \frac{P_i - P_{i-1}}{h_{i-1}} + (1-\mu_i) \frac{P_{i+1} - P_i}{h_i} \right)$$

Restano due gradi di libertà quindi possiamo fissare
2 condizioni aggiuntive, in modo analogo a quanto visto

Cerchiamo una rappresentazione delle curve spline in \mathbb{R}^d
del tipo $C(u) = \sum_{i=0}^n P_i f_i(u)$ con $u \in [a, b]$ dove $P_0, \dots, P_n \in \mathbb{R}^d$
sono punti di controllo e f_i sono funzioni polinomiali a
tratti rispetto a $a = u_0 < \dots < u_n = b$

Piecewise polynomial interpolation

We would like to represent a curve in the form

$$\mathcal{C}(t) = \sum_{i=0}^n f_i(t) P_i$$

where

- ▶ P_0, \dots, P_n are the **control points**,
- ▶ $f_0(t), \dots, f_n(t)$ form a basis of the space of **piecewise polynomial functions** of prescribed degree and regularity w.r.t. a given knot sequence.

B-splines

The development and analysis of B-splines begins with I. J. Schönberg (1946).

Several possible definitions:

- ▶ divided differences of truncated power functions (Curry and Schönberg),
- ▶ blossoming (Ramshaw),
- ▶ recurrence relation (De Boor, Cox and Mansfield).

The recurrence formula is the most useful for practical implementation.

We follow here Les Piegl and Wayne Tiller, *The NURBS Book*.

B-spline basis: definition

✓ non necessariamente
distanziati!

Let u_0, \dots, u_m be a nondecreasing sequence of real numbers (**knots**). The i -th B-spline basis function of **degree p** (order $p+1$), denoted by $N_{i,p}(u)$, is defined by the recursive relation

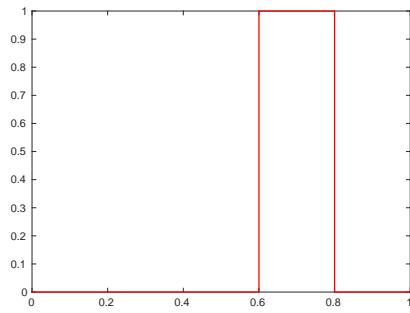
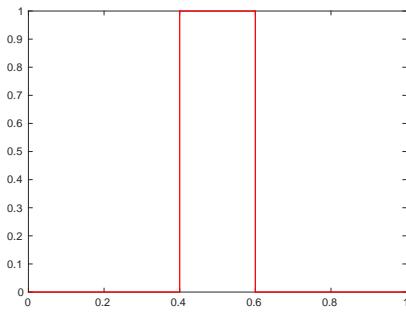
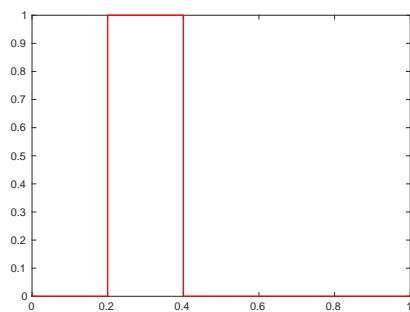
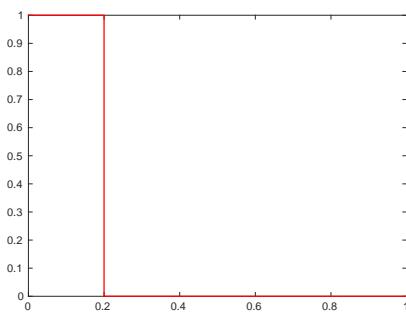
$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \leq u < u_{i+1} \\ 0 & \text{otherwise,} \end{cases} \quad \text{and}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u),$$

for $p > 0$, where terms with zero denominator are taken to be zero.
Here $i = 0, 1, \dots, m - p - 1$.

B-spline basis: example

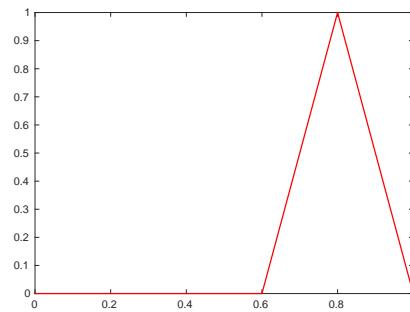
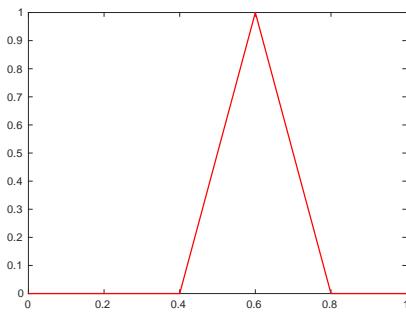
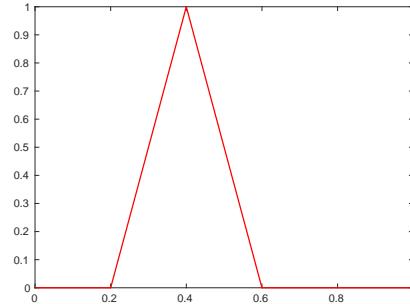
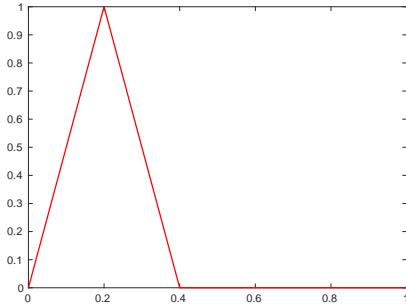
Take knots $0, 0.2, 0.4, 0.6, 0.8, 1$.



Graphs show $N_{i,0}(u)$ for $i = 0, 1, 2, 3$. (non c'era spazio per $i=4$)

B-spline basis: example

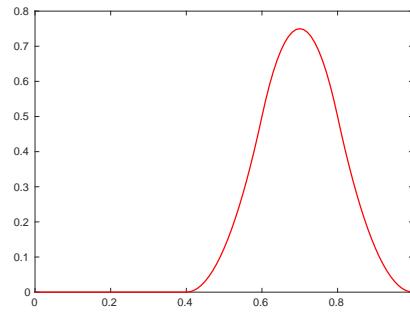
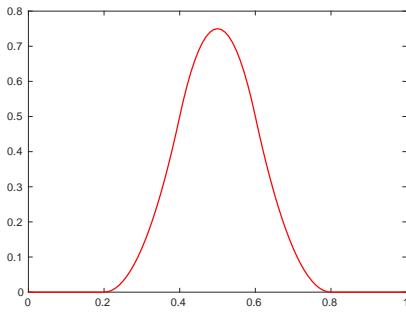
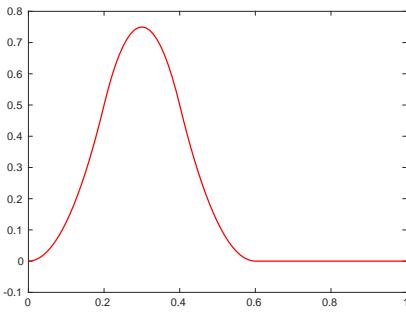
$$N_{i,1}(u) = \frac{u - u_i}{u_{i+1} - u_i} N_{i,0}(u) + \frac{u_{i+2} - u}{u_{i+2} - u_{i+1}} N_{i+1,0}(u)$$



Graphs show $N_{i,1}(u)$ for $i = 0, 1, 2, 3$.

B-spline basis: example

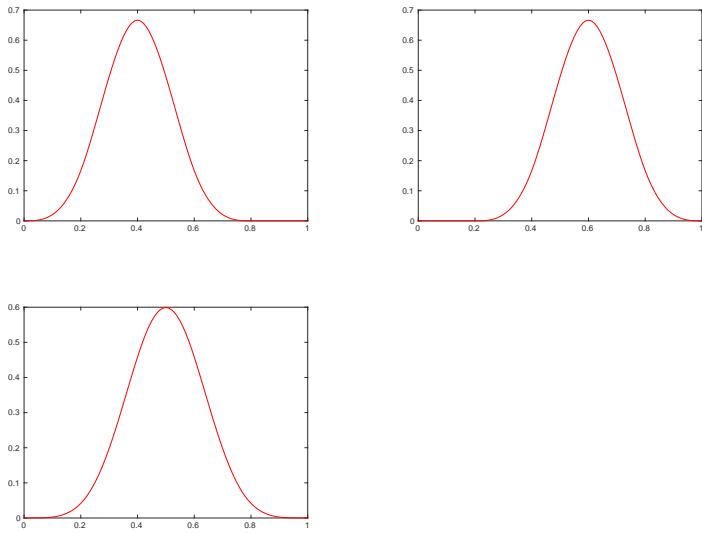
$$N_{i,2}(u) = \frac{u - u_i}{u_{i+2} - u_i} N_{i,1}(u) + \frac{u_{i+3} - u}{u_{i+3} - u_{i+1}} N_{i+1,1}(u)$$



Graphs show $N_{i,2}(u)$ for $i = 0, 1, 2$.

B-spline basis: example

$$N_{i,3}(u) = \frac{u - u_i}{u_{i+3} - u_i} N_{i,2}(u) + \frac{u_{i+4} - u}{u_{i+4} - u_{i+1}} N_{i+1,2}(u)$$



Graphs show $N_{0,3}(u)$, $N_{1,3}(u)$, and
 $N_{0,4}(u) = \frac{u-u_0}{u_4-u_0} N_{0,3}(u) + \frac{u_5-u}{u_5-u_1} N_{1,3}(u)$.

B-spline basis: multiple endpoints

- ▶ In order to cover properly the whole interval $[u_0, u_m]$, we will add extra knots at the endpoints.
- ▶ For B-splines of degree p , the endpoint knots are usually taken to have multiplicity $p + 1$:

$$u_0 = \dots = u_p \leq u_{p+1} \leq \dots u_{m-p+1} \leq u_{m-p} = \dots = u_m$$

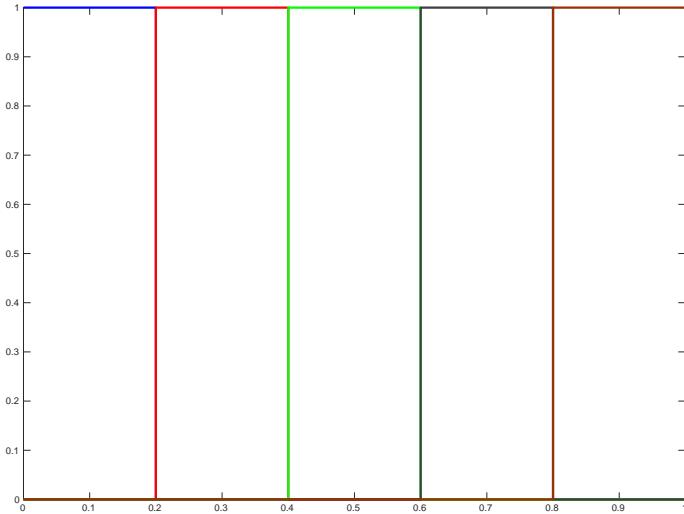
(such knot vectors are called **nonperiodic** or **clamped** or **open**).

- ▶ Back to our example on $[0, 1] \dots$

NON SONO "COBERTI BENE" DALLE
FUNZIONI APPENA VISTE

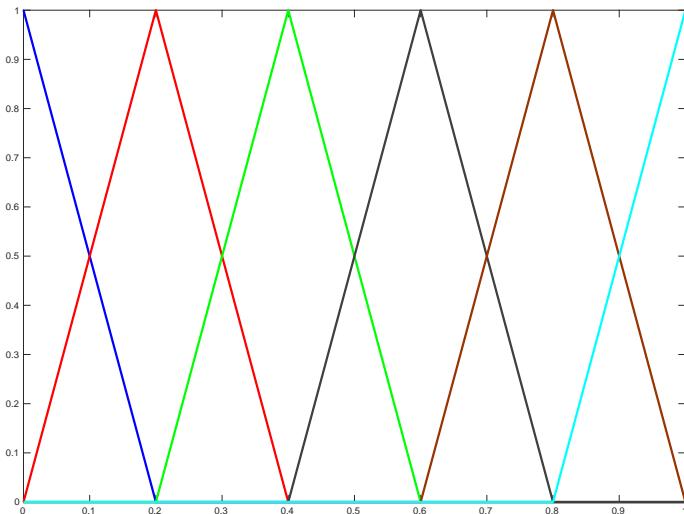
B-spline basis: example

Take knots $0, 0.2, 0.4, 0.6, 0.8, 1$. The B-spline basis functions of degree zero are:



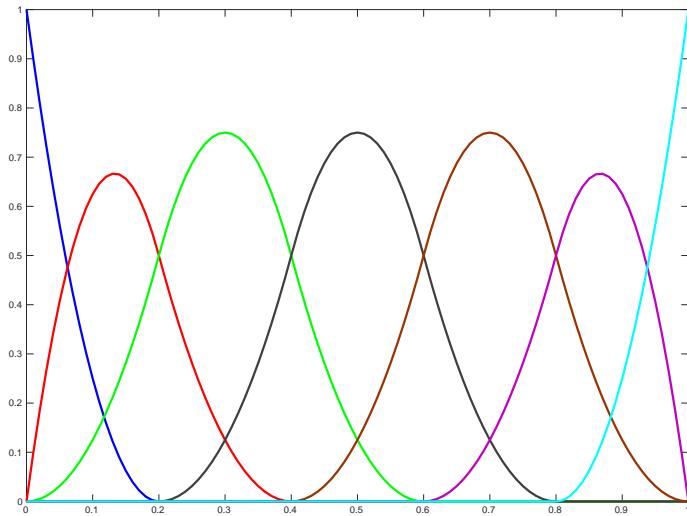
B-spline basis: example

Take knots $0, 0, 0.2, 0.4, 0.6, 0.8, 1, 1$. The B-spline basis functions of degree one are:



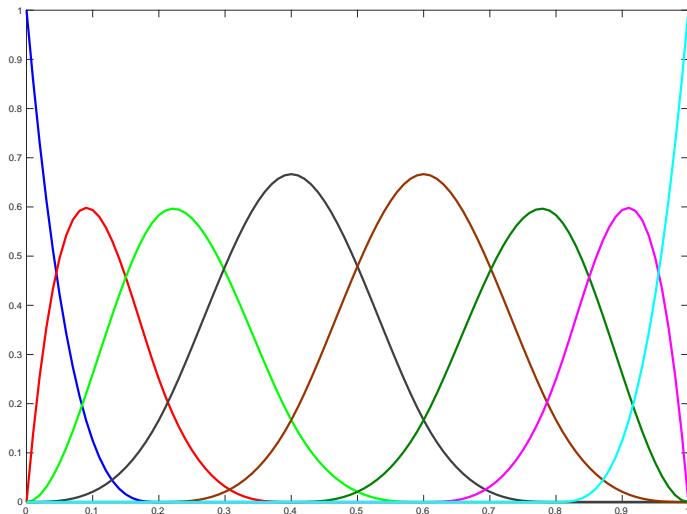
B-spline basis: example

Take knots $0, 0, 0, 0.2, 0.4, 0.6, 0.8, 1, 1, 1$. The B-spline basis functions of degree two are:



B-spline basis: example

Take knots $0, 0, 0, 0, 0.2, 0.4, 0.6, 0.8, 1, 1, 1, 1$. The B-spline basis functions of degree three are:



B-spline basis: properties

The computation of B-spline basis functions follows a triangular table

$$\begin{matrix} N_{0,0} & & & \\ & N_{0,1} & & \\ N_{1,0} & & N_{0,2} & \\ & N_{1,1} & & N_{0,3} \\ N_{2,0} & & N_{1,2} & \\ & N_{2,1} & & N_{1,3} \\ N_{3,0} & & N_{2,2} & \vdots \\ & N_{3,1} & & \vdots \\ N_{4,0} & & \vdots & \\ \vdots & & & \end{matrix}$$

B-spline basis: properties

Take a knot sequence u_0, \dots, u_m and choose a degree p . The B-spline basis functions have a number of important properties (besides actually being a basis...more on this later):

- ▶ **Local support:** $N_{i,p}(u)$ is zero outside the interval $[u_i, u_{i+p+1}]$.
- ▶ In any given knot span $[u_i, u_{i+1}]$, at most $p + 1$ basis functions are nonzero, namely $N_{i-p,p}, \dots, N_{i,p}$.
- ▶ **Nonnegativity:** $N_{i,p}(u) \geq 0$ for all i , p and u .
- ▶ **Partition of unity:** $\sum_{j=i-p}^i N_{j,p}(u) = 1$ for all $u \in [u_i, u_{i+1}]$
- ▶ **Continuity:** $N_{i,p}(u)$ is C^∞ in the interior of a knot span (polynomial). At a knot of multiplicity k , it is $p - k$ times continuously differentiable.
- ▶ **Maximum:** for $p > 0$, $N_{i,p}(u)$ attains exactly one maximum value.

B-spline basis: derivatives

The (right) derivative of a basis function is given by

$$N'_{i,p}(u) = \frac{p}{u_{i+p} - u_i} N_{i,p-1}(u) - \frac{p}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u).$$

(Proof by induction on p).

The k -th derivative of a basis function is given by

$$\begin{aligned} N_{i,p}^{(k)}(u) &= p \left(\frac{N_{i,p-1}^{(k-1)}(u)}{u_{i+p} - u_i} - \frac{N_{i+1,p-1}^{(k-1)}(u)}{u_{i+p+1} - u_{i+1}} \right) = \\ &= \frac{p!}{(p-k)!} \sum_{j=0}^k a_{k,j} N_{i+j,p-k}, \end{aligned}$$

with

$$\begin{aligned} a_{0,0} &= 1, & a_{k,0} &= \frac{a_{k-1,0}}{u_{i+p-k+1} - u_i}, \\ a_{k,j} &= \frac{a_{k-1,j} - a_{k-1,j-1}}{u_{i+p+j-k+1} - u_{i+j}}, j = 1, \dots, k-1, & a_{k,k} &= \frac{-a_{k-1,k-1}}{u_{i+p+1} - u_{i+k}}. \end{aligned}$$

B-spline basis: exercises

1. What are the B-spline basis functions of degree 2 for the knot sequence $(0, 0, 0, 1, 1, 1)$?
2. Compute and draw the B-spline basis functions of degrees 0, 1 and 2 for the knot sequence $(0, 0, 0, 1, 2, 3, 4, 4, 5, 5, 5)$.

Ex [1 d. Bspline.pdf]

(0, 0, 0, 1, 1, 1) nodi, determinare le $N_{i,p}(u)$

Sol La definizione per ricorrenza era

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u)$$

Allora le Bspline di grado 0 sono

$$N_{0,0} = 0$$

$$N_{1,0} = 0$$

$$N_{2,0}(u) = \begin{cases} 1 & \text{se } u \in [0,1] \\ 0 & \text{altrimenti} \end{cases}$$

$$N_{3,0} = 0$$

$$N_{4,0} = 0$$

Le Bspline di grado 1 sono

$$N_{0,1} = 0$$

$$N_{1,1}(u) = \frac{u_3 - u}{u_3 - u_2} \cdot N_{0,0}(u) = 1 - u$$

$$N_{2,1}(u) = \frac{u - u_2}{u_3 - u_2} \cdot N_{0,0}(u) = u$$

$$N_{3,1} = 0$$

Le Bspline di grado 2 sono

$$N_{0,2}(u) = \frac{u_3 - u}{u_3 - u_1} \cdot N_{1,1}(u) = (1 - u)^2 = B_0^{(2)}(u)$$

$$N_{1,2}(u) = \frac{u - u_1}{u_3 - u_1} \cdot N_{1,1}(u) + \frac{u_4 - u}{u_4 - u_2} \cdot N_{2,1}(u) = 2u(1 - u) = B_1^{(2)}(u)$$

$$N_{2,2}(u) = \frac{u - u_2}{u_4 - u_2} \cdot N_{2,1}(u) = u^2 = B_2^{(2)}(u)$$

In generale le Bspline di grado p sul vettore $(\overbrace{0 \dots 0}^{p+1} \overbrace{1 \dots 1}^{p+1})$ sono i polinomi di Bézier di grado p

Abbiamo visto che per le Bspline può convenire considerare nodi con molteplicità multipla

In generale, data una spline di grado p , la regolarità in un nodo di molteplicità K è \mathcal{C}^{p-k}

Proprietà

- **RICORRENZA:** $N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u)$
- **SUPPORTO LOCALE:** $N_{i,p}(u)$ ha supporto in $[u_i, u_{i+p+1}]$
- Su $[u_i, u_{i+p}]$ sono (potenzialmente) non nulli le $N_{j,p}(u)$, $j=i-p, \dots, i$
- Non-negatività: $N_{i,p}(u) \geq 0 \quad \forall i, p, u$

Dim Per induzione su p

$p=0$ OK per definizione di $N_{i,0}(u)$

$p-1 \Rightarrow p$ per ricorrenza

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u)$$

dove il primo termine è prodotto di termini non negativi su $u \in [u_i, u_{i+p}]$ e si annulla per $u \notin [u_i, u_{i+p}]$
poiché si annulla $N_{i,p-1}(u)$; in generale il primo termine è non negativo $\forall u$

Analogamente il secondo termine è non negativo $\forall u$

Quindi anche $N_{i,p}(u) \geq 0$ poiché somma di termini non negativi

□

- Partizione dell'unità: $\sum_{j=0}^n N_{j,p}(u) = 1 \quad \forall u \in [u_0, u_m] = [a, b]$

Dim Sia $i < u < u_{i+1}$, allora per la terza

proprietà la tesi diventa $\sum_{j=i-p}^i N_{j,p}(u) = 1$

Procediamo per induzione su p

$p=0$ OK per definizione

$$\begin{aligned}
 p-1 \Rightarrow p & \quad \sum_{j=i-p}^i N_{j,p}(u) = \sum_{j=i-p}^i \left(\frac{u - u_j}{u_{j+p} - u_j} N_{j,p-1}(u) + \frac{u_{j+p+1} - u}{u_{j+p+1} - u_{j+1}} N_{j+1,p-1}(u) \right) = \\
 & = \sum_{j=i-p}^i \frac{u - u_j}{u_{j+p} - u_j} N_{j,p-1}(u) + \sum_{j=i-p}^i \frac{u_{j+p+1} - u}{u_{j+p+1} - u_{j+1}} N_{j+1,p-1}(u) = \\
 & \text{per supporto delle } N_{j,p-1} \text{ (e shift sulla seconda sommatoria)} \\
 & \text{sì annulla il termine di indice } j=i-p \\
 & = \sum_{j=i-p+1}^i \left(\dots \right) + \sum_{j=i-p+1}^i \frac{u_{j+p} - u}{u_{j+p} - u_j} N_{j,p-1}(u) = \\
 & = \sum_{j=i-(p-1)}^i \underbrace{\left(\frac{u - u_j + u_{j+p} - u}{u_{j+p} - u_j} \right)}_{=1} N_{j,p-1}(u) = \\
 & = \sum_{j=i-(p-1)}^i N_{j,p-1}(u) = 1 \quad \text{per hp induttiva}
 \end{aligned}$$

□

Siano $u_0 < u_1 < \dots < u_K$ nodi e fissiamo un grado p e degli interi r_0, \dots, r_K tutti ≥ -1

Considero le funzioni polinomiali a tratti (rispetto agli u_i) di grado p aventi ordine di regolarità r_i nel nodo u_i , $i=0, \dots, K$ (dove regolarità -1 vuol dire discontinuità, e r_0 e r_K si riferiscono al raccordo tra la funzione considerata e la funzione identicamente nulla)

Queste funzioni formano uno spazio vettoriale V . Cosa si può

dire su $\dim V = ?$

Nel caso d: $r_0 = \dots = r_k = -1$ si ha per ogni intervallo s : ha un polinomio indipendente dagli altri, quind. si avrà $\dim V = k(p+1)$

In generale, poiché ad ogni aumento ad un r_j sto togliendo un grado d: libertà alle funzioni, si ha $\dim V = K(p+1) - \sum_{j=0}^k (r_j + 1)$

Consideriamo le Bspline d: grado p definite rispetto al vettore dei nodi: $(\underbrace{u_0, \dots, u_0}_{s_0}, \underbrace{u_1, \dots, u_1}_{s_1}, \dots, \underbrace{u_k, \dots, u_k}_{s_k})$ dove $s_j \geq 1$

Poiché sappiamo che dato un nodo d: molteplicità s_j l'ordine d: continuità è $p-s_j$, per imporre i vincoli r_0, \dots, r_k scelti prima dovrà valere $r_j = p-s_j$, quind: $1 \leq s_j \leq p+1$

Avremo allora che queste Bspline appartengono a V con $\dim V = K(p+1) - \sum_{j=0}^k (r_k + 1) = K(p+1) - k(p+1) + \sum_{j=0}^k s_j - (k+1) =$

$$= \sum_{j=0}^k s_j - p - 1 = m - p \quad \text{dove } m+1 = \sum_{j=0}^k s_j \text{ è il numero d: nodi considerati}$$

Poiché le Bspline d: grado p su m nodi è proprio $m-p$, se mostriamo che queste sono indipendenti avremo che sono una base d: V

Polinomi duali

Def Sia $N_{i,p}$ Bspline associata al vettore d: nodi $\{u_j\}_{j=0, \dots, m}$
Il **POLINOMIO DUALE** corrispondente è

$$\Psi_{i,p}(y) = (y - u_{i+1}) - (y - u_{i+p}) \quad (\Psi_{i,0}(y) = 1)$$

Proprietà (RICORRENZA DUALE)

$$(y-x) \Psi_{i,p-1}(y) = \frac{x-u_i}{u_{i+p}-u_i} \Psi_{i,p}(y) + \frac{u_{i+p}-x}{u_{i+p}-u_i} \Psi_{i-1,p}(y)$$

Dimo Fissato $y \in \mathbb{R}$ e considero $\ell_y: \mathbb{R} \rightarrow \mathbb{R}$, $\ell_y(x) = y-x$

uso l'interpolazione lineare su $(u_i, \ell_y(u_i))$, $(u_{i+p}, \ell_y(u_{i+p}))$
(poiché ℓ_y è una retta vale)

$$\ell_y(x) = \frac{x-u_i}{u_{i+p}-u_i} \ell_y(u_{i+p}) + \frac{u_{i+p}-x}{u_{i+p}-u_i} \ell_y(u_i)$$

e moltiplicando per $\Psi_{i,p-1}(y)$ si ha

$$(y-x) \Psi_{i,p-1}(y) = \frac{x-u_i}{u_{i+p}-u_i} \frac{(y-u_{i+p}) \Psi_{i,p-1}(y)}{\Psi_{i,p}(y)} + \frac{u_{i+p}-x}{u_{i+p}-u_i} \frac{(y-u_i) \Psi_{i,p-1}(y)}{\Psi_{i-1,p}(y)}$$

□

Derivando rispetto a x la ricorrenza duale si ha

$$-\Psi_{i,p-1}'(y) = \frac{1}{u_{i+p}-u_i} (\Psi_{i,p}(y) - \Psi_{i-1,p}(y))$$

ovvero $\Psi_{i,p-1}'(y) = \frac{1}{u_{i+p}-u_i} (\Psi_{i-1,p}(y) - \Psi_{i,p}(y))$

Teo (INDIPENDENZA DI MARSDEN)

Fissato $i \leq h \leq i+p$ t.c. $u_h < u_{h+1}$ e $x \in [u_h, u_{h+1}]$

Valgono le seguenti relazioni:

$$(1) \quad (y-x)^p = \sum_{i=h-p}^h N_{i,p}(x) \Psi_{i,p}(y), \quad y \in \mathbb{R}$$

$$(2) \quad (y-x)^p = \sum_{i=h-p}^h N_{i,p}^{\{h\}}(x) \Psi_{i,p}(y) \quad x, y \in \mathbb{R}$$

dove $N_{i,p}^{\{h\}}(x): \mathbb{R} \rightarrow \mathbb{R}$ è il polinomio che ristretto a $[u_h, u_{h+1}]$ coincide con $N_{i,p}(x)$

Mostriamo che le Bspline $\{N_{i,p}(u)\}_{i=0,\dots,n}$ rispetto
al vettore dei nodi $(u_j)_{j=0,\dots,m}$ ($m=n+p+1$) sono base dello
spazio vettoriale V delle funzioni polinomiali a tratti di grado p
rispetto ad una data partizione di $[a,b]$ con regolarità fissata
nei nodi della partizione.

In particolare, per osservazioni fatte sulla dimensione dello spazio,
ci resta solo da dimostrare l'indipendenza delle $\{N_{i,p}(u)\}$.

Cominciamo dimostrandone l'ultimo teorema enunciato

Dim (indipendenza di Marsden)

Osserviamo che ② segue immediatamente da ① per continuità

Mostriamo quindi ① per induzione su p

$p=0$ Per $p=0$ l'enunciato diventa

$$1 = (y-x)^0 = \sum_{i=h-0}^h N_{i,0}(x) \Psi_{i,0}(y) = \frac{N_{h,0}(x)}{=1} \frac{\Psi_{h,0}(y)}{=1}$$

che è verificata

$p-1 \Rightarrow p$ Per il passo induttivo riscriviamo

$$\begin{aligned} (y-x)^p &= (y-x)(y-x)^{p-1} = \quad \text{per la p induttiva} \\ &= (y-x) \sum_{i=h-p+1}^h \Psi_{i,p-1}(y) N_{i,p-1}(x) = \\ &= \sum_{i=h-p+1}^h \left[\frac{x-u_i}{u_{i+p}-u_i} \Psi_{i,p}(y) + \frac{u_{i+p}-x}{u_{i+p}-u_i} \Psi_{i-1,p}(y) \right] N_{i,p-1}(x) = \\ &= \sum_{i=h-p+1}^h \frac{x-u_i}{u_{i+p}-u_i} \Psi_{i,p}(y) N_{i,p-1}(x) + \\ &\quad + \sum_{i=h-p+1}^h \frac{u_{i+p}-x}{u_{i+p}-u_i} \Psi_{i-1,p}(y) N_{i,p-1}(x) = \end{aligned}$$

Poiché per $i=h-p$ il termine della prima sommatoria si annulla posso partire da $i=h-p$

Poiché per $i=h+1$ il termine della seconda sommatoria si annulla, posso arrivare a $i=h+1$ e poi fare uno shift

$$\begin{aligned}
 &= \sum_{i=h-p}^h (\text{" }) + \sum_{i=h-p}^h \frac{u_{i+p+1}-x}{u_{i+p+1}-u_{i+1}} \Psi_{i,p}(y) N_{i+1,p-1}(x) = \\
 &= \sum_{i=h-p}^h \left[\Psi_{i,p}(y) \left(\underbrace{\frac{x-u_i}{u_{i+p}-u_i} N_{i,p-1}(x) + \frac{u_{i+p+1}-x}{u_{i+p+1}-u_{i+1}} N_{i+1,p-1}(x)}_{\text{formula di ricorrenza di } N_{i,p}(x)} \right) \right] = \\
 &= \sum_{i=h-p}^h \Psi_{i,p}(y) N_{i,p}(x)
 \end{aligned}$$

□

Corollari:

(1) Rappresentazione locale dei polinomi:

Nelle stesse ip del teorema, se $p \geq k$ si ha

$$x^k = \sum_{i=h-p}^h \left[(-1)^k \frac{k!}{p!} D^{(p-k)} \Psi_{i,p}(0) \right] N_{i,p}(x)$$

(2) Partizione dell'unità locale

$$\sum_{i=h-p}^h N_{i,p}(x) \equiv 1 \quad \forall x \in [u_h, u_{h+1})$$

(3) Indipendenza lineare locale

Sia $\{N_{i,p}(x)\}_{i=h-p, \dots, h}$ sia $\{\Psi_{i,p}(y)\}_{i=h-p, \dots, h}$ sono basi di P_p su $[u_h, u_{h+1})$

Dim

(1) Basta derivare l'identità di Mersden $p-k$ volte rispetto a y , ottenendo:

⋮

$$p(p-1)\dots(k+1)(y-x)^k = \sum_{i=h-p}^h D^{(p-k)} \Psi_{i,p}(y) N_{i,p}(x)$$

Valutando in $y=0$ ottengo

$$\frac{p!}{k!} (-x)^k = \sum_{i=h-p}^h D^{(p-k)} \Psi_{i,p}(0) N_{i,p}(x)$$

da cui segue immediatamente la tesi d: ①

(2) Se considero il risultato d: ① in $k=0$ si ottiene

$$1 = \sum_{i=h-p}^h \frac{1}{p!} D^{(p)} \Psi_{i,p}(0) N_{i,p}(x)$$

$= p!$, poiché $\Psi_{i,p}(y) = y^p + \text{"roba di grado } < p\text{"}$

Cioè la tesi d: ②

(3) Dalla ① deduco che le $\{N_{i,p}(x)\}_{i=h-p, \dots, h}$ sono una base locale poiché generano $\{x^k\}_{i=h-p, \dots, h}$ (che è a sua volta base) e sono in numero giusto

Per le $\{\Psi_{i,p}(y)\}_{i=h-p, \dots, h}$ vale un ragionamento analogo

□

Passiamo ora alle proprietà locali delle $N_{i,p}(x)$. Sia $(U_j)_{j=0, \dots, m}$ il vettore dei nodi, in cui chiediamo che le molteplicità dei nodi non sia così elevata da avere delle B-spline interne identicamente nulle, ovvero chiediamo che $U_j < U_{j+p+1}$ per $j=0, \dots, n$. Chiediamo anche che i nodi estremi abbiano entrambi molteplicità $p+1$.

Dimostriamo che le $\{N_{i,p}(u)\}_{i=0, \dots, n}$ sono indipendenti su tutto $[U_0, U_n]$

Sia $s(x) = \sum_{j=0}^n c_j N_{j,p}(x) \equiv 0$ (vogliamo mostrare $c_0 = \dots = c_n = 0$)

Fisso j arbitrario tra 0 e n . Per le ipotesi fatte esiste m_j t.c. $j \leq m_j \leq j+p$ e $U_{m_j} < U_{m_j+1}$

Restringendo $x \in [u_{m_j}, u_{m_j+1})$, per supporto locale abbiamo

$$S(x) = \sum_{i=m_j-p}^{m_j} c_i N_{i,p}(x) = 0$$

Per l'indipendenza lineare locale otteniamo $c_{m_j-p} = \dots = c_{m_j} = 0$

quindi in particolare, valendo $m_j-p \leq j \leq m_j$, vale $c_j = 0$

Per arbitrarietà di j vale $c_0 = \dots = c_n = 0$

Quindi le $\{N_{i,p}(x)\}_{i=0, \dots, n}$ sono linearmente indipendenti e quindi base di V

B-spline curves

A B-spline curve of degree p is defined as

$$\mathcal{C}(u) = \sum_{i=0}^n N_{i,p}(u) P_i, \quad a \leq u \leq b$$

where

- ▶ P_0, \dots, P_n are the control points,
- ▶ $N_{0,p}(u), \dots, N_{n,p}(u)$ are the B-spline basis functions of degree p defined on the knot vector

$$U = \underbrace{\{a, \dots, a\}}_{p+1}, u_{p+1}, \dots, u_{m-p-1}, \underbrace{\{b, \dots, b\}}_{p+1}.$$

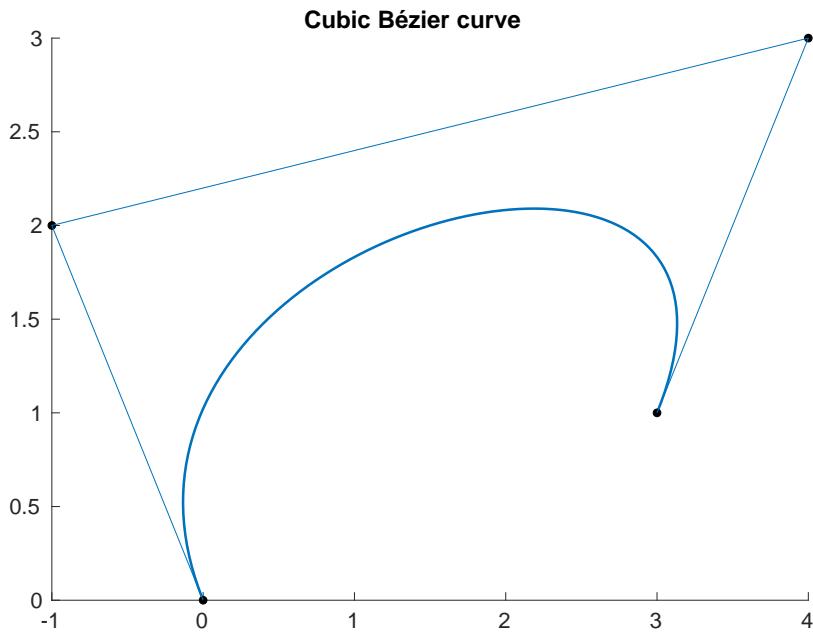
containing $m + 1$ knots with $m = n + p + 1$.

B-spline curves: properties

- ▶ If $n = p$ and $U = \{0, \dots, 0, 1, \dots, 1\}$, then $\mathcal{C}(u)$ is a Bézier curve.
- ▶ Endpoint interpolation: $\mathcal{C}(a) = P_0$ and $\mathcal{C}(b) = P_n$.
- ▶ Affine invariance: an affine transformation is applied to the curve by applying it to the control points.
- ▶ Strong convex hull property:
 - ▶ the curve is contained in the convex hull of its control polygon,
 - ▶ if $u \in [u_i, u_{i+1})$ with $p \leq i < m - p - 1$, then $\mathcal{C}(u)$ is in the convex hull of the control points P_{i-p}, \dots, P_i . "inviluppo convesso locale"
- ▶ Local modification: moving P_i changes $\mathcal{C}(u)$ only in the interval $[u_i, u_{i+p+1})$.
- ▶ Variation diminishing property: no line (plane) has more intersections with the curve than with the control polygon.

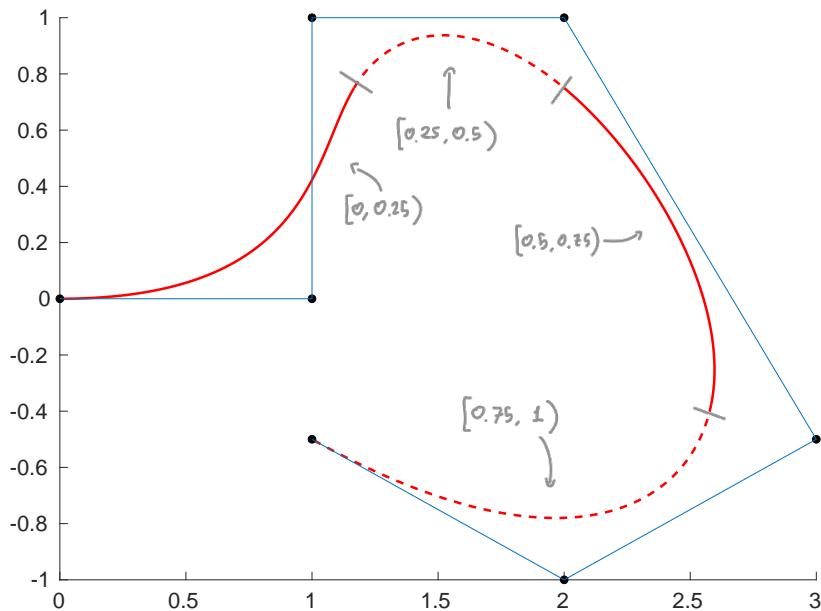
B-spline curves: examples

A cubic Bézier curve: knots $\{0, 0, 0, 0, 1, 1, 1, 1\}$.



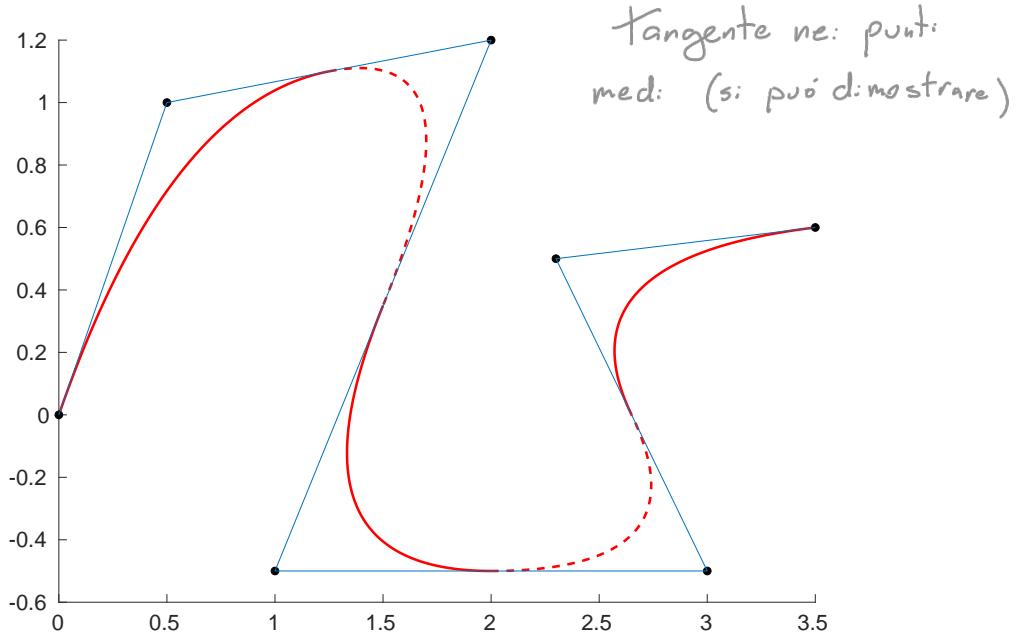
B-spline curves: examples

A cubic curve: knots $\{0, 0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1, 1\}$.



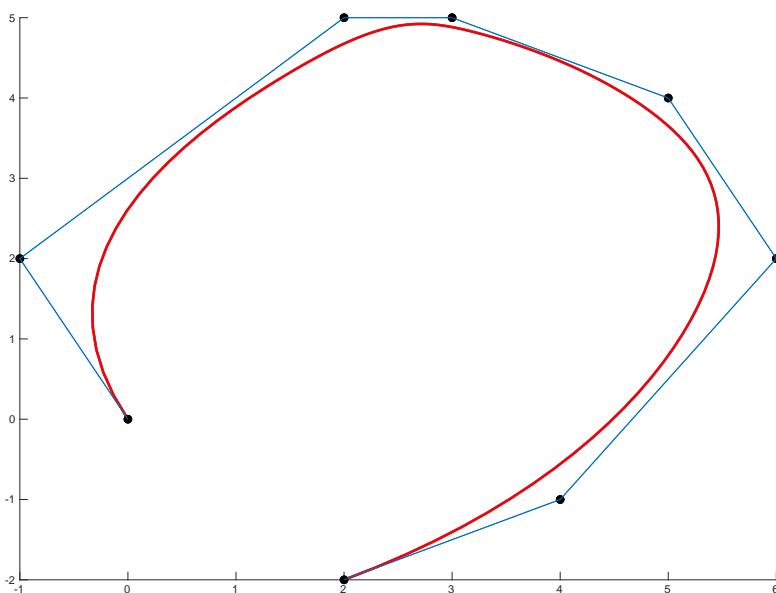
B-spline curves: examples

A quadratic curve: knots $\{0, 0, 0, 0.2, 0.4, 0.6, 0.8, 1, 1, 1\}$.



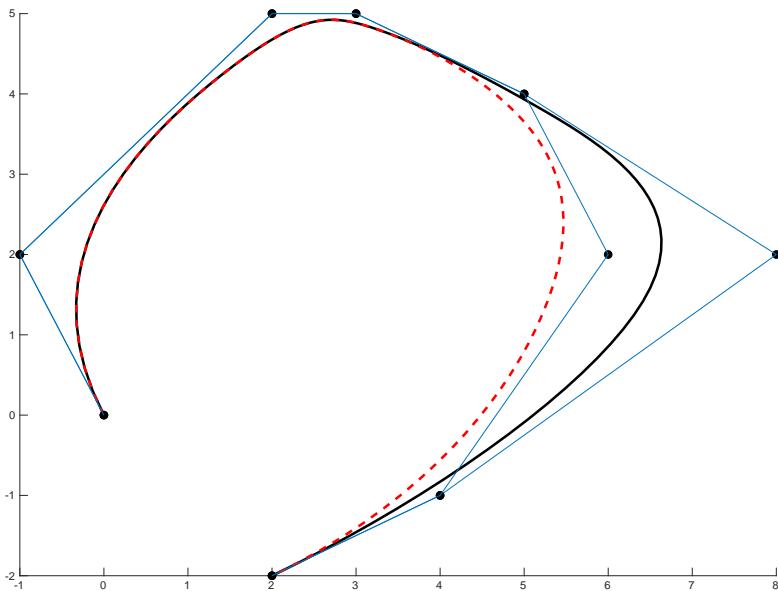
B-spline curves: examples

A cubic curve: knots $\{0, 0, 0, 0, 0.2, 0.4, 0.6, 0.8, 1, 1, 1, 1\}$.



B-spline curves: examples

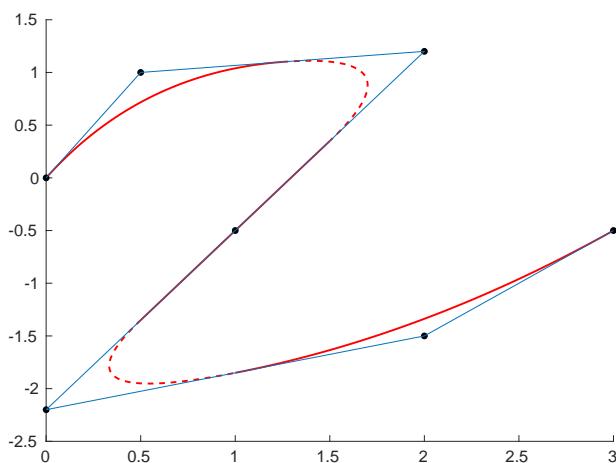
Same cubic curve after modifying one control point.



An exercise

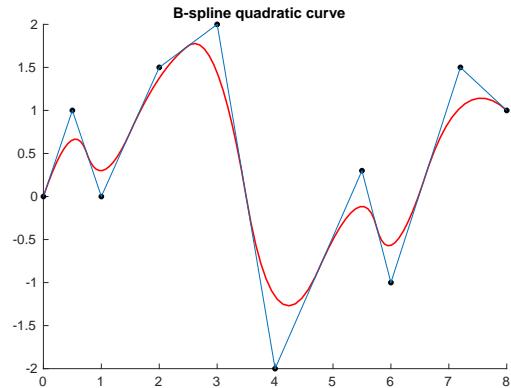
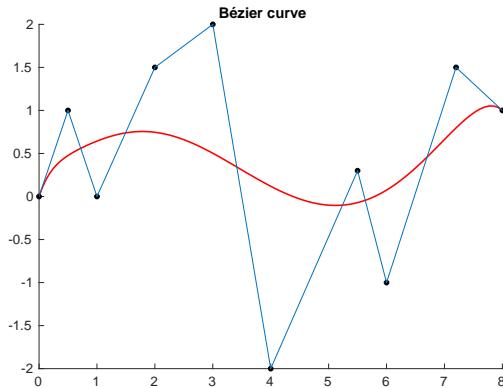
How can we construct a quadratic B-spline curve containing a straight line segment?

Idea: use the strong convex hull property and choose three consecutive control points to be collinear.



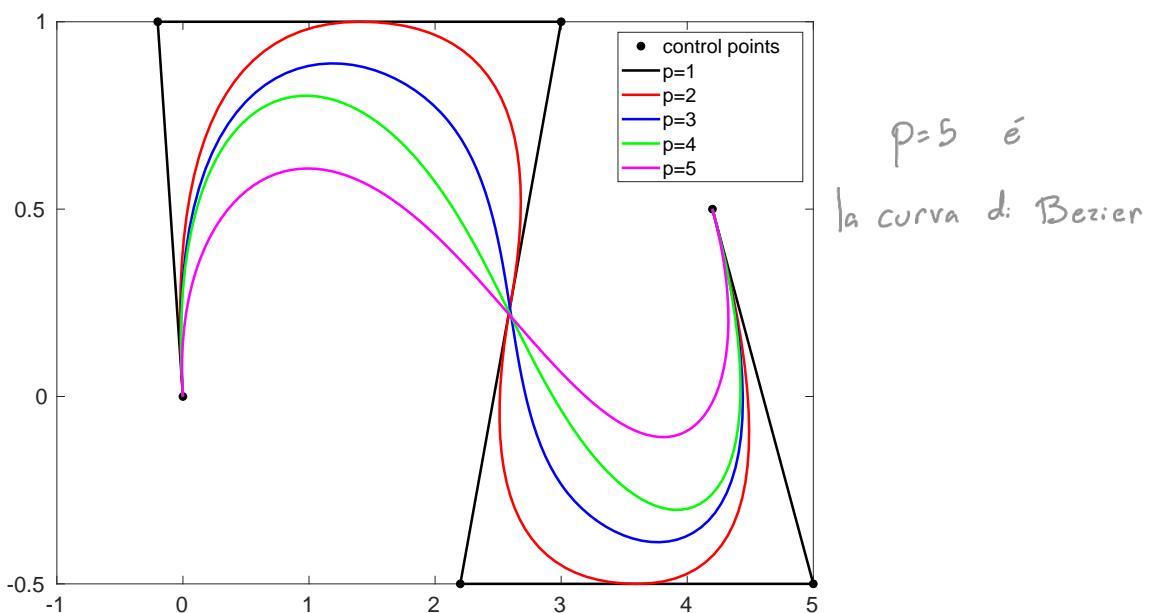
B-spline curves: more properties

In general: the lower the degree, the closer the curve follows the control polygon.



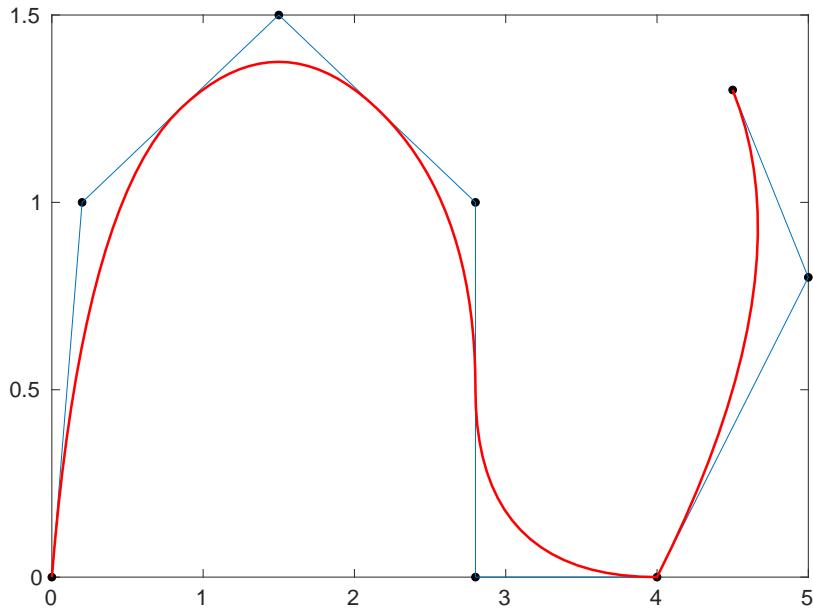
Comparison between two B-spline curves with the same control points: the Bézier curve and the quadratic curve with uniformly spaced interior knots.

B-spline curves: more properties



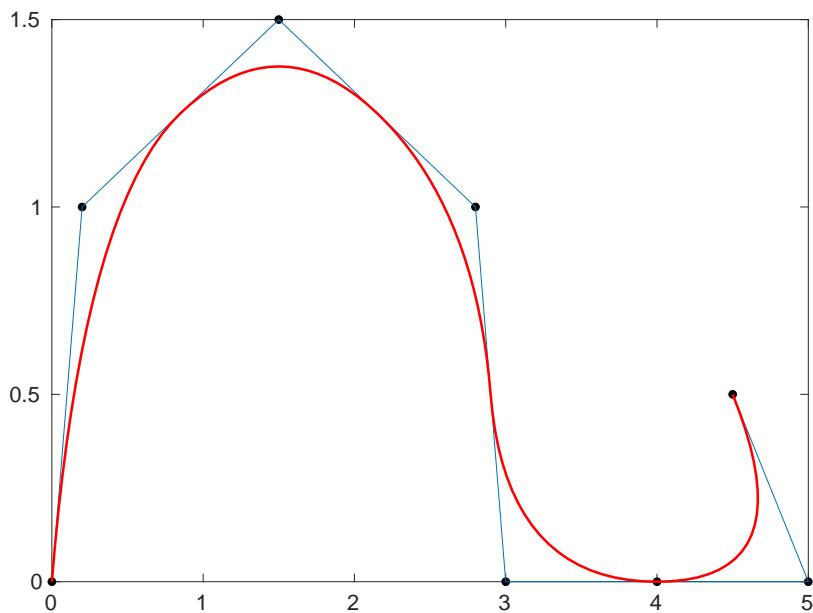
B-spline curves of different degrees with uniformly spaced interior knots and same control points.

B-spline curves: continuity



Quadratic curve, knots $\{0, 0, 0, \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}, \frac{4}{5}, 1, 1, 1\}$. The curve is C^0 but not C^1 at $u = \frac{4}{5}$.

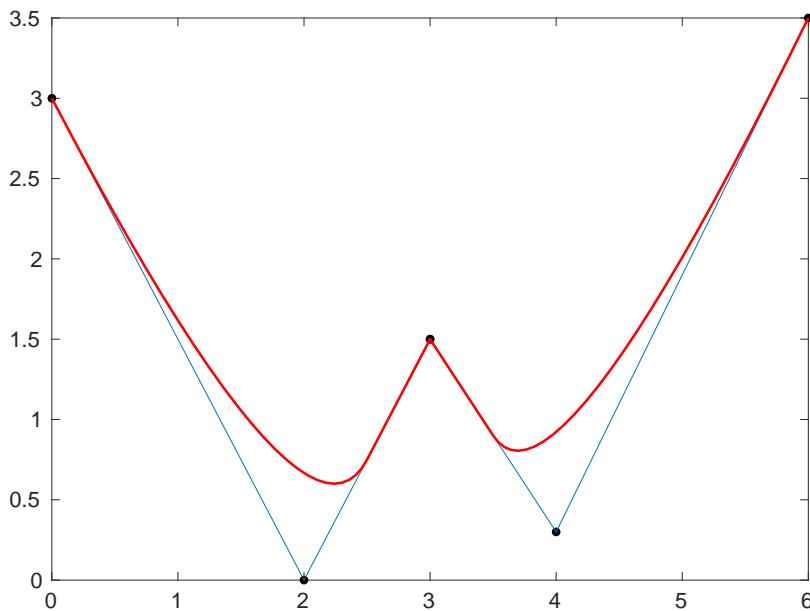
B-spline curves: continuity



Quadratic curve, knots $\{0, 0, 0, \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}, \frac{4}{5}, 1, 1, 1\}$. This curve happens to be C^1 also at $u = \frac{4}{5}$. solo perché i punti di controllo sono presi in modo speciale

B-spline curves: continuity

What happens if we choose multiple control points?



Quadratic curve, knots $\{0, 0, 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1, 1, 1\}$. $P_2 = P_3$ is a double control point. How can the curve be still C^1 ?

Knot insertion

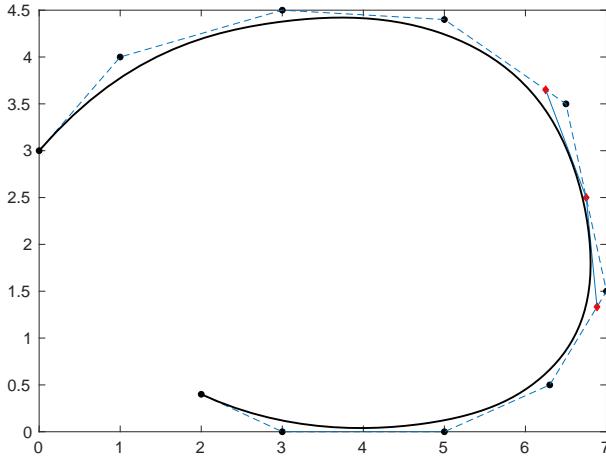
- ▶ Given a B-spline curve $\mathcal{C}(u)$ with control points P_0, \dots, P_n and knots $0, \dots, 0, u_{p+1}, \dots, u_n, 1, \dots, 1$, we want to insert a new knot \tilde{u} in the knot vector, without changing the shape of the curve.
- ▶ The number of control points will also increase by one (because $m = n + p + 1$).
- ▶ Suppose $\tilde{u} \in [u_k, u_{k+1}]$. The strong convex hull property implies that $\mathcal{C}(\tilde{u})$ lies in the convex hull of $P_k, P_{k-1}, \dots, P_{k-p}$.
- ▶ We will introduce new control points Q_{k-p+1}, \dots, Q_k , so the the polygonal line $P_{k-p} P_{k-p+1} \dots P_k$ is replaced by $P_{k-p} Q_{k-p+1} \dots Q_k P_k$.

Knot insertion

The formula for computing the new control points is

$$Q_i = (1 - \alpha_i)P_{i-1} + \alpha_i P_i$$

with $\alpha_i = \frac{\tilde{u} - u_i}{u_{i+p} - u_i}$ for $k - p + 1 \leq i \leq k$.



De Boor's algorithm

- ▶ Method for evaluating a B-spline curve $\mathcal{C}(u)$, given control points P_0, \dots, P_n and knots $0, \dots, 0, u_{p+1}, \dots, u_n, 1, \dots, 1$.
- ▶ Generalizes De Casteljau's algorithm.
- ▶ First, determine k such that $u \in [u_k, u_{k+1})$, e.g., by bisection.
- ▶ Then set $P_{i,0} = P_i$ for $i = k - p, \dots, k$...
- ▶ ...and set up a p -step algorithm: at the r -th step define

$$P_{i,r} = (1 - \alpha_{i,r})P_{i-1,r-1} + \alpha_{i,r} P_{i,r-1}$$

with $\alpha_{i,r} = \frac{u - u_i}{u_{i+p-r+1} - u_i}$ and $k - p + r \leq i \leq k$.

- ▶ After p steps we have $\mathcal{C}(u) = P_{k,p}$.

De Boor's algorithm: an example

Consider a cubic B-spline curve with uniformly spaced knots $\{0, 0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1, 1\}$ and 6 control points. We want to compute the point $\mathcal{C}(u)$ for $u = 0.4 \in [u_4, u_5]$.

$$P_1 = P_{1,0}$$

$$P_{2,1}$$

$$P_2 = P_{2,0}$$

$$P_{3,2}$$

$$P_3 = P_{3,0}$$

$$P_{4,2}$$

$$P_4 = P_{4,0}$$

$$P_{3,1}$$

$$P_{4,1}$$

$$\alpha_{4,1} = \frac{u - u_4}{u_{4+3} - u_4} = 0.2$$

$$\alpha_{3,1} = \frac{u - u_3}{u_{3+3} - u_3} = \frac{8}{15} \approx 0.53$$

$$\alpha_{2,1} = \frac{u - u_2}{u_{2+3} - u_2} = 0.8$$

$$P_{4,1} = (1 - \alpha_{4,1})P_{3,0} + \alpha_{4,1}P_{4,0}$$

$$P_{3,1} = (1 - \alpha_{3,1})P_{2,0} + \alpha_{3,1}P_{3,0}$$

$$P_{2,1} = (1 - \alpha_{2,1})P_{1,0} + \alpha_{2,1}P_{2,0}$$

De Boor's algorithm: an example

Consider a cubic B-spline curve with uniformly spaced knots $\{0, 0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1, 1\}$ and 6 control points. We want to compute the point $\mathcal{C}(u)$ for $u = 0.4 \in [u_4, u_5]$.

$$P_1 = P_{1,0}$$

$$P_{2,1}$$

$$P_2 = P_{2,0}$$

$$P_{3,2}$$

$$P_3 = P_{3,0}$$

$$P_{4,2}$$

$$P_4 = P_{4,0}$$

$$P_{3,1}$$

$$P_{4,1}$$

$$\alpha_{4,2} = \frac{u - u_4}{u_{4+3-1} - u_4} = 0.3$$

$$\alpha_{3,2} = \frac{u - u_3}{u_{3+3-1} - u_3} = 0.8$$

$$P_{4,2} = (1 - \alpha_{4,2})P_{3,1} + \alpha_{4,2}P_{4,1}$$

$$P_{3,2} = (1 - \alpha_{3,2})P_{2,1} + \alpha_{3,2}P_{3,1}$$

De Boor's algorithm: an example

Consider a cubic B-spline curve with uniformly spaced knots $\{0, 0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1, 1\}$ and 6 control points. We want to compute the point $C(u)$ for $u = 0.4 \in [u_4, u_5]$.

$$P_1 = P_{1,0}$$

$$P_{2,1}$$

$$P_2 = P_{2,0}$$

$$P_{3,2}$$

$$\alpha_{4,3} = \frac{u - u_4}{u_{4+3-2} - u_4} = 0.6$$

$$P_{3,1}$$

$$P_{4,3}$$

$$P_3 = P_{3,0}$$

$$P_{4,2}$$

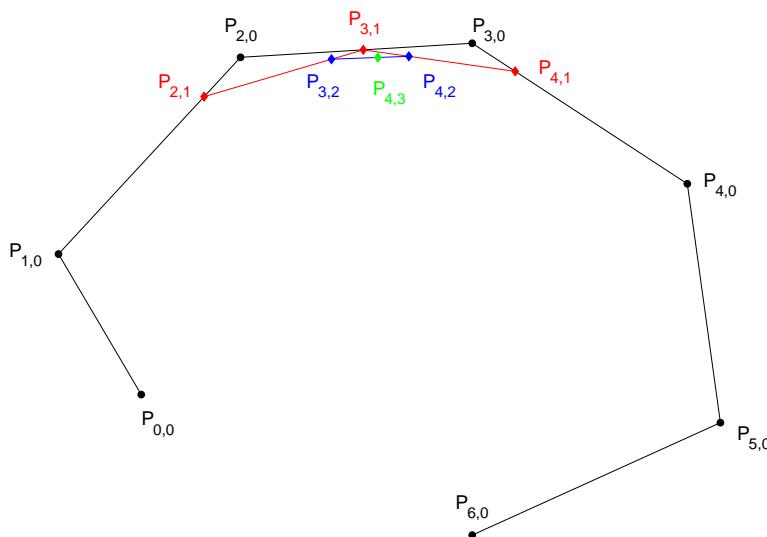
$$P_{4,3} = (1 - \alpha_{4,3})P_{3,2} + \alpha_{4,3}P_{4,2}$$

$$P_{4,1}$$

$$P_4 = P_{4,0}$$

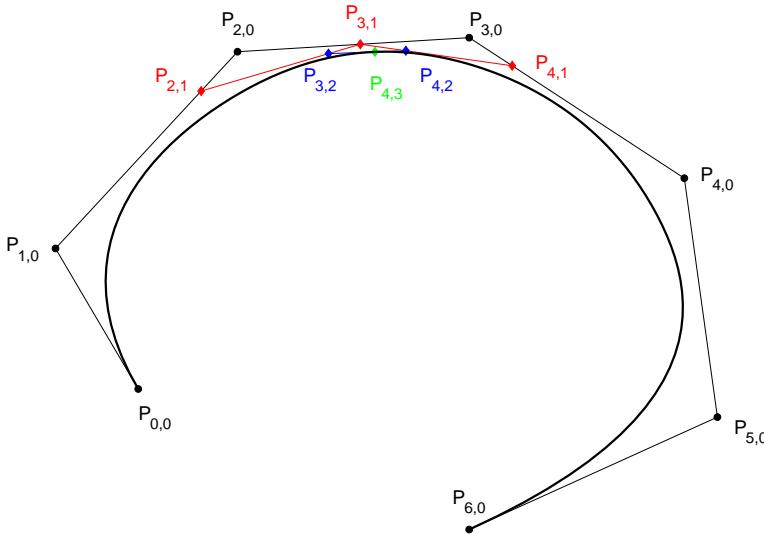
De Boor's algorithm: an example

Consider a cubic B-spline curve with uniformly spaced knots $\{0, 0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1, 1\}$ and 6 control points. We want to compute the point $C(u)$ for $u = 0.4 \in [u_4, u_5]$.



De Boor's algorithm: an example

Consider a cubic B-spline curve with uniformly spaced knots $\{0, 0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1, 1\}$ and 6 control points. We want to compute the point $\mathcal{C}(u)$ for $u = 0.4 \in [u_4, u_5]$.



De Boor's algorithm: a geometric viewpoint

- ▶ The following **property** of B-spline curves holds: if a knot has multiplicity k , then there are at most $p - k + 1$ basis functions that are nonzero at this knot.
- ▶ So if a knot u_i has multiplicity p , there is only one basis function that is nonzero at u_i , and its value is 1.
- ▶ Now, suppose we want to compute $\mathcal{C}(u)$ at \tilde{u} . Let us repeatedly **insert \tilde{u} as a new knot p times**. Then the last newly inserted control point will be $\mathcal{C}(\tilde{u})$.

Derivatives of B-spline curves

Consider a B-spline curve

$$\mathcal{C}(u) = \sum_{i=0}^n N_{i,p}(u) P_i$$

defined on the knot vector

$$U = \underbrace{\{a, \dots, a\}}_{p+1}, u_{p+1}, \dots, u_{m-p-1}, \underbrace{\{b, \dots, b\}}_{p+1}.$$

By using the formula for derivatives of B-spline basis functions and rearranging the resulting sum, we obtain

$$\mathcal{C}'(u) = \sum_{i=0}^{n-1} N_{i+1,p-1}(u) Q_i,$$

where

$$Q_i = p \frac{P_{i+1} - P_i}{u_{i+p+1} - u_{i+1}}.$$

Derivatives of B-spline curves

Let U' be the knot vector

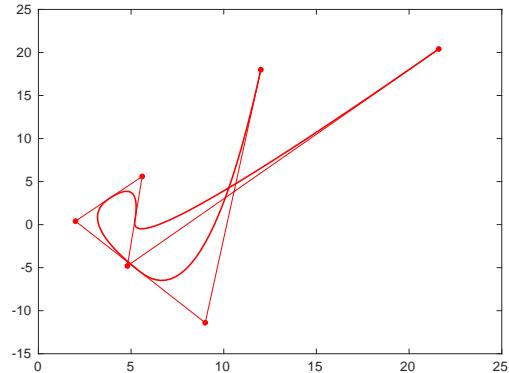
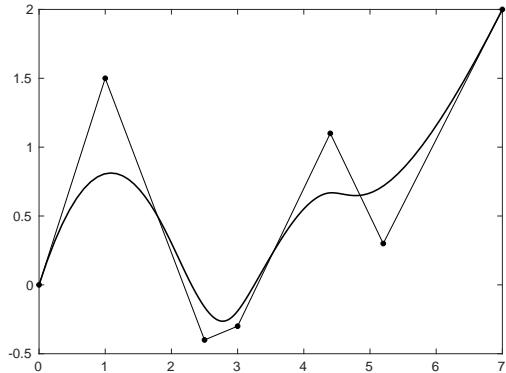
$$U' = \underbrace{\{a, \dots, a\}}_p, u_{p+1}, \dots, u_{m-p-1}, \underbrace{\{b, \dots, b\}}_p.$$

It is easy to see that $N_{i+1,p-1}(u)$ computed on U is the same as $N_{i,p-1}(u)$ computed on U' . Therefore $\mathcal{C}'(u)$ is a B-spline curve of degree $p-1$ defined on the knot vector U' :

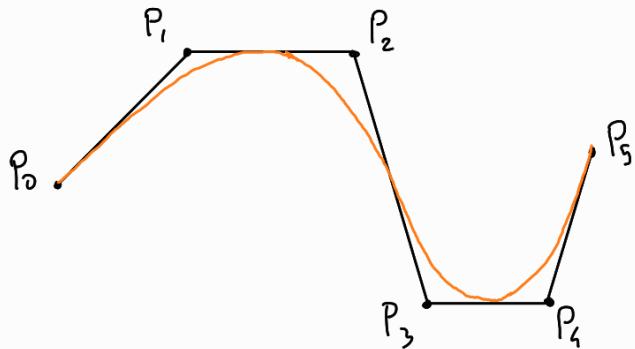
$$\mathcal{C}'(u) = \sum_{i=0}^{n-1} N_{i,p-1}(u) Q_i.$$

Derivatives of B-spline curves

An example of cubic B-spline curve and its derivative.



Ex Data una curva Bspline quadratica con nodi interni distinti: $U_0 = U_1 = U_2 < U_3 < \dots < U_n < U_{n+1} = U_{n+2} = U_{n+3}$



mostrare che

- la curva è tangente al poligono di controllo in P_0 e P_n
 - la curva è tangente ai segmenti: $\overline{P_1P_2}, \overline{P_2P_3}, \dots, \overline{P_{n-2}P_{n-1}}$
- Inoltre se i nodi sono equispaziati la tangenza avviene nei punti medi:

$$\underline{\text{Sol}} \quad \text{Sia } C(u) = \sum_{i=0}^n N_{i,p}(u) P_i \quad \text{con } p=2$$

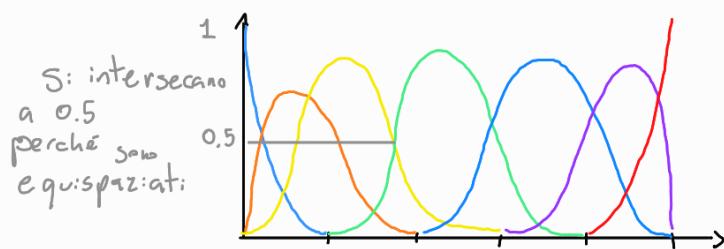
Sappiamo che per dominio delle $N_{i,p}$, considerando $u \in [U_j, U_{j+1})$

$$\text{vale } C(u) = \sum_{i=j-2}^j N_{i,2}(u) P_i$$

Consideriamo $u = U_j$ per un certo j

Es Siano i nodi $0, 0, 0, 0.2, 0.4, 0.6, 0.8, 1, 1, 1$

le Bspline allora sono



ovvero, in un punto generico 3 Bspline sono non nulle, ma nei nodi esattamente 2 Bspline sono non nulle

Poiché sugli U_j solo 2 Bspline sono non nulle possiamo

restringere ulteriormente la sommatoria

$$C(u_j) = N_{j-2,2}(u_j)P_{j-2} + N_{j-1,2}(u_j)P_{j-1}$$

Quindi: $C(u_j)$ si trova sul segmento $\overline{P_{j-2} P_{j-1}}$

Se i nodi sono equispaziati, $N_{j-2,2}(u_j) = N_{j-1,2}(u_j) = \frac{1}{2}$
(si mostra graficamente in modo facile o co: cont: se ci fieni particolarmente) quindi: $C(u_j)$ è il punto medio del segmento

Per la tangenza uso la formula per la derivata

$$C'(u) = \sum_{i=0}^{n-1} N_{i+1,p-1}(u) Q_i \quad \text{dove } Q_i = P \frac{P_{i+1} - P_i}{U_{i+p+1} - U_i}$$

Nel nostro caso sempre prendendo $u=u_j$ vale, per dominio su $[U_j, U_{j+1})$

$$C'(u_j) = \sum_{i=j-1}^j N_{i+1,1}(u_j) Q_i$$

Ma in realtà poiché stiamo prendendo proprio u_j e non un generico $u \in [U_j, U_{j+1})$ è non nulla solo la Bspline $N_{j-1,1}$ che, per partizione dell'unità, è uguale a 1, quindi:

$$C'(u_j) = N_{j-1,1}(u_j) Q_{j-2}$$

oss il motivo per cui è $N_{j-1,1}$ e non N_j è perché stiamo considerando un vettore d: nodi per Bspline d: grado 2 mentre

queste sono Bspline d: grado 1, ovvero c'è un nodo in più quindi c'è da aggiustare la formula che conosciamo e venne chiamata gli indici.

D

Algoritmo d: de Boor

È possibile generalizzare l'algoritmo d: de Casteljau per calcolare curve Bspline. L'algoritmo ottenuto è l'algoritmo d: de Boor

Osserviamo che, utilizzando la formula di ricorrenza delle $N_{i,p}$

$$\begin{aligned} \mathcal{C}(u) &= \sum_{i=0}^n N_{i,p}(u) P_i = \\ &= \sum_{i=0}^n P_i \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \sum_{i=0}^n P_i \frac{u_{i+p-1} - u}{u_{i+p-1} - u_{i+1}} N_{i+1,p-1}(u) = \end{aligned}$$

Poiché per $i=0$ il termine della prima sommatoria si annulla posso partire da $i=1$

Poiché per $i=n$ il termine della seconda sommatoria si annulla, posso fermarmi al $i=n-1$ e poi fare uno shift dell'indice

$$\begin{aligned} &= \sum_{i=1}^n (\quad \quad \quad) + \sum_{i=1}^n \frac{u_{i+p} - u}{u_{i+p} - u_i} N_{i,p-1}(u) P_{i-1} = \\ &= \sum_{i=1}^n N_{i,p}(u) \left[\frac{u - u_i}{u_{i+p} - u_i} P_i + \frac{u_{i+p} - u}{u_{i+p} - u_i} P_{i-1} \right] \\ &= \sum_{i=1}^n N_{i,p}(u) P_i^{(1)} \quad \quad \quad =: P_i^{(1)} \end{aligned}$$

Iterando, preso $u \in [u_k, u_{k+1})$ vale $\mathcal{C}(u) = P_k^{(p)}$ dove

$$P_i^{(j)}(u) = \begin{cases} (1 - \alpha_i^{(j)}) P_{i-1}^{(j-1)} + \alpha_i^{(j)} P_i^{(j-1)} & \text{se } j > 0 \\ P_i & \text{se } j = 0 \end{cases}$$

$$\alpha_i^{(j)} = \frac{u - u_i}{u_{i+p-j+1} - u_i}$$

Curve di Bézier razionali:

Siano $P_0, \dots, P_n \in \mathbb{R}^d$ punti di controllo e $w_0, \dots, w_n > 0$ pesi.

La curva di Bézier razionale associata è

$$\mathcal{C}(t) = \frac{\sum_{i=0}^n \boxed{B_i^{(n)}(t) w_i P_i}}{\boxed{\sum_{j=0}^n w_j B_j^{(n)}(t)}} = R_i^{(n)}(t) P_i$$

Proprietà degli $R_i^{(n)}$

- Non-negatività: $R_i^{(n)}(t) \geq 0 \quad \forall i, n \quad \forall t \in [0, 1]$

segue direttamente dalla definizione e dalla non negatività delle $B_j^{(n)}$

- Partizione dell'unità:

$$\begin{aligned} \sum_{i=0}^n R_i^{(n)}(t) &= \sum_{i=0}^n \frac{B_i^{(n)}(t) w_i}{\sum_{j=0}^n B_j^{(n)}(t) w_j} = \\ &= \frac{\sum_{i=0}^n B_i^{(n)}(t) w_i}{\sum_{j=0}^n B_j^{(n)}(t) w_j} = \\ &= 1 \end{aligned}$$

- $R_0^{(n)}(0) = 1, \quad i \neq 0 \Rightarrow R_i^{(n)}(0) = 0$

- $R_n^{(n)}(1) = 1, \quad i \neq n \Rightarrow R_i^{(n)}(1) = 0$

- $R_i^{(n)}(t)$ ha esattamente un punto di massimo in $[0, 1]$

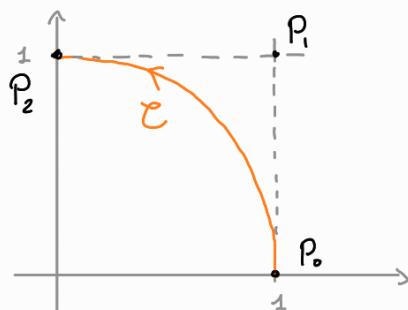
- Se $w_0 = \dots = w_n \quad R_i^{(n)}(t) = B_i^{(n)}(t) \quad \forall i$

Proprietà delle curve di Bézier razionali:

- involucro convesso
 - invarianza affine
 - variation diminishing
 - interpolazione agli estremi, cioè $C(0) = P_0$ e $C(1) = P_n$
 - le curve di Bézier "classiche" sono un caso particolare di quelle razionali
 - le derivate K-esime agli estremi dei primi (ultimi) $K+1$ punti di controllo ($K=1 \Rightarrow$ tangenza agli estremi)
- conseguenze della partizione dell'unità

Es Scriviamo un arco di circonferenza come curva di:

Bézier razionale di grado 2



dobbiamo scegliere 3 punti
di controllo. Per interpolazione

agli estremi deve valere

$$P_0 = (1,0) \text{ e } P_2 = (0,1)$$

Per P_1 osserviamo che per tangenza, deve giacere sull'intersezione delle tangenti agli estremi quindi: $P_1 = (1,1)$

Per la scelta dei pesi, cerchiamo di produrre la parametrizzazione classica, ovvero

$$x(t) = \frac{1-t^2}{1+t^2}, \quad y(t) = \frac{2t}{1+t^2}$$

ovvero scegliamo w_0, w_1, w_2 t.c. $\sum_{i=0}^2 B_i^{(n)}(t) w_i = 1+t^2$

Esplcitando otteniamo

$$(1-t)^2 w_0 + 2t(1-t) w_1 + t^2 w_2 = 1+t^2$$

$$w_0 + t(2w_1 - 2w_0) + t^2(w_0 - 2w_1 + w_2) = 1+t^2$$

$$\begin{cases} w_0 = 1 \\ 2w_1 - 2w_0 = 0 \\ w_0 - 2w_1 + w_2 = 1 \end{cases} \Rightarrow \begin{cases} w_0 = 1 \\ w_1 = 1 \\ w_2 = 2 \end{cases}$$

Questa scelta dei pesi fa sì che i denominatori della nostra parametrizzazione siano gli stessi di quella "classica"

Vale anche che

$$(1-t)^2 \cdot 1 \cdot \binom{1}{0} + 2t(1-t) \cdot 1 \cdot \binom{1}{1} + t^2 \cdot 2 \cdot \binom{0}{1} = \binom{1-t^2}{2t}$$

quindi anche i numeratori coincidono quindi è la stessa parametrizzazione

Vediamo ora un'interpretazione proiettiva

Fixiamo $d=3$. Dato $(x, y, z) \in \mathbb{R}^3$ e $w \in \mathbb{R} \setminus \{0\}$ posso omogeneizzare $(x, y, z) \rightsquigarrow (wx, wy, wz, w) \in \mathbb{R}^4 \setminus \{0\}$

Viceversa dato $(X, Y, Z, W) \in \mathbb{R}^4 \setminus \{W=0\}$ posso proiettarlo

$$(X, Y, Z, W) \rightsquigarrow \left(\frac{X}{W}, \frac{Y}{W}, \frac{Z}{W} \right) \in \mathbb{R}^3$$

Sia $C(t) = \sum_{i=0}^n R_i^{(n)}(t) P_i$, $P_i \in \mathbb{R}^3$

Per ciascun $P_i = (x_i, y_i, z_i)$ definisco $P_i^w = (x_i w_i, y_i w_i, z_i w_i, w_i)$ ovvero l'omogeneizzato di P_i rispetto al peso i -esimo w_i .

Considero $C^w(t) = \sum_{i=0}^n B_i^{(n)}(t) P_i^w$. Esplorando e proiettando si ottiene

$$C^w(t) = \begin{pmatrix} \sum_{i=0}^n B_i^{(n)}(t) x_i w_i \\ \sum_{i=0}^n B_i^{(n)}(t) y_i w_i \\ \sum_{i=0}^n B_i^{(n)}(t) z_i w_i \\ \sum_{i=0}^n B_i^{(n)}(t) w_i \end{pmatrix} \rightsquigarrow \begin{pmatrix} \frac{\sum_{i=0}^n B_i^{(n)}(t) x_i w_i}{\sum_{i=0}^n B_i^{(n)}(t) w_i} \\ \vdots \\ \frac{\sum_{i=0}^n B_i^{(n)}(t) z_i w_i}{\sum_{i=0}^n B_i^{(n)}(t) w_i} \\ \frac{\sum_{i=0}^n B_i^{(n)}(t) w_i}{\sum_{i=0}^n B_i^{(n)}(t) w_i} \end{pmatrix} = C(t)$$

Quindi una curva d. Bézier razionale è la proiezione di una curva d. Bézier classica di dimensione superiore

4/11

Consideriamo le curve d. Bézier razionali di grado 2

Siano P_0, P_1 e P_2 punti di controllo e w_0, w_1, w_2 pesi

$$C(u) = \frac{w_0 P_0 (1-u)^2 + 2u(1-u)w_1 P_1 + u^2 w_2 P_2}{w_0 (1-u)^2 + 2u(1-u)w_1 + u^2 w_2}$$

Una curva così prodotta è un arco di conica. Infatti, mostriamo

che $\mathcal{E}(u)$ soddisfa l'equazione d'una conica nel sistema d:

riferimento $(P_1, P_0 - P_1, P_2 - P_1)$:

Per proprietà dell'inviluppo convesso, $\forall u \in [0, 1] \exists \alpha(u), \beta(u) > 0$ t.c.

$$\mathcal{E}(u) = P_1 + \alpha(u)(P_0 - P_1) + \beta(u)(P_2 - P_1)$$

$$= \alpha(u)P_0 + (1 - \alpha(u) - \beta(u))P_1 + \beta(u)P_2$$

da cui si ricava $\alpha(u) = R_0^{(2)}(u)$, $\beta(u) = R_2^{(2)}(u)$, $(1 - \alpha(u) - \beta(u)) = R_1^{(2)}(u)$

Osserviamo che $R_0^{(2)}(u) \cdot R_2^{(2)}(u) = \frac{u^2(1-u)^2 w_0 w_2}{d(u)^2}$ (dove $d(u) = \sum R_i^{(2)}(u) w_i$)

$$R_1^{(2)}(u)^2 = \frac{4u^2(1-u)^2 w_1^2}{d(u)^2}$$

Cioè $4 \frac{w_1^2}{w_0 w_2} R_0^{(2)}(u) \cdot R_2^{(2)}(u) = R_1^{(2)}(u)^2$ e sostituendo ottieniamo

$$4 \frac{w_1^2}{w_0 w_2} \alpha(u) \beta(u) = (1 - \alpha(u) - \beta(u))^2$$

quindi α e β rispettano le relazioni d'una conica (a meno d. sviluppare) e sono quindi parametrizzazioni delle coordinate d'una conica.

Studiamo che tipo d'conica è studiandone i punti all'infinito, ponendo il denominatore $d(u) = 0$

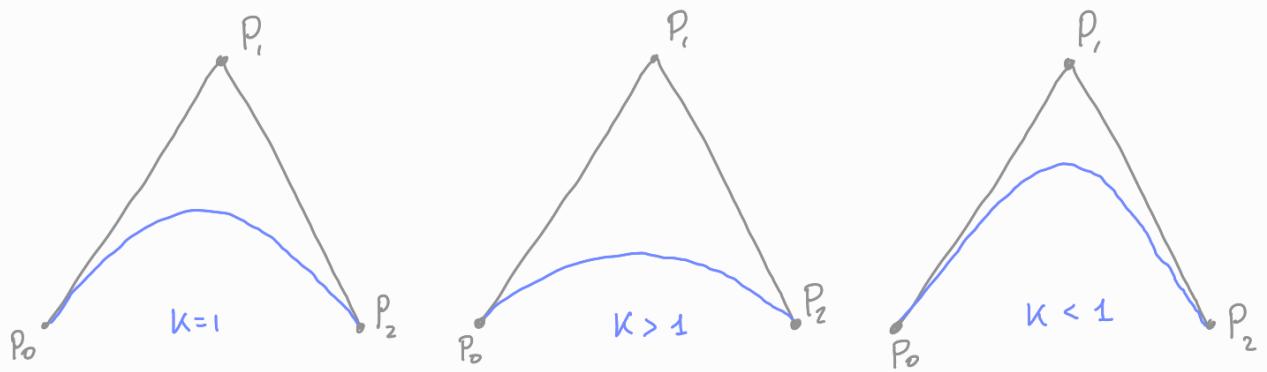
$$u^2(w_0 - 2w_1 + w_2) + 2u(w_1 - w_0) + w_0 = 0$$

$$\frac{\Delta}{4} = (w_1 - w_0)^2 - w_0(w_0 - 2w_1 + w_2)$$

$$= w_1^2 + w_0^2 - 2w_0 w_1 - w_0^2 + 2w_0 w_1 - w_0 w_2 =$$

$$= w_1^2 - w_0 w_2$$

Sia $\kappa = \frac{w_0 w_2}{w_1^2}$, allora per $\kappa = 1$ abbiamo una parabola, per $\kappa > 1$ abbiamo un'ellisse e per $\kappa < 1$ abbiamo un'iperbole



Vedremo che a meno d'riparametrizzare possiamo sempre

supporre $\omega_0 = \omega_2 = 1$, quindi: $K = \frac{1}{\omega_1^2}$, ovvero

$\omega_1 = 1 \Rightarrow$ parabola, $\omega_1 < 1 \Rightarrow$ ellisse $\omega_1 > 1 \Rightarrow$ iperbole

Il caso limite (non ammesso) $\omega_1 = 0$ risulta nel segmento P_0P_2

In pratica si usa spesso lo **SHOULDER POINT** $S = C(\frac{1}{2})$

Vale $S = C(\frac{1}{2}) = \frac{\frac{1}{4}\omega_0 P_0 + \frac{1}{2}\omega_1 P_1 + \frac{1}{4}\omega_2 P_2}{\frac{1}{4}\omega_0 + \frac{1}{2}\omega_1 + \frac{1}{4}\omega_2}$

Se WLOG prende $\omega_0 = \omega_2 = 1$ ottieniamo

$$S = \frac{P_0 + 2\omega_1 P_1 + P_2}{2(1+\omega_1)} = \underbrace{\frac{1-s}{1+\omega_1} \frac{P_0 + P_2}{2} + \frac{s}{1+\omega_1} P_1}_{\substack{\text{punto medio} \\ \text{del segmento } P_0P_2, \text{ chiamiamolo } M}}$$

$$\Rightarrow S = (1-s)M + sP_1$$

Allora $s=0 \Rightarrow$ retta, $s=\frac{1}{2} \Rightarrow$ parabola
 $s \in (0, \frac{1}{2}) \Rightarrow$ ellisse, $s \in (\frac{1}{2}, 1) \Rightarrow$ iperbole

Oss Proprietà della precisione lineare

- Per curve d'Bezier polinomiali ^{"classiche" aka non razionali:}
 se P_0, \dots, P_n sono allineati e equispaziati, allora la curva associata è la parametrizzazione lineare del segmento P_0P_n

Infatti, detto $P_0 = P$, $P_n = Q$, per hp sappiamo che

$$P_j = (1 - \frac{j}{n})P + \frac{j}{n}Q$$

$$\begin{aligned} \text{Allora } C(t) &= \sum_{j=0}^n P_j B_j^{(n)}(t) = \sum_{j=0}^n \left[(1 - \frac{j}{n})P + \frac{j}{n}Q \right] B_j^{(n)}(t) = \\ &= \left[\sum_{j=0}^n (1 - \frac{j}{n}) B_j^{(n)}(t) \right] P + \left[\sum_{j=0}^n \frac{j}{n} B_j^{(n)}(t) \right] Q = \\ &= (1-t)P + tQ \end{aligned}$$

- Per curve d: Bézier razionali:

per ogni scelta d: $P_j = (1 - f(t))P + f(t)Q$, $0 \leq f(t) \leq 1$

$$\exists w_0, \dots, w_n \text{ tc } C(t) = (1-t)P + tQ$$

Algoritmo di de Casteljau razionale

Sia $C(t) = \sum_{j=0}^n P_j R_j^{(n)}(t)$ curva d: Bézier razionale, $t \in [0, 1]$.

Vediamo due metodi per calcolare $C(t)$ per un certo t fissato:

- (1) Considero la curva omogeneizzata $C^w(t) = \sum_{j=0}^n P_j^w B_j^{(n)}(t)$ e calcolo il punto omogeneizzato con de Casteljau per curve d: Bézier polinomiali e poi proietto il punto ottenuto

- (2) Implicitamente omogenizzo e proietto ad ogni passo

$$P_j^{(i+1)} = \frac{(1-t)w_j^{(i)}P_j^{(i)} + t w_{j+1}^{(i)}P_{j+1}^{(i)}}{(1-t)w_j^{(i)} + t w_{j+1}^{(i)}}$$

$$w_j^{(i+1)} = (1-t)w_j^{(i)} + t w_{j+1}^{(i)}$$

I pesi, che nell'omogeneizzato erano la 4a coordinata, si calcolano con de Casteljau classico (perché poi proiettando loro non cambiano)

i punti invece vengono calcolati con de Casteljau classico sulla curva omogeneizzata e poi proiettati (quindi si divide per il peso)

Supponiamo di avere $C(t)$, $t \in [a, b]$ curva di Bézier razionale. Vogliamo riparametrizzarla con $\tilde{f}: [c, d] \rightarrow [a, b]$ continua strettamente crescente bivaluosa (con eventualmente la regolarità richiesta)

Possiamo quindi studiare $\tilde{C}(s) = C(\tilde{f}(s))$.

Alcune possibili applicazioni sono:

(1) Siano A_1, \dots, A_n punti di C

posso chiedere che la curva passi per A_j per un pre determinato valore s_j del parametro

Più precisamente, se $C(t_j) = A_j$ posso scegliere

$\tilde{f}: [0, 1] \rightarrow [0, 1]$ t.c. $\tilde{f}(s_j) = t_j$ e riparametrizzando

ottengo $\tilde{C}(s_j) = C(t_j) = A_j$

Questo tipo di riparametrizzazione è noto come internal point mapping

(2) Possiamo modificare il modulo della derivata lungo la curva, ad esempio agli estremi per eventuali raccordi, infatti se

$\tilde{C}(s) = C(\tilde{f}(s))$ allora

$$\frac{d}{ds} \tilde{C}(s) = \tilde{f}'(s) C'(\tilde{f}(s))$$

(3) Possiamo portare la curva in "forma standard" modificando i pesi. Usiamo funzioni del tipo $\tilde{f}(s) = \frac{\alpha s + \beta}{\gamma s + \delta}$ poiché assicurano

che $\tilde{C}(s)$ sia razionale e dello stesso grado d: $C(t)$

Possiamo decidere d: fissare $s=1$ poiché f non cambia a meno di un fattore al numeratore e al denominatore.

Chiedendo $f(0)=0$ e $f(1)=1$ otteniamo $\beta=0$ e $\gamma=\alpha-1$
quindi consideriamo $f(s) = \frac{\alpha s}{(\alpha-1)s+1}$

$$\text{Allora se } C(t) = \frac{w_0(1-t)^n P_0 + nt(1-t)^{n-1} w_1 P_1 + \dots + w_n t^n P_n}{w_0(1-t)^n + nt(1-t)^{n-1} w_1 + \dots + w_n t^n},$$

$$\text{sostituendo } t = \frac{\alpha s}{(\alpha-1)s+1} \quad (\text{e quindi } 1-t = \frac{1-s}{(\alpha-1)s+1})$$

Poiché tutti i monomi al numeratore e tutti i monomi al denominatore avranno un denominatore $[(\alpha-1)s+1]^n$ posso già implicitamente semplificare, e quello che rimane è

$$\tilde{C}(s) = \frac{w_0(1-s)^n P_0 + n\alpha s(1-s)^{n-1} w_1 P_1 + \dots + w_n \alpha^n s^n P_n}{w_0(1-s)^n + n w_1 \alpha s(1-s)^{n-1} + \dots + w_n \alpha^n s^n}$$

Che è una curva con gli stessi punti di controllo di quella originale ma con pesi aggiornati: $\tilde{w}_i = \alpha^i w_i$.

In particolare se chiedo d: avere $\tilde{w}_0 = \tilde{w}_n$, poiché $\tilde{w}_0 = w_0$ e $\tilde{w}_n = \alpha^n w_n$ basta prendere $\alpha = \sqrt[n]{\frac{w_0}{w_n}}$

Quindi a meno di riparametrizzare posso supporre $w_0 = w_n$ e poiché i pesi sono determinati a meno d: un fattore globale posso sempre supporre WLOG $w_0 = w_n = 1$

Una curva in questa forma è detta in forma standard

NURBS

NURBS=Non-Uniform Rational B-Splines.

NURBS are the standard curve and surface representation method in most commercial CAD software.

Let us start by taking a look at rational B-spline curves.

Rational B-spline curves

A rational B-spline curve of degree p in \mathbb{R}^d is represented as

$$\mathcal{C}(u) = \frac{\sum_{i=0}^n N_{i,p}(u) w_i P_i}{\sum_{i=0}^n N_{i,p}(u) w_i}, \quad u \in [a, b],$$

where

- ▶ $P_0, \dots, P_n \in \mathbb{R}^d$ are the **control points**,
- ▶ $N_{0,p}(u), \dots, N_{n,p}(u)$ are the **B-spline basis functions** of degree p defined on the knot vector

$$U = \underbrace{\{a, \dots, a\}}_{p+1}, \underbrace{u_{p+1}, \dots, u_{m-p-1}\}}_{p+1}, \underbrace{b, \dots, b\}}_{p+1}.$$

containing $m + 1$ knots with $m = n + p + 1$,

- ▶ $w_0, \dots, w_n > 0$ are the **weights**.

Rational B-spline curves

We may equivalently write

$$\mathcal{C}(u) = \sum_{i=0}^n R_{i,p}(u) P_i, \quad u \in [a, b],$$

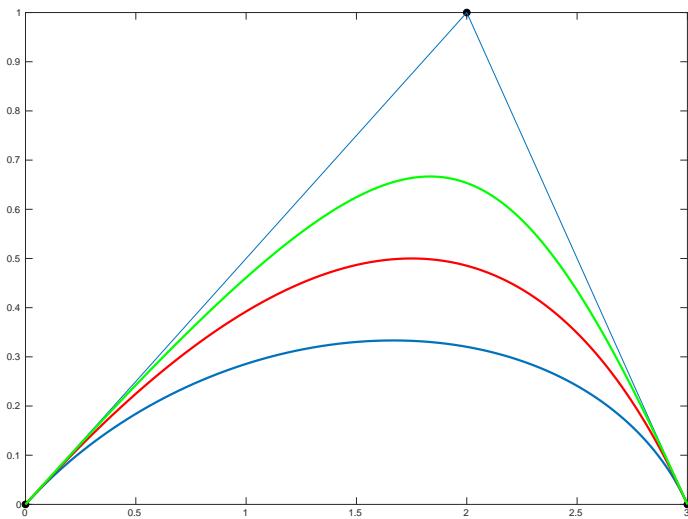
where the functions

$$R_{i,p}(u) := \frac{N_{i,p}(u) w_i}{\sum_{j=0}^n N_{j,p}(u) w_j}, \quad u \in [a, b], \quad i = 0, \dots, n$$

are the **rational B-spline basis functions**.

In the following we will choose $a = 0$, $b = 1$.

Example: rational Bézier curve of degree 2

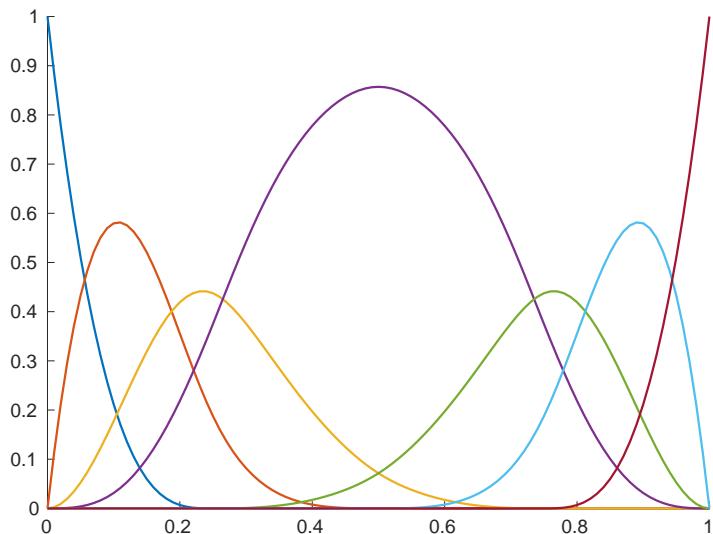


Weights $w_0 = w_2 = 1$. Blue: $w_1 = 0.5$ (ellipse). Red: $w_1 = 1$ (parabola). Green: $w_1 = 2$ (hyperbola).

Properties of rational B-spline basis functions

- ▶ Nonnegativity: $R_{i,p}(u) \geq 0$ for all $u \in [0, 1]$.
- ▶ Partition of unity: $\sum_{i=0}^n R_{i,p}(u) = 1$.
- ▶ $R_{0,p}(0) = R_{n,p}(1) = 1$.
- ▶ For $p > 0$, all $R_{i,p}(u)$ attain exactly one maximum on $[0, 1]$.
- ▶ Local (compact) support: $R_{i,p}(u) = 0$ for $u \notin [u_i, u_{i+p+1})$. Moreover, on a given knot span $[u_i, u_{i+1})$ the only (potentially) nonzero basis functions are $R_{i-p,p}(u), \dots, R_{i,p}(u)$.
- ▶ Smoothness related to knot multiplicity: at a knot of multiplicity k , the function $R_{i,p}(u)$ is $p - k$ times continuously differentiable. It is C^∞ in the interior of each knot span.
- ▶ B-splines are a particular case of NURBS: if the weights are all equal, then $R_{i,p}(u) = N_{i,p}(u)$ for all i .

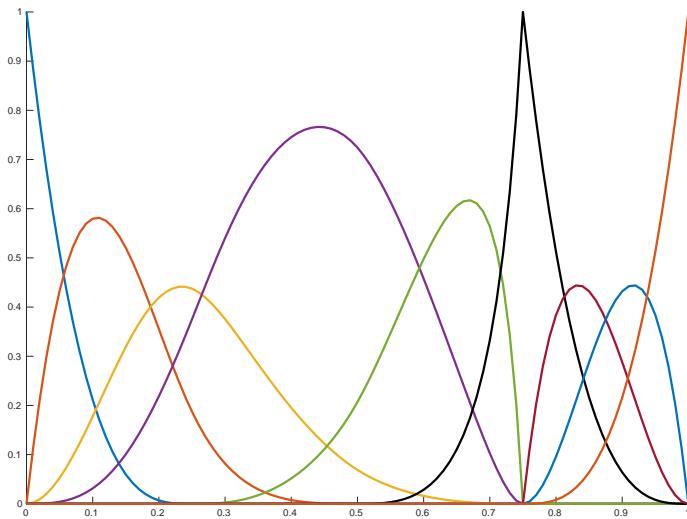
Example: rational basis functions of degree 3



Weights $\{1, 1, 1, 3, 1, 1, 1\}$.

Knots $\{0, 0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1, 1\}$.

Example: another rational basis functions of degree 3



Weights $\{1, 1, 1, 3, 2, 1, 1, 1, 1\}$.

Knots $\{0, 0, 0, 0, 0.25, 0.5, 0.75, 0.75, 0.75, 1, 1, 1, 1\}$.

Properties of rational B-spline basis functions

Theorem: The NURBS basis is normalized totally nonnegative.

Proof: Given values $t_1 < t_2 < \dots < t_r$ in $[0, 1]$, we need to prove that the NURBS collocation matrix

$$\begin{pmatrix} R_{1,p}(t_1) & \dots & R_{n,p}(t_1) \\ \vdots & & \vdots \\ R_{1,p}(t_r) & \dots & R_{n,p}(t_r) \end{pmatrix}$$

is totally nonnegative (TN).

Properties of rational B-spline basis functions

The NURBS collocation matrix is TN because it can be factored as the product of TN matrices:

$$\begin{pmatrix} w(t_1) \\ \ddots \\ w(t_r) \end{pmatrix} \begin{pmatrix} N_{1,p}(t_1) & \dots & N_{n,p}(t_1) \\ \vdots & & \vdots \\ N_{1,p}(t_r) & \dots & N_{n,p}(t_r) \end{pmatrix} \begin{pmatrix} w_1 \\ \ddots \\ w_r \end{pmatrix},$$

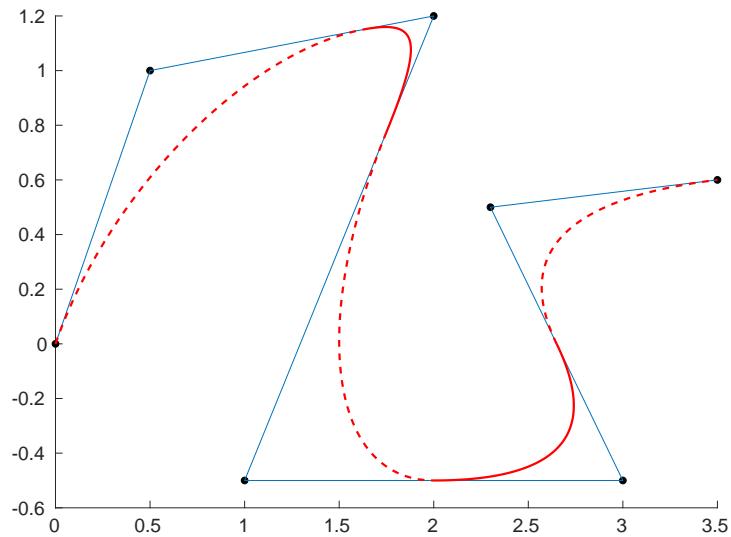
$$\text{where } w(t) = \frac{1}{\sum_{j=1}^n w_j N_{j,p}(t)}.$$

Normalization follows from partition of unity.

Properties of rational B-spline curves

- ▶ Endpoint interpolation: $\mathcal{C}(0) = P_0$ and $\mathcal{C}(1) = P_n$.
- ▶ Invariance under affine transformations.
- ▶ Strong convex hull property. *cioé locale*
- ▶ $\mathcal{C}(u)$ is C^∞ on the interior of knot spans and it is $p - k$ times differentiable at a knot of multiplicity k . *almeno*
- ▶ Variation diminishing property (consequence of previous theorem).
- ▶ Rational and nonrational Bézier curves, as well as polynomial B-splines, are special cases of rational B-splines.
- ▶ Local approximation: moving a control point P_i or changing a weight w_i affects only the portion of the curve on the interval $[u_i, u_{i+p+1}]$.

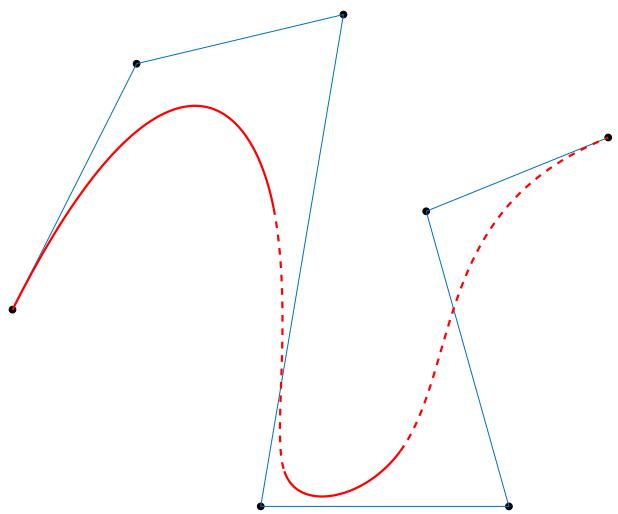
Example: NURBS curve of degree 2



Weights $\{1, 1, 3, 1, 1, 1, 1, 1\}$.

Knots $\{0, 0, 0, 0.2, 0.4, 0.6, 0.8, 1, 1, 1\}$.

Example: NURBS curve of degree 3

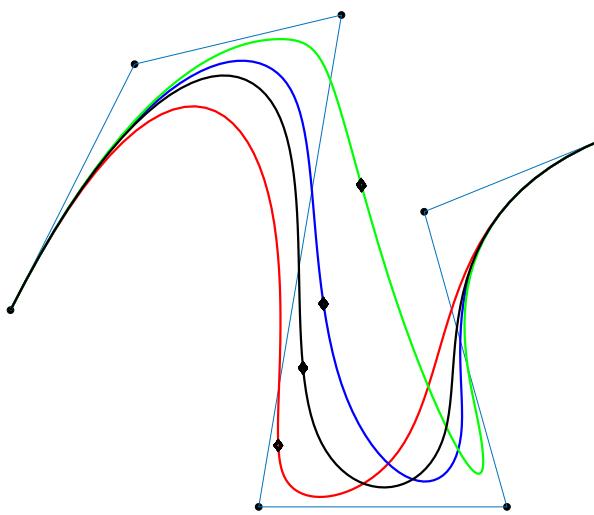


Weights $\{1, 1, 1, 3, 1, 1, 1\}$.

Knots $\{0, 0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1, 1\}$.

Example: NURBS curve of degree 3

Weights can be used for shape control.



se avessimo
più nodi: noteremmo
che la perturbazione
è locale

i punt. $C(0.45)$ al variare
di w sono allineati

Weights $\{1, 1, 1, w, 1, 1, 1\}$.

Knots $\{0, 0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1, 1\}$.

Red: $w = 3$. Black: $w = 1$. Blue: $w = 0.5$. Green: $w = 0.1$.

Black diamond: $C(0.45)$.

The effect of weight modification

The previous figure suggests that increasing (decreasing) w_k pushes the curve towards (away from) P_k . Moreover, for a fixed $u \in [u_k, u_{k+p+1})$, the curve points obtained for changing values of w_k lie on the same line. Let us prove this property.

$$C(u) = \frac{1}{N_{k,p}(u)w_k + \sum_{i \neq k} N_{i,p}(u)w_i} \left(N_{k,p}(u)w_k P_k + \sum_{i \neq k} N_{i,p}(u)w_i P_i \right)$$

Let us simplify the notation: set

$$A = N_{k,p}(u)w_k, \quad B = \sum_{i \neq k} N_{i,p}(u)w_i, \quad X = \sum_{i \neq k} N_{i,p}(u)w_i P_i.$$

The effect of weight modification

Then we have

$$\mathcal{C}(u) = \frac{1}{A+B}(AP_k + X)$$

and for $w_k = 0$, w_k non può "essere" zero, stiamo considerando

$$\mathcal{C}^0(u) = \frac{X}{B}.$$

The vector from $\mathcal{C}^0(u)$ to $\mathcal{C}(u)$ is given by

$$\begin{aligned}\mathcal{C}(u) - \mathcal{C}^0(u) &= \frac{1}{A+B}(AP_k + X) - \frac{X}{B} \\ &= \frac{A}{A+B} \left(P_k - \frac{X}{B} \right) \\ &= \frac{A}{A+B} (P_k - \mathcal{C}^0(u)).\end{aligned}$$

Since $0 \leq \frac{A}{A+B} \leq 1$, we conclude that $\mathcal{C}(u)$ lies on the segment of endpoints P_k and $\mathcal{C}^0(u)$.

Projective point of view

A rational B-spline curve in \mathbb{R}^d can be seen as projection of a polynomial B-spline curve in \mathbb{R}^{d+1} . For instance, suppose we have knots $\{u_j\}_{j=0}^m$ and control points $P_0, \dots, P_n \in \mathbb{R}^3$.

- ▶ Homogenize the coordinates of the control points using the weights w_0, \dots, w_n :

$$P_i = (x_i, y_i, z_i) \in \mathbb{R}^3 \longrightarrow P_i^w = (x_i w_i, y_i w_i, z_i w_i, w_i) \in \mathbb{R}^4.$$

- ▶ Construct the polynomial B-spline curve in \mathbb{R}^4 :

$$\mathcal{C}^w(u) := \sum_{i=0}^n N_{i,p}(u) P_i^w.$$

- ▶ Project on the hyperplane $\{W = 1\}$:

$$\mathcal{C}^w(u) \subset \mathbb{R}^4 \longrightarrow \mathcal{C}(u) \subset \mathbb{R}^3,$$

with

$$\mathcal{C}(u) = \sum_{i=0}^n R_{i,p}(u) P_i.$$

Derivatives of a NURBS curve

Let $\mathcal{C}(u)$ be a rational B-spline curve in \mathbb{R}^d . Write

$$\mathcal{C}(u) = \frac{w(u)\mathcal{C}(u)}{w(u)} = \frac{\mathbf{A}(u)}{w(u)},$$

where $\mathbf{A}(u) : [0, 1] \rightarrow \mathbb{R}^d$ is the vector-valued function that gives the first d coordinates of $\mathcal{C}^w(u)$.

Then we have

$$\mathcal{C}'(u) = \frac{w(u)\mathbf{A}'(u) - w'(u)\mathbf{A}(u)}{w(u)^2} = \frac{\mathbf{A}'(u) - w'(u)\mathcal{C}(u)}{w(u)}.$$

Since $\mathbf{A}(u)$ and $w(u)$ represent the coordinates of $\mathcal{C}^w(u)$, which is a polynomial B-spline curve, we can compute their derivatives using known formulas for polynomial B-splines.

Derivatives of a NURBS curve

For k -th derivatives we obtain:

$$\mathcal{C}^{(k)}(u) = \frac{\mathbf{A}^{(k)}(u) - \sum_{i=1}^k \binom{k}{i} w^{(i)}(u) \mathcal{C}^{(k-i)}(u)}{w(u)}.$$

In particular, first derivatives at endpoints are given by:

$$\mathcal{C}'(0) = \frac{w_1 p}{w_0 u_{p+1}} (\mathbf{P}_1 - \mathbf{P}_0)$$

$$\mathcal{C}'(1) = \frac{w_{n-1} p}{w_n (1 - u_{m-p-1})} (\mathbf{P}_n - \mathbf{P}_{n-1}).$$

Knot insertion

Knot insertion is a fundamental tool in the implementation of NURBS curves and surfaces.

Statement for curves:

- ▶ Let

$$\mathcal{C}^w(u) = \sum_{i=0}^n N_{i,p}(u) P_i^w$$

be a NURBS curve defined on the knot vector

$$U = \{u_0, \dots, u_m\}.$$

- ▶ Given $\bar{u} \in [u_k, u_{k+1})$, we want to insert \bar{u} into U , that is, define a new knot vector $\bar{U} = \{\bar{u}_0, \dots, \bar{u}_{m+1}\}$ with

$$\bar{u}_j = \begin{cases} u_j & \text{for } 0 \leq j \leq k, \\ \bar{u} & \text{for } j = k + 1, \\ u_{j-1} & \text{for } k + 2 \leq j \leq m + 1. \end{cases}$$

Knot insertion

$\mathcal{C}^w(u)$ has a new representation on \bar{U} of the form

$$\mathcal{C}^w(u) = \sum_{i=0}^{n+1} \bar{N}_{i,p}(u) Q_i^w.$$

Problem: find the $\{Q_i^w\}_{i=0}^{n+1}$.

The image of the curve stays the same. We are simply looking for a new representation that uses a different (larger) vector space of functions.

Applications include:

- ▶ evaluation of points on curves and surfaces,
- ▶ subdivision of curves and surfaces,
- ▶ increase flexibility in shape control by adding control points.

Knot insertion

We look for the points $\{Q_i^w\}_{i=0}^{n+1}$ that solve

$$\sum_{i=0}^n N_{i,p}(u) P_i^w = \sum_{i=0}^{n+1} \bar{N}_{i,p}(u) Q_i^w$$

Matrix-based solution:

- ▶ Substitute $n + 2$ suitable values of u in the equation.
- ▶ Solve the resulting linear system. This is a non-singular, banded system of $n + 2$ equations in $n + 2$ unknowns.

*perché le $N_{i,p}$ hanno
supporto locale*

Knot insertion

$$\sum_{i=0}^n N_{i,p}(u) P_i^w = \sum_{i=0}^{n+1} \bar{N}_{i,p}(u) Q_i^w$$

But there is a more efficient approach: from $\bar{u} \in [u_k, u_{k+1})$ and known properties of the B-spline basis functions we deduce (compare knot insertion for B-splines in previous slides):

$$Q_i^w = \alpha_i P_i^w + (1 - \alpha_i) P_{i-1}^w,$$

where

$$\alpha_i = \begin{cases} 1 & i \leq k-p \\ \frac{\bar{u}-u_i}{u_{i+p}-u_i} & k-p+1 \leq i \leq k \\ 0 & i \geq k+1 \end{cases}.$$

Note that only p new control points must be computed.

Fundamental algorithms

Besides knot insertion, other fundamental geometric algorithms include:

- ▶ knot refinement (insertion of many knots at once),
- ▶ knot removal (reverse process of knot insertion),
- ▶ degree elevation,
- ▶ degree reduction.

For more details see e.g., *The NURBS Book* by Les Piegl and Wayne Tiller, Springer 1995.

Reparameterization

- ▶ NURBS curves can be **reparameterized** in order to modify weights and knots without changing the image of the curve.
- ▶ Let us use **Möbius transformations**, so that a rational B-spline curve $\mathcal{C}(u)$ is transformed into another rational B-spline curve $\tilde{\mathcal{C}}(s)$ of the same degree):

$$g(u) = \frac{\alpha u + \beta}{\gamma u + \delta}, \quad f(s) = g^{-1}(s) = \frac{\delta s + \beta}{\gamma s - \alpha},$$

where $\alpha\delta - \gamma\beta \neq 0$ and the denominators do not vanish on the chosen domains.

- ▶ Then the reparameterized curve $\tilde{\mathcal{C}}(s) := \mathcal{C}(f(s))$ has the **same control points** as $\mathcal{C}(u)$, **new knots**

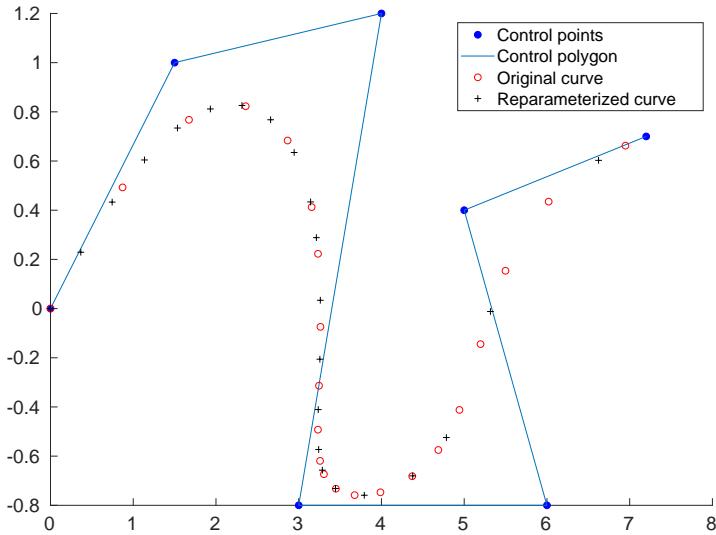
$$s_i = g(u_i), \quad i = 0, \dots, m,$$

and **new weights**

$$\tilde{w}_i = w_i \prod_{j=1}^p (\gamma s_{i+j} - \alpha) = \frac{w_i}{\prod_{j=1}^p (\gamma u_{i+j} + \delta)}, \quad i = 0, \dots, n.$$

Reparameterization: an example

Take $\mathcal{C}(u)$ with knots $\{0, 0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1, 1\}$ and weights $\{1, 1, 1, 3, 1, 1, 1\}$ (previous example), $s = g(u) = \frac{2u+1}{3u+2}$.



The plot shows a uniform sampling of both curves.

Reparameterization: curves in standard form

We want the reparameterized curve $\tilde{\mathcal{C}}(s)$ to be in **standard form**, that is, such that $\tilde{w}_0 = \tilde{w}_n$. Let us look for a suitable reparameterization.

From the formula for new weights:

$$\tilde{w}_0 = \frac{w_0}{\prod_{j=1}^p (\gamma u_j + \delta)} = \frac{w_0}{(\gamma u_0 + \delta)^p}$$

$$\tilde{w}_n = \frac{w_n}{\prod_{j=1}^p (\gamma u_{n+j} + \delta)} = \frac{w_n}{(\gamma u_{n+1} + \delta)^p}$$

so the condition we need to impose is

$$\frac{\gamma u_0 + \delta}{\gamma u_{n+1} + \delta} = \sqrt[p]{\frac{w_0}{w_n}} =: \omega \neq 1.$$

This yields $\gamma(\omega u_{n+1} - u_0) = \delta(1 - \omega)$.

Reparameterization: curves in standard form

The other coefficients α and β are determined by $s_0 = g(u_0)$ and $s_{n+1} = g(u_{n+1})$. The reparameterization function we obtain is

$$s = g(u) = \frac{\sqrt[p]{w_0}(u_{n+1} - u)s_0 + \sqrt[p]{w_n}(u - u_0)s_{n+1}}{\sqrt[p]{w_0}(u_{n+1} - u) + \sqrt[p]{w_n}(u - u_0)}.$$

We conclude that any NURBS curve can be put in standard form through a suitable reparameterization.

Therefore we can assume to work with NURBS curves having $w_0 = w_n = 1$.

Ex Rappresentare la circonferenza unitaria come NURBS

Sol Studieremo lo **Schema a 9 punti** (uno dei tanti possibili.)

Abbiamo già visto come rappresentare un quarto di circonferenza come curva di Bezier razionale, con

$$P_0 = (1, 0), \quad P_1 = (1, 1) \quad e \quad P_2 = (0, 1)$$

$$w_0 = w_1 = 1, \quad w_2 = 2$$

Adesso sappiamo che si può ri-parametrizzare per imporre $w_0 = w_2 = 1$, ottenendo $w_1 = \frac{\sqrt{2}}{2}$

Possiamo rappresentare l'intera circonferenza con 4 quarti:

$$\text{di circonferenza prendendo } P_3 = (-1, 1), \quad P_4 = (-1, 0)$$

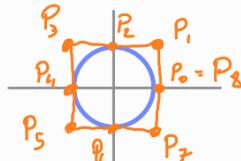
$$P_5 = (-1, -1), \quad P_6 = (0, -1), \quad P_7 = (1, -1), \quad P_8 = P_0$$

La curva è di grado $p=2$, abbiamo $n=8$ (per 9 punti)

e dobbiamo scegliere i nodi:

$$m = n + p + 1 = 11$$

$$U = \left\{ 0, 0, 0, \frac{1}{4}, \frac{1}{4}, \frac{1}{2}, \frac{1}{2}, \frac{3}{4}, \frac{3}{4}, 1, 1, 1 \right\}$$



In teoria avere nodi multipli dovrebbe abbassare la regolarità, ma poiché i punti sono allineati ciò non accade

Più precisamente (da dimostrare) la B-spline omogeneizzata è C^0 ma non C^1 nei nodi di molteplicità 2, ma la NURBS ottenuta proiettando è C^1 ovunque

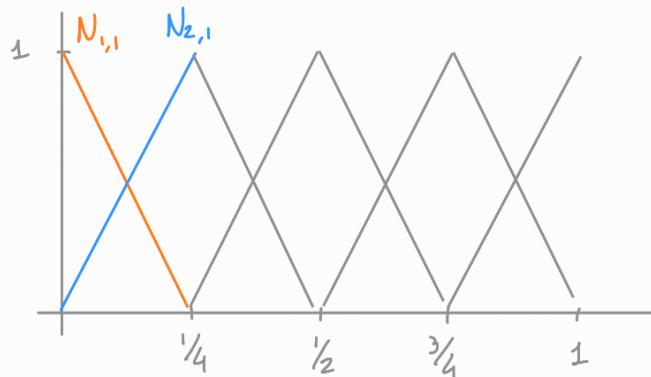
$$\text{Per i pesi prendo } w = \left\{ 1, \frac{\sqrt{2}}{2}, \frac{1}{2}, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, \frac{1}{2}, \frac{\sqrt{2}}{2}, 1 \right\}$$

Riprendendo l'esercizio della scorsa lezione, studiamo il comportamento della derivata di $C(u)$ e di $C^w(u)$ nel nodo $\frac{1}{4}$

Consideriamo prima C^w . Vale $u_2 = 0$, $u_3 = \frac{1}{4}$, prendiamo per derivata sx

$u \in [u_2, u_3]$. Vale $C^{w'}(u) = \sum_{i=0}^7 N_{i+1,1}(u) Q_i^w$. Le $N_{i,1}$ sono

della forma



per via della
moltiplicità de:
nodi:

Allora $C^{w'}(u) = N_{0,1}(u) Q_0^w + N_{1,1}(u) Q_1^w$

$$\text{dove } Q_0^w = 2 \frac{P_1^w - P_0^w}{\frac{1}{4}} = 8 \begin{pmatrix} \frac{\sqrt{2}}{2} - 1 \\ \frac{\sqrt{2}}{2} - 0 \\ \frac{\sqrt{2}}{2} - 1 \end{pmatrix}, \quad Q_1^w = 8 \begin{pmatrix} 0 - \frac{\sqrt{2}}{2} \\ 1 - \frac{\sqrt{2}}{2} \\ 1 - \frac{\sqrt{2}}{2} \end{pmatrix}$$

Calcoliamo $\lim_{u \rightarrow \frac{1}{4}^-} C^{w'}(u)$

$$\lim_{u \rightarrow \frac{1}{4}^-} C^{w'}(u) = \lim_{u \rightarrow \frac{1}{4}^-} N_{0,1}(u) Q_0^w + N_{1,1}(u) Q_1^w = Q_1^w = 4 \begin{pmatrix} -\sqrt{2} \\ 2 - \sqrt{2} \\ 2 - \sqrt{2} \end{pmatrix}$$

Per la derivata destra prendiamo $u \in [\frac{1}{4}, \frac{1}{2})$ e vale

$$\begin{aligned} \lim_{u \rightarrow \frac{1}{4}^+} C^{w'}(u) &= \lim_{u \rightarrow \frac{1}{4}^+} N_{3,1}(u) Q_2^w + N_{4,1}(u) Q_3^w = Q_2^w = 2 \frac{P_3^w - P_2^w}{\frac{1}{4}} = \\ &= 4 \begin{pmatrix} -\sqrt{2} \\ \sqrt{2} - 2 \\ \sqrt{2} - 2 \end{pmatrix} \end{aligned}$$

Quindi: $\lim_{u \rightarrow \frac{1}{4}^-} \mathcal{C}'(u) \neq \lim_{u \rightarrow \frac{1}{4}^+} \mathcal{C}'(u)$ quindi $\mathcal{C}(u)$ non è C^1 ne' nodi

Per quanto riguarda $\mathcal{C}(u)$, vale

$$\mathcal{C}'(u) = \frac{A'(u) - w'(u) \mathcal{C}(u)}{w(u)} \quad \text{con} \quad \mathcal{C}^w(u) = \begin{pmatrix} A(u) \\ w(u) \end{pmatrix}$$

Allora per quanto calcolato prima

$$\lim_{u \rightarrow \frac{1}{4}^-} A'(u) = \begin{pmatrix} -4\sqrt{2} \\ 8-4\sqrt{2} \end{pmatrix} \quad \lim_{u \rightarrow \frac{1}{4}^+} A'(u) = \begin{pmatrix} -4\sqrt{2} \\ 4\sqrt{2}-8 \end{pmatrix}$$

$$\lim_{u \rightarrow \frac{1}{4}^-} w'(u) = 8-4\sqrt{2} \quad \lim_{u \rightarrow \frac{1}{4}^+} w'(u) = 4\sqrt{2}-8$$

$$\text{Inoltre } w\left(\frac{1}{4}\right) = \sum_{i=0}^8 N_{i,2}\left(\frac{1}{4}\right) w_i = \frac{N_{2,2}\left(\frac{1}{4}\right)}{1} w_2 = 1$$

$$\mathcal{C}\left(\frac{1}{4}\right) = P_2$$

$$\text{Quindi: } \lim_{u \rightarrow \frac{1}{4}^-} \mathcal{C}'(u) = \frac{\begin{pmatrix} -4\sqrt{2} \\ 8-4\sqrt{2} \end{pmatrix} - (8-4\sqrt{2}) \begin{pmatrix} 0 \\ 1 \end{pmatrix}}{1} = \begin{pmatrix} -4\sqrt{2} \\ 0 \end{pmatrix}$$

$$\lim_{u \rightarrow \frac{1}{4}^+} \mathcal{C}'(u) = \frac{\begin{pmatrix} -4\sqrt{2} \\ 4\sqrt{2}-8 \end{pmatrix} - (4\sqrt{2}-8) \begin{pmatrix} 0 \\ 1 \end{pmatrix}}{1} = \begin{pmatrix} -4\sqrt{2} \\ 0 \end{pmatrix}$$

ovvero $\mathcal{C}(u)$ è C^1 in $\frac{1}{4}$ (e analogamente negli altri nodi)

Oss Nell'esercizio abbiamo utilizzato lo schema a 9 punti inscrivendo la circonferenza in un quadrato, rompendola in segmenti, rappresentando i segmenti come curve razionali di Bezier e unendole. In realtà questo procedimento può essere eseguito inscrivendo in un qualsiasi poligono regolare. Con un triangolo equilatero si ottiene lo schema a 7 punti.

Per la circonferenza con schema a 7 punti abbiamo

$$n=6, \quad m=9, \quad \omega = (1, \frac{1}{2}, 1, \frac{1}{2}, 1, \frac{1}{2}, 1)$$

$$U = (0, 0, 0, \frac{1}{3}, \frac{1}{3}, \frac{2}{3}, \frac{2}{3}, 1, 1, 1)$$

Tensor product surfaces

Just like a curve $\mathcal{C}(u)$ is represented by a vector-valued function of one parameter, a surface is represented by a vector-valued function of two parameters $\mathcal{S}(u, v)$. For a surface in \mathbb{R}^3 , this function takes the form

$$\mathcal{S}(u, v) = (x(u, v), y(u, v), z(u, v)).$$

A widespread way to represent surfaces is the **tensor product scheme**. Just as in curve representation, it uses basis functions and geometric coefficients.

- ▶ The **basis functions** are bivariate functions of u and v , which are constructed as products of univariate basis functions.
- ▶ The **geometric coefficients** (control points) are arranged in a bidirectional $n \times m$ net.

Tensor product surfaces

A **tensor product surface** takes the form

$$\mathcal{S}(u, v) = \sum_{i=0}^n \sum_{j=0}^m f_i(u)g_j(v)\mathbf{P}_{i,j}$$

with

$$\mathbf{P}_{i,j} = (x_{i,j}, y_{i,j}, z_{i,j}), \quad 0 \leq u, v \leq 1.$$

Here $\{f_i(u)\}_{i=0}^n$, $\{g_j(v)\}_{j=0}^m$ are univariate basis functions.

The **tensor product basis** is formed by the functions $\{f_i(u)g_j(v)\}_{0 \leq i \leq n, 0 \leq j \leq m}$.

Isoparametric curves

Fix $u = u_0$; then

$$\mathcal{C}_{u_0}(v) := \mathcal{S}(u_0, v) = \sum_{j=0}^m Q_j(u_0) g_j(v),$$

where

$$Q_j(u_0) = \sum_{i=0}^n P_{i,j} f_i(u_0).$$

is a curve lying on $\mathcal{S}(u, v)$, called **isoparametric curve**. Same for $\mathcal{C}_{v_0}(u) := \mathcal{S}(u, v_0)$.

The **boundary curves** are the four special isoparametric curves $\mathcal{S}(0, v)$, $\mathcal{S}(1, v)$, $\mathcal{S}(u, 0)$ and $\mathcal{S}(u, 1)$.

Power basis and Bézier surfaces

Power basis surfaces: defined by

$$\mathcal{S}(u, v) = \sum_{i=0}^n \sum_{j=0}^m P_{i,j} u^i v^j.$$

Bézier surfaces: defined by

$$\mathcal{S}(u, v) = \sum_{i=0}^n \sum_{j=0}^m B_i^n(u) B_j^m(v) P_{i,j}, \quad 0 \leq u, v \leq 1,$$

where $B_i^n(u)$, $B_j^m(v)$ are Bernstein polynomials.

Bézier surfaces

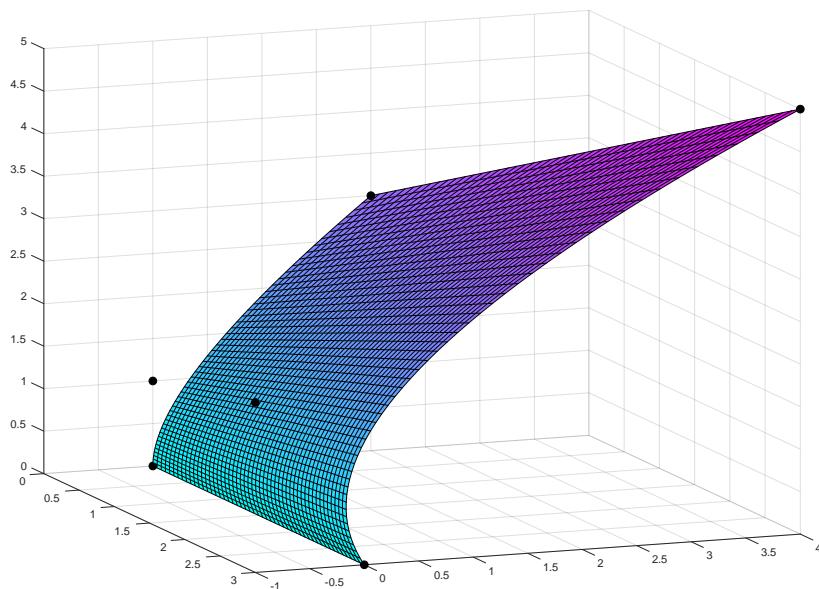
Properties of Bézier surfaces:

- ▶ invariant for affine transformations (because $\sum_{i=0}^n \sum_{j=0}^m B_i^n(u) B_j^m(v) = 1$),
- ▶ contained in the convex hull of their control points,
- ▶ interpolation of the four corner control points,
- ▶ when triangulated, the control net forms a planar polyhedral approximation to the surface.

Note that there is no known variation diminishing property for Bézier surfaces.

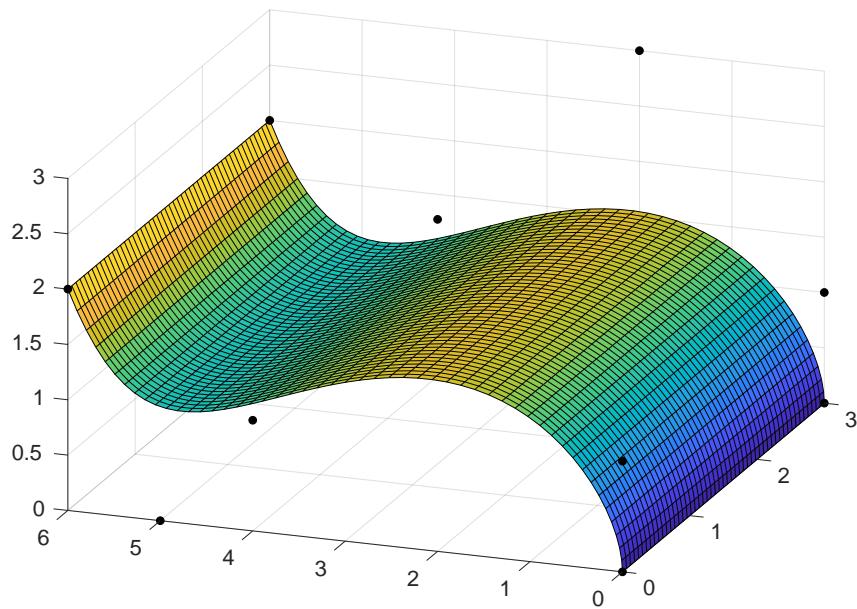
An example

Take $n = 1, m = 2$:



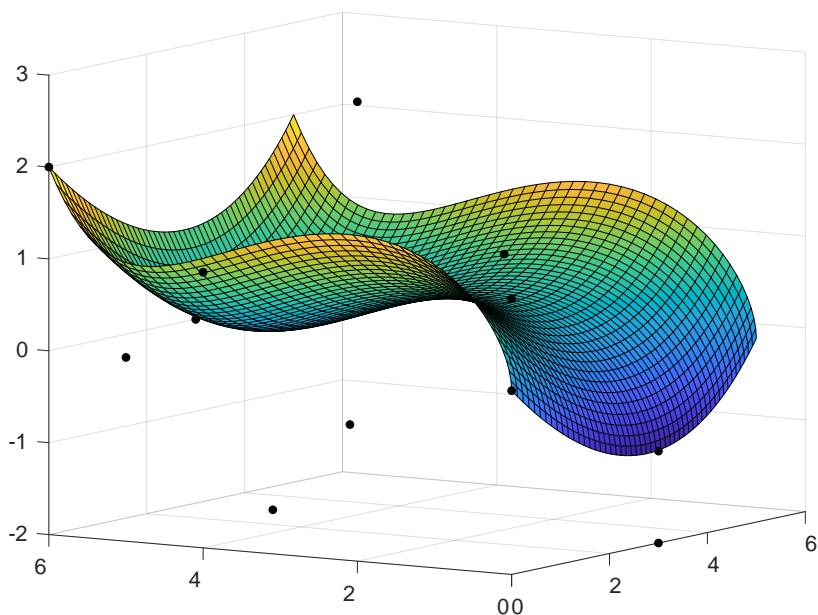
Another example

Take $n = 1, m = 5$:



And yet another one

Take $n = 2, m = 5$:



Bézier surfaces

The **de Casteljau algorithm** extends to Bézier surfaces.

- ▶ Fix $(u_0, v_0) \in [0, 1] \times [0, 1]$.
- ▶ Consider an index j_0 and the associated sequence of control points $\{P_{i,j_0}\}_{i=0}^n$. The classical de Casteljau algorithm applied to $\{P_{i,j_0}\}_{i=0}^n$ provides the point $Q_{j_0}(u_0) = \sum_{i=0}^n B_i^n(u_0) P_{i,j_0}$.
- ▶ Therefore if we apply de Casteljau for each j , that is, $m + 1$ times, we obtain all the points $Q_j(u_0)$ for $j = 0, \dots, m$.
- ▶ Now apply de Casteljau one more time to the points $Q_j(u_0)$ with parameter v_0 and obtain

$$S(u_0, v_0) = \sum_{j=0}^m B_j^m(v_0) Q_j(u_0).$$

The process requires

$$\frac{n(n+1)(m+1)}{2} + \frac{m(m+1)}{2}$$

linear interpolations. (By symmetry, better start on isoparametric curves with fixed values of v if $n > m$).

B-spline surfaces

Given a net of control points $P_{i,j}$, $i = 0, \dots, n$, $j = 0, \dots, m$, and two knot vectors of lengths $r + 1$, $s + 1$

$$U = \underbrace{\{0, \dots, 0\}}_{p+1}, u_{p+1}, \dots, u_n, \underbrace{\{1, \dots, 1\}}_{p+1},$$

$$V = \underbrace{\{0, \dots, 0\}}_{q+1}, v_{q+1}, \dots, v_m, \underbrace{\{1, \dots, 1\}}_{q+1},$$

with $r = n + p + 1$ and $s = m + q + 1$, the associated B-spline surface is defined as

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) P_{i,j}.$$

B-spline surfaces

Properties of the tensor product B-spline basis functions:

- ▶ Nonnegativity: $N_{i,p}(u)N_{j,q}(v) \geq 0$ for all i, j, u, v .
- ▶ Partition of unity: $\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u)N_{j,q}(v) = 1$ for all $(u, v) \in [0, 1] \times [0, 1]$.
- ▶ If $n = p$, $m = q$, $U = \{0, \dots, 0, 1, \dots, 1\}$ and $V = \{0, \dots, 0, 1, \dots, 1\}$, then $N_{i,p}(u)N_{j,q}(v) = B_i^n(u)B_j^m(v)$, that is, products of B-spline functions degenerate to products of Bernstein polynomials.
- ▶ In any given knot rectangle $[u_{i_0}, u_{i_0+1}) \times [v_{j_0}, v_{j_0+1})$, at most $(p+1)(q+1)$ basis functions are nonzero, that is, the functions $N_{i,p}(u)N_{j,q}(v)$ for $i_0 - p \leq i \leq i_0$ and $j_0 - q \leq j \leq j_0$.

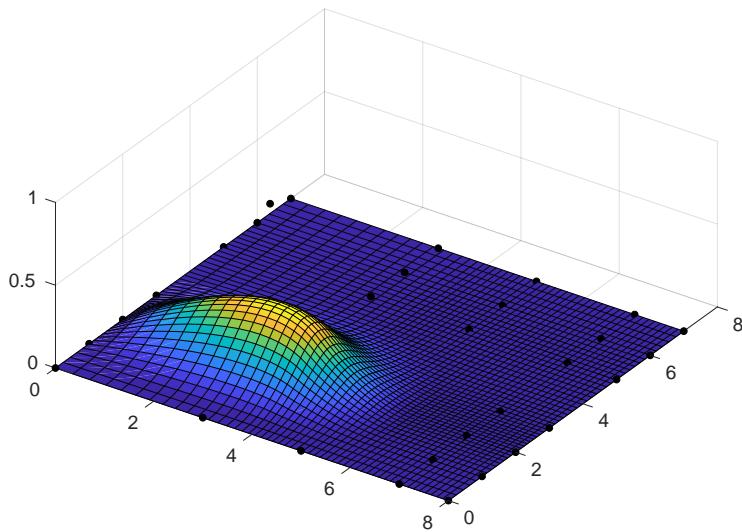
B-spline surfaces

Properties of the tensor product B-spline basis functions:

- ▶ If $p > 0$ and $q > 0$, then the basis function $N_{i,p}(u)N_{j,q}(v)$ attains exactly one maximum value.
- ▶ $N_{i,p}(u)N_{j,q}(v)$ is a bivariate polynomial on the interior of each knot rectangle; in particular, all its partial derivatives exist. At a u knot of multiplicity k , each basis function is $p - k$ times differentiable in the u direction. At a v knot of multiplicity h , each basis function is $q - h$ times differentiable in the v direction.

Examples of tensor product B-spline basis functions

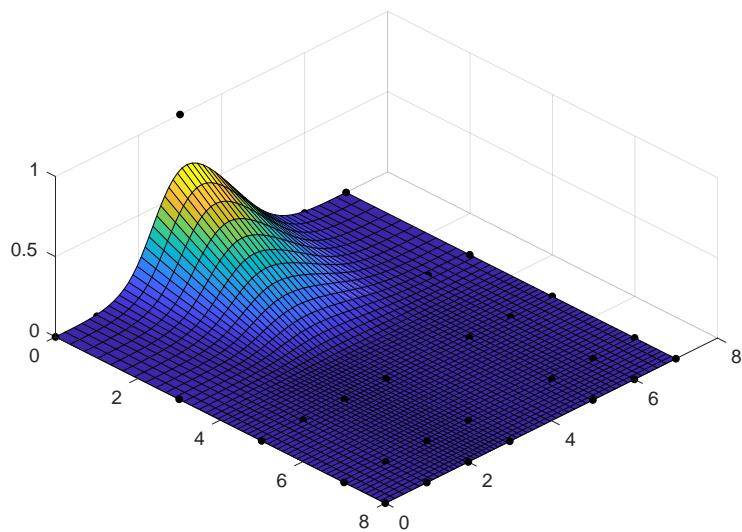
Take $U = \{0, 0, 0, 0.33, 0.66, 1, 1, 1\}$ and
 $V = \{0, 0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1, 1\}$, with a 5×7 control net.



Plot of $N_2^2(u)N_3^3(v)$.

Examples of tensor product B-spline basis functions

Take $U = \{0, 0, 0, 0.33, 0.66, 1, 1, 1\}$ and
 $V = \{0, 0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1, 1\}$, with a 5×7 control net.



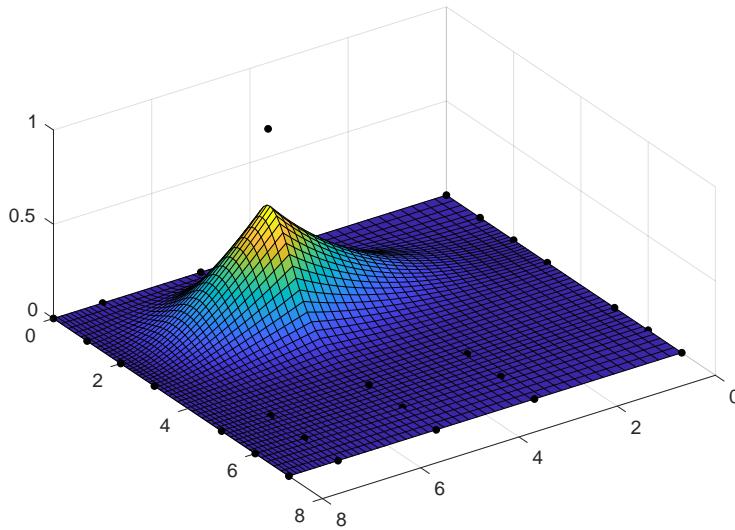
Plot of $N_1^2(u)N_4^3(v)$.

Examples of tensor product B-spline basis functions

nodo multiplo, quando cresta

Take $U = \{0, 0, 0, 0.5, 0.5, 1, 1, 1\}$ and

$V = \{0, 0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1, 1\}$, with a 5×7 control net.



Plot of $N_3^2(u)N_3^3(v)$.

B-spline surfaces

Properties of B-spline surfaces:

- ▶ If $n = p$, $m = q$, $U = \{0, \dots, 0, 1, \dots, 1\}$ and $V = \{0, \dots, 0, 1, \dots, 1\}$, then $\mathcal{S}(u, v)$ is a Bézier surface.
- ▶ The surface interpolates the corner control points $P_{0,0}$, $P_{n,0}$, $P_{0,m}$, $P_{n,m}$.
- ▶ Invariance w.r.t. affine transformations.
- ▶ Strong convex hull property: if $(u, v) \in [u_{i_0}, u_{i_0+1}] \times [v_{j_0}, v_{j_0+1}]$, then $\mathcal{S}(u, v)$ is in the convex hull of the control points $P_{i,j}$ for $i_0 - p \leq i \leq i_0$ and $j_0 - q \leq j \leq j_0$.
- ▶ If triangulated, the control net forms a piecewise planar approximation to the surface. As for curves, the lower the degree, the better the approximation.

B-spline surfaces

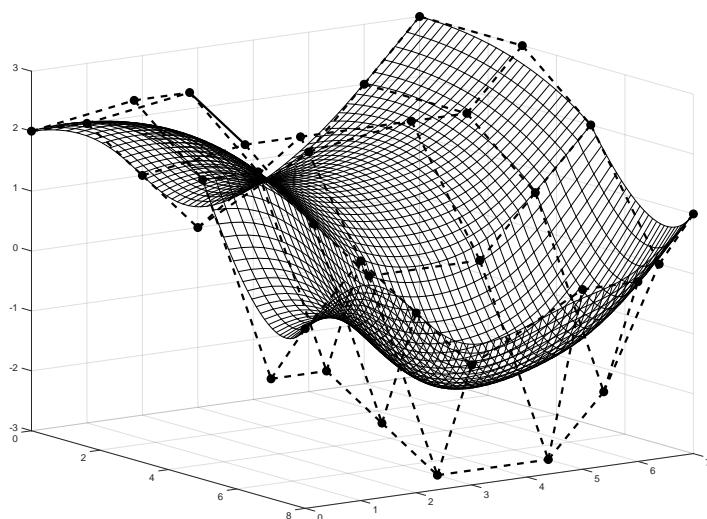
Properties of B-spline surfaces:

- ▶ Local modification: moving $P_{i,j}$ affects the surface only on the rectangle $[u_i, u_{i+p+1}) \times [v_j, v_{j+q+1})$.
- ▶ The surface is differentiable $p - k$ times in the u direction at a u knot of multiplicity k . It is differentiable $q - h$ times in the v direction at a v knot of multiplicity h .

There is no known variation diminishing property for B-spline surfaces.

B-spline surfaces

Knots $U = \{0, 0, 0, 0.5, 0.5, 1, 1, 1\}$ and $V = \{0, 0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1, 1\}$, with a 5×7 control net as above.



De Boor's algorithm for surfaces

De Boor's algorithm for B-spline curves extends to surfaces. The equation of a B-spline surface can be rewritten as

$$S(u, v) = \sum_{i=0}^n N_{i,p}(u) \left(\sum_{j=0}^m N_{j,q}(v) P_{i,j} \right),$$

so for a fixed i_0 , the expression in red is a B-spline curve defined by control points $P_{i_0,j}$, $j = 0, \dots, m$.

De Boor's algorithm for surfaces

Given parameters (u, v) where we want to evaluate the surface, we can proceed as follows:

- ▶ find h, k such that $u \in [u_h, u_{h+1}]$ and $v \in [v_k, v_{k+1}]$,
- ▶ for $i = h - p$ to h , apply de Boor's algorithm (for curves) to control points $P_{i,k-q}, \dots, P_{i,k}$ w.r.t. v , yielding points Q_i ,
- ▶ apply again de Boor's algorithm to points Q_{h-p}, \dots, Q_h w.r.t. u , yielding $S(u, v)$.

Rational Bézier surfaces

A **rational Bézier surface** is the perspective projection in \mathbb{R}^3 of a polynomial Bézier surface in \mathbb{R}^4 .

$$\mathcal{S}^w(u, v) = \sum_{i=0}^n \sum_{j=0}^m B_i^n(u) B_j^m(v) P_{i,j}^w$$

$$\mathcal{S}(u, v) = H(\mathcal{S}^w(u, v)) = \sum_{i=0}^n \sum_{j=0}^m R_{i,j}(u, v) P_{i,j}$$

with

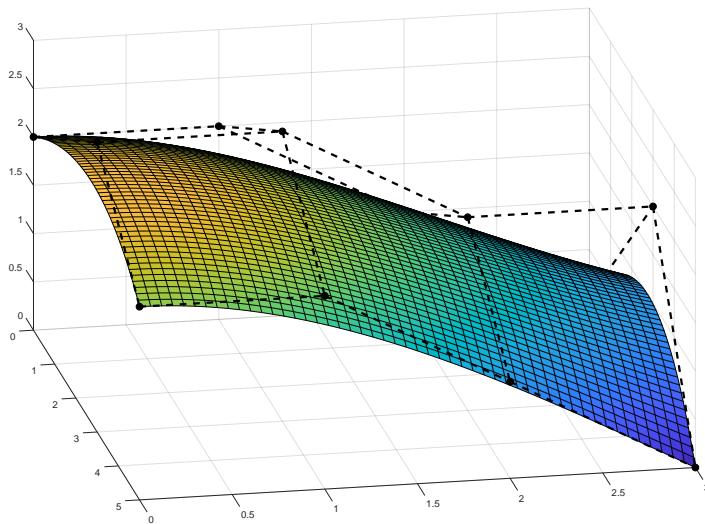
$$R_{i,j}(u, v) = \frac{B_i^n(u) B_j^m(v) w_{i,j}}{\sum_{r=0}^n \sum_{s=0}^m B_r^n(u) B_s^m(v) w_{r,s}}.$$

Rational Bézier surfaces

- ▶ Note that $R_{i,j}(u, v)$ is not the product of basis functions. Therefore $\mathcal{S}(u, v)$ is not a tensor product surface (but $\mathcal{S}^w(u, v)$ is).
- ▶ Assuming positive weights, the properties listed earlier for polynomial Bézier surfaces also hold for rational Bézier surfaces.
- ▶ If all the $w_{i,j}$ are equal, then $R_{i,j}(u, v) = B_i^n(u) B_j^m(v)$. Therefore, polynomial Bézier surfaces are a special case of rational Bézier surfaces.

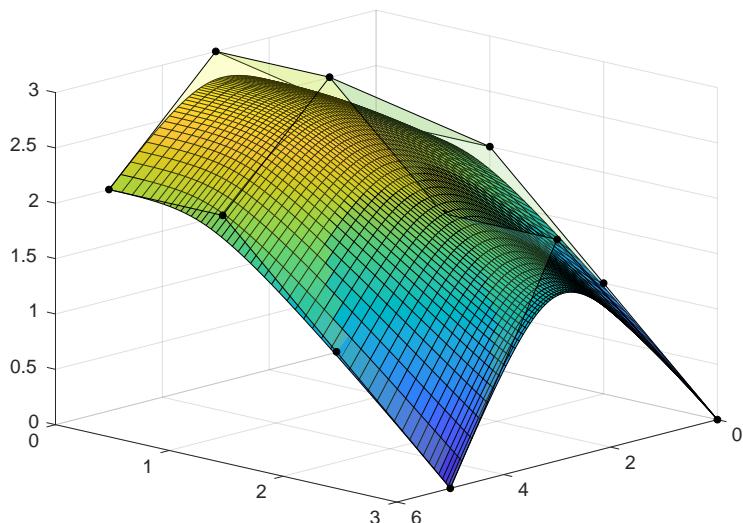
Rational Bézier surfaces

A quadratic \times cubic Bézier surface...



Rational Bézier surfaces

...and a rational Bézier surface with same control net but nontrivial weights.



NURBS surfaces

A NURBS surface of degree p in the u direction and of degree q in the v direction is a bivariate vector-valued piecewise rational function of the form

$$S(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) w_{i,j} P_{i,j}}{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) w_{i,j}}, \quad 0 \leq u, v \leq 1,$$

where

- ▶ the points $P_{i,j}$, $0 \leq i \leq n$, $0 \leq j \leq m$ form the **control net**,
- ▶ the $w_{i,j}$ are **positive weights**,
- ▶ the B-spline basis functions $N_{i,p}(u)$ and $N_{j,q}(v)$ are defined w.r.t. knot vectors

$$U = \{ \underbrace{0, \dots, 0}_{p+1}, u_{p+1}, \dots, u_n, \underbrace{1, \dots, 1}_{p+1} \},$$

$$V = \{ \underbrace{0, \dots, 0}_{q+1}, v_{q+1}, \dots, v_m, \underbrace{1, \dots, 1}_{q+1} \}.$$

NURBS surfaces

We introduce the **rational basis functions**

$$R_{i,j}(u, v) = \frac{N_{i,p}(u) N_{j,q}(v) w_{i,j}}{\sum_{k=0}^n \sum_{\ell=0}^m N_{k,p}(u) N_{\ell,q}(v) w_{k,\ell}}$$

and write

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m R_{i,j}(u, v) P_{i,j}.$$

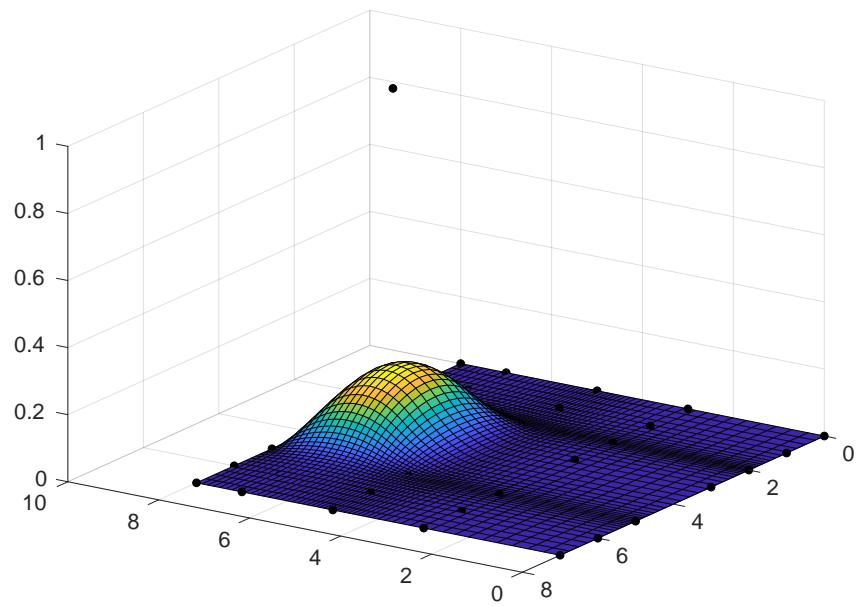
In homogeneous coordinates:

$$S^w(u, v) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) P_{i,j}^w$$

$$H(S^w(u, v)) = S(u, v),$$

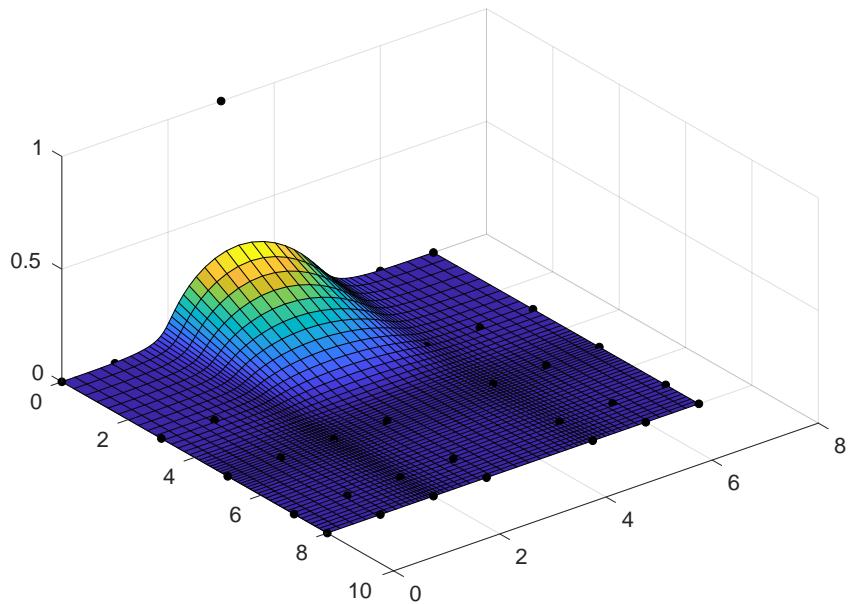
where H is the usual perspective projection.

NURBS surface basis functions



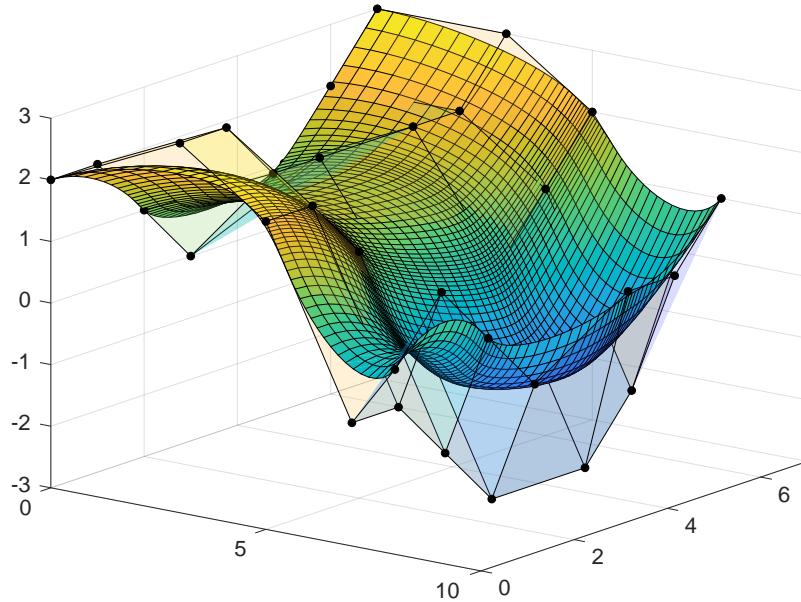
$$R_{3,3}(u, v), \quad p = 2, \quad q = 3.$$

NURBS surface basis functions



$$R_{0,3}(u, v), \quad p = 2, \quad q = 3.$$

NURBS surfaces



Derivatives of NURBS surfaces

We proceed just like for curves:

$$\mathcal{S}(u, v) = \frac{\mathbf{A}(u, v)}{w(u, v)}$$

with

$$\mathbf{A}(u, v) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) w_{i,j} \mathbf{P}_{i,j},$$

$$w(u, v) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) w_{i,j}.$$

For partial first derivatives we have

$$\mathcal{S}_u(u, v) = \frac{\mathbf{A}_u(u, v) - w_u(u, v)\mathcal{S}(u, v)}{w(u, v)},$$

$$\mathcal{S}_v(u, v) = \frac{\mathbf{A}_v(u, v) - w_v(u, v)\mathcal{S}(u, v)}{w(u, v)}.$$

Derivatives of NURBS surfaces

In particular, we obtain at the $(0, 0)$ corner:

$$\mathcal{S}_u(0, 0) = \frac{p w_{1,0}}{u_{p+1} w_{0,0}} (P_{1,0} - P_{0,0}),$$

$$\mathcal{S}_v(0, 0) = \frac{q w_{0,1}}{v_{q+1} w_{0,0}} (P_{0,1} - P_{0,0}).$$

Weight modification

The effect of modifying one surface weight $w_{k,\ell}$ is similar to the curve case. Fix $u_0 \in [u_k, u_{k+p+1})$ and $v_0 \in [v_\ell, v_{\ell+q+1})$ and define points

$$R = \mathcal{S}(u_0, v_0; w_{k,\ell} = 0),$$

cioé limite per $w_{k,\ell} \rightarrow 0$

$$P = \mathcal{S}(u_0, v_0; w_{k,\ell} > 0).$$

Then we have

$$P = (1 - t)R + t P_{k,\ell}$$

with

$$t = \frac{N_{k,p}(u_0)N_{\ell,q}(v_0)w_{k,\ell}}{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u_0)N_{j,q}(v_0)w_{i,j}}.$$

Matlab implementation

- ▶ Curve Fitting Toolbox: use `rsmak` to build rational B-spline and `fnplt` to plot.

```
sp = rsmak(knots,hcoefs);  
fnplt(sp)
```

- ▶ NURBS Toolbox (by D. M. Spink): use `nrbmak` (or one of several variants) to build rational B-spline and `nrbplot` to plot.

```
sp = nrbmak(hcoefs,knots);  
nrbplot(sp)
```

Bilinear Surfaces

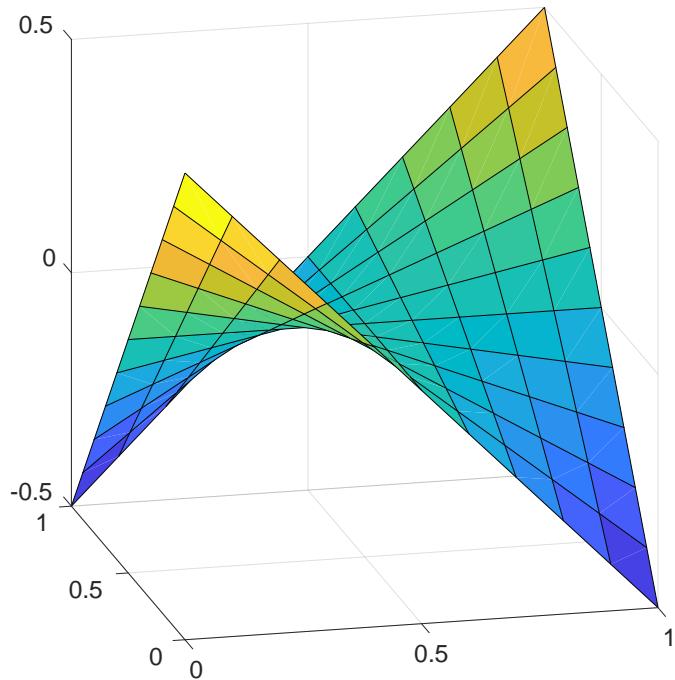
- ▶ Let $P_{0,0}, P_{1,0}, P_{0,1}, P_{1,1}$ be four points in \mathbb{R}^3 .
- ▶ These points define four line segments $P_{0,0}P_{1,0}, P_{0,1}P_{1,1}, P_{0,0}P_{0,1}, P_{1,0}P_{1,1}$.

NURBS representation of the associated bilinear interpolation surface:

$$S(u, v) = \sum_{i=0}^1 \sum_{j=0}^1 N_{i,1}(u) N_{j,1}(v) P_{i,j},$$

with $U, V = \{0, 0, 1, 1\}$. (This is a polynomial representation).

Bilinear Surfaces



Cylinders

- ▶ Given a NURBS curve of degree p , knot vector U , weights w_i

$$\mathcal{C}(u) = \sum_{i=0}^n R_{i,p}(u) P_i,$$

- ▶ and a unit vector W ,

we seek a NURBS representation of the cylinder $\mathcal{S}(u, v)$ obtained by sweeping $\mathcal{C}(u)$ along the direction W , for a distance d .

Let $0 \leq v \leq 1$ be the parameter for the sweep direction; then

- ▶ for a fixed u_0 , $\mathcal{S}(u_0, v)$ is a straight line segment from $\mathcal{C}(u_0)$ to $\mathcal{C}(u_0) + dW$,
- ▶ for a fixed v_0 , $\mathcal{S}(u, v_0)$ is a straight line segment from $\mathcal{C}(u)$ to $\mathcal{C}(u) + v_0 dW$. $d = \text{altezza del cilindro}$

$$\mathcal{S}(u, v_0) = \mathcal{C}(u) + v_0 dW = \sum_{i=0}^n R_{i,p}(u) (P_i + v_0 dW).$$

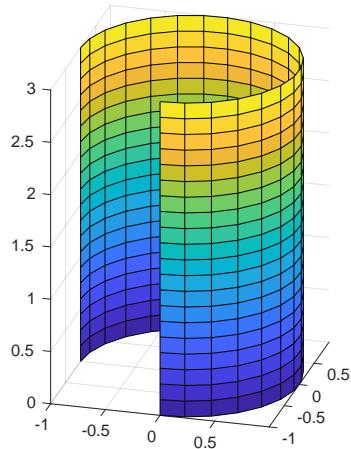
Cylinders

The desired representation is given by

$$\mathcal{S}(u, v) = \sum_{i=0}^n \sum_{j=0}^1 R_{i,p;j,1}(u, v) P_{i,j}$$

with

- ▶ knot vectors U as above, $V = \{0, 0, 1, 1\}$,
- ▶ control points $P_{i,0} = P_i$ and $P_{i,1} = P_i + dW$,
- ▶ weights $w_{i,0} = w_{i,1} = w_i$ for $i = 0, \dots, n$.



Ruled Surfaces

"interpolazione lineare"

Suppose we have two NURBS curves

$$\mathcal{C}_1(u) = \sum_{i=0}^{n_1} R_{i,p_1}(u) P_i^1,$$

$$\mathcal{C}_2(u) = \sum_{i=0}^{n_2} R_{i,p_2}(u) P_i^2,$$

defined on knot vectors U_1 and U_2 .

We seek a surface that is ruled in the v direction, i.e., that linearly interpolates $\mathcal{C}_1(u)$ and $\mathcal{C}_2(u)$ for each value of u .

Ruled surfaces

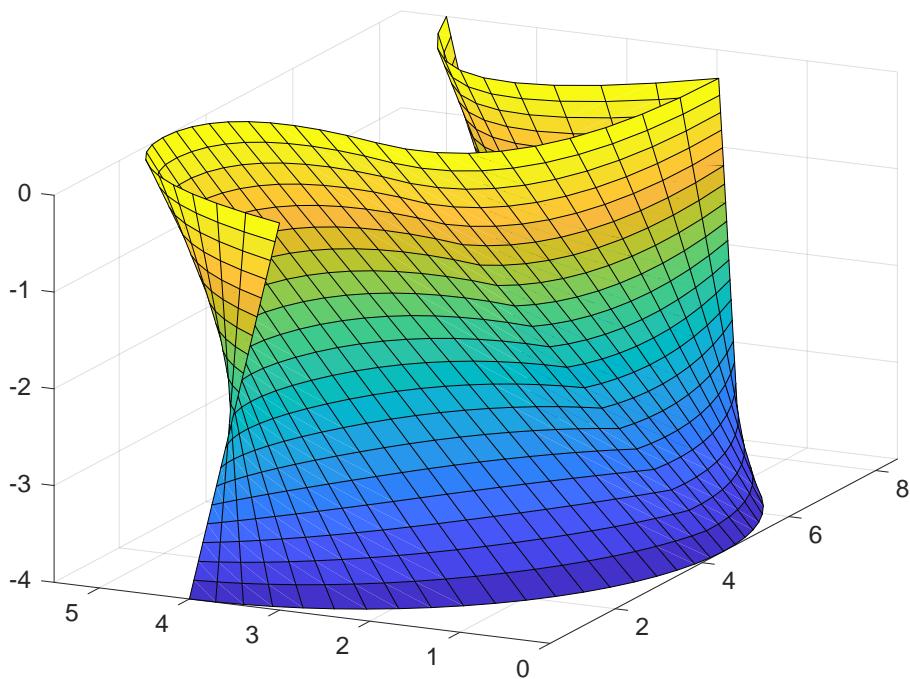
The surface has a NURBS representation

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^n R_{i,p;j,1}(u, v) P_{i,j}, \quad \text{per avere dei segmenti nella direzione } d: \text{interpolazione}$$

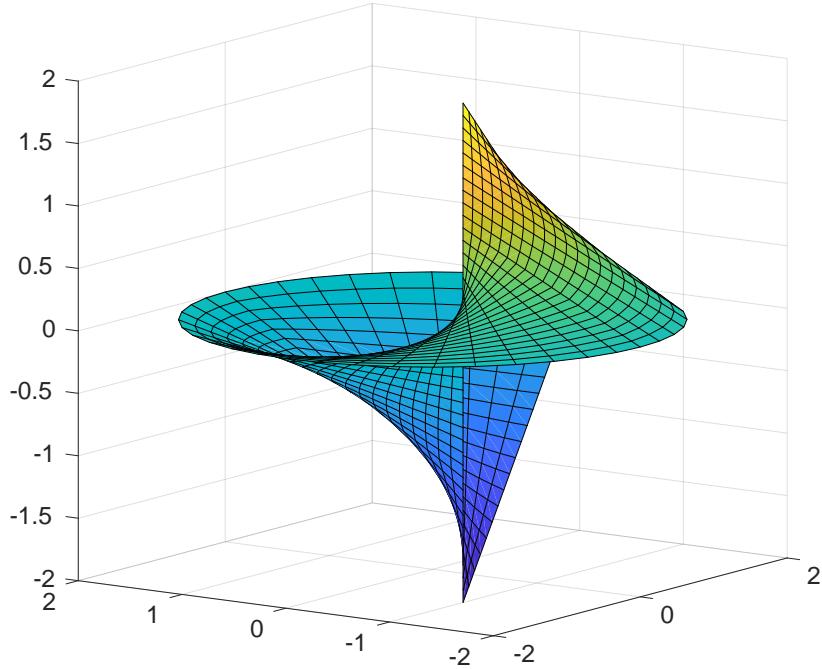
with $V = \{0, 0, 1, 1\}$. But we need to find $n, p, U, w_{i,j}$ and $P_{i,j}$.

- ▶ Ensure that $C_1(u)$ and $C_2(u)$ are defined on the same parameter range.
- ▶ Set $p = \max\{p_1, p_2\}$ and raise degree of lower degree curve to p .
- ▶ Merge U_1 and U_2 to determine U . (*tenendo le molte plicche*)
- ▶ Apply knot refinement to both curves. This step yields $n, w_{i,j}$, and $P_{i,j}$.

Ruled surfaces



Ruled surfaces



Surfaces of revolution

Let $\mathcal{C}(v) = \sum_{j=0}^m R_{j,q}(v)P_j$ be a NURBS curve of degree q on the knot vector V , with weights w_j . We seek a NURBS representation of the surface constructed by revolving $\mathcal{C}(v)$ (the **generatrix**) around an axis.

Suppose that $\mathcal{C}(v)$ lies in the (x, z) -plane, and that it is revolved about the z -axis by an angle of 2π . The resulting surface $\mathcal{S}(u, v)$ is such that

- ▶ for fixed u_0 , $\mathcal{S}(u_0, v)$ is the generatrix rotated by some angle about the z -axis,
- ▶ for fixed v_0 , $\mathcal{S}(u, v_0)$ is a full circle which lies in a plane perpendicular to the z -axis, and has its center on the z -axis.

Surfaces of revolution

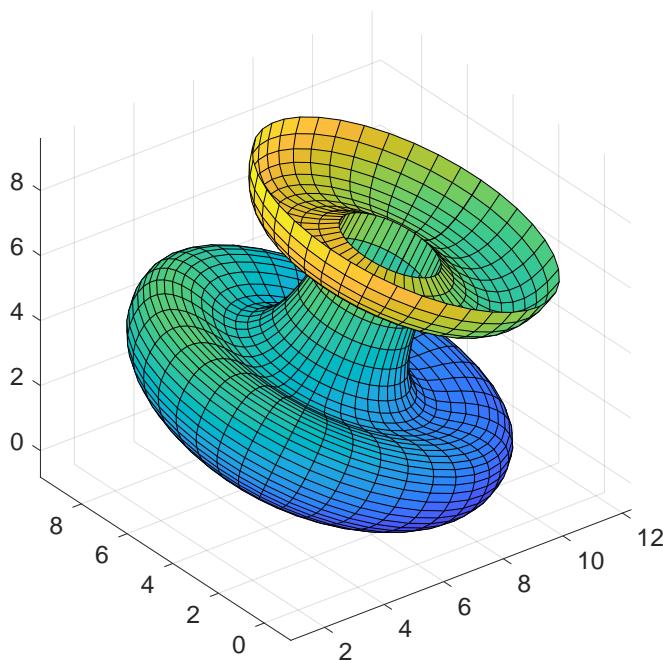
Let us use the 9-point circle representation (see exercise). Then the revolution surface can be parameterized as (proof omitted):

$$\mathcal{S}(u, v) = \sum_{i=0}^8 \sum_{j=1}^m R_{i,2;j,q}(u, v) P_{i,j},$$

where

- ▶ the control points are such that $P_{0,j} = P_j$ and, for fixed j , the $P_{i,j}$ lie in the plane $z = z_j$, on the square of width $2x_j$ and center on the z -axis,
- ▶ the knot vectors are $U = \{0, 0, 0, \frac{1}{4}, \frac{1}{4}, \frac{1}{2}, \frac{1}{2}, \frac{3}{4}, \frac{3}{4}, 1, 1, 1\}$ and V ,
- ▶ $w_{i,j} = \nu_i w_j$, where the ν_i are $\{1, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, 1, \frac{\sqrt{2}}{2}, 1\}$.

Surfaces of revolution



Esercizio Data la seguente rete di controllo in \mathbb{R}^3

$$P_{00} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad P_{01} = \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix} \quad P_{02} = \begin{pmatrix} 4 \\ 0 \\ 0 \end{pmatrix}$$

$$P_{10} = \begin{pmatrix} 0 \\ 2 \\ 0 \end{pmatrix} \quad P_{11} = \begin{pmatrix} 2 \\ 2 \\ 0 \end{pmatrix} \quad P_{12} = \begin{pmatrix} 4 \\ 2 \\ 2 \end{pmatrix}$$

$$P_{20} = \begin{pmatrix} 0 \\ 4 \\ 0 \end{pmatrix} \quad P_{21} = \begin{pmatrix} 2 \\ 4 \\ 4 \end{pmatrix} \quad P_{22} = \begin{pmatrix} 4 \\ 4 \\ 4 \end{pmatrix}$$

determinare $S(\frac{1}{2}, \frac{1}{2})$

Soluzione Applichiamo de Casteljau bivariate

$$\begin{array}{ll} P_{00} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} & P_{10} = \begin{pmatrix} 0 \\ 2 \\ 0 \end{pmatrix} \\ P_{01} = \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix} & P_{11} = \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix} \\ P_{02} = \begin{pmatrix} 4 \\ 0 \\ 0 \end{pmatrix} & P_{12} = \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix} = Q_0 \\ & \\ P_{20} = \begin{pmatrix} 0 \\ 4 \\ 0 \end{pmatrix} & P_{21} = \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix} \\ P_{21} = \begin{pmatrix} 2 \\ 4 \\ 4 \end{pmatrix} & P_{22} = \begin{pmatrix} 4 \\ 4 \\ 4 \end{pmatrix} = Q_1 \\ P_{22} = \begin{pmatrix} 4 \\ 4 \\ 4 \end{pmatrix} & \end{array}$$

$$\begin{array}{l} P_{20} = \begin{pmatrix} 0 \\ 4 \\ 0 \end{pmatrix} \\ P_{21} = \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix} \\ P_{22} = \begin{pmatrix} 4 \\ 4 \\ 4 \end{pmatrix} = Q_2 \end{array}$$

$$\begin{array}{l} Q_0 = \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix} \\ Q_1 = \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix} \\ Q_2 = \begin{pmatrix} 2 \\ 4 \\ 3 \end{pmatrix} \\ Q_0 = \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix} \\ Q_1 = \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix} \\ Q_2 = \begin{pmatrix} 2 \\ 4 \\ 3 \end{pmatrix} \\ Q_0 = \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix} \\ Q_1 = \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix} \\ Q_2 = \begin{pmatrix} 2 \\ 4 \\ 3 \end{pmatrix} = S\left(\frac{1}{2}, \frac{1}{2}\right) \end{array}$$

Osservazione Posso scrivere una superficie di Bézier (Bézier patch)

anche usando una forma matriciale

$$S(u, v) = \left[B_0^{(n)}(u), \dots, B_n^{(n)}(u) \right] \begin{pmatrix} P_{00} & \dots & P_{0m} \\ \vdots & & \vdots \\ P_{nn} & \dots & P_{nm} \end{pmatrix} \begin{pmatrix} B_0^{(m)}(v) \\ \vdots \\ B_m^{(m)}(v) \end{pmatrix}$$

La scrittura in forma matriciale può essere utile per i cambiamenti:

d: base: volendo passare dai polinomi d: Bernstein alla base monomiale basta sostituire i vettori d: polinomi d: Bernstein a vettori d: monomi moltiplicati per le matrici d: cambiamento d: base

[resto del pdf sulle superfici]

18/11

Interpolazione

Siano dati dei punti $Q_0, Q_1, \dots, Q_n \in \mathbb{R}^d$ (data points) e $p \leq n$ intero positivo. Vogliamo trovare una curva B-spline $C(t)$ di grado p che interpoli i data points nell'ordine dato, cioè $C(t_i) = Q_i$, $C(t_j) = Q_j$, $i < j \Rightarrow t_i < t_j$

Sia $U = \{t_0, \dots, t_m\}$ vettore dei nodi, $m = n + p + 1$ e scegliamo t_0, \dots, t_n t.c. $C(t_i) = Q_i$

Dobbiamo determinare i punti d: controllo

Abbiamo $C(t) = \sum_{i=0}^n N_{i,p}(t) P_i$ e le coordinate de: P_i

Sono soluzione d: un sistema lineare

Chiamiamo $P_i = (P_{i1}, \dots, P_{id})$ e $Q_i = (Q_{i1}, \dots, Q_{id})$

Ottieniamo

$$\begin{bmatrix} Q_{01} & \dots & Q_{0d} \\ \vdots & & \vdots \\ Q_{n1} & \dots & Q_{nd} \end{bmatrix} = \begin{bmatrix} N_{0,p}(t_0) & \dots & N_{n,p}(t_0) \\ \vdots & & \vdots \\ N_{0,p}(t_n) & \dots & N_{n,p}(t_n) \end{bmatrix} \begin{bmatrix} P_{01} & \dots & P_{0d} \\ \vdots & & \vdots \\ P_{n1} & \dots & P_{nd} \end{bmatrix}$$

che è risolvibile sotto opportune ipotesi che imponremo.

Vediamo ora come scegliere i parametri:

Per i t_j , abbiamo le seguenti euristiche:

- equispaziati: $t_j = \frac{j}{n}$

scelta semplice ma potenzialmente problematica se i data points sono distribuiti in modo molto disomogeneo, ex dei loop o delle gobbe per datapoint troppo vicini

- chordal length:

$$\ell := \sum_{k=1}^n |Q_k - Q_{k-1}| \quad \text{dove } |A - B| \text{ è la distanza euclidea}$$

pongo $t_0 = 0$, $t_j = \frac{|Q_j - Q_{j-1}|}{\ell} + t_{j-1}$

- centripetal length

$$\text{sia } 0 < \alpha \leq 1, \text{ definisco } \ell_\alpha := \sum_{k=1}^n |Q_k - Q_{k-1}|^\alpha$$

$$\text{pongo } t_0 = 0 \text{ e } t_j = \frac{|Q_j - Q_{j-1}|^\alpha}{\ell_\alpha} + t_{j-1}$$

Spesso si considera $\alpha = 1/2$

Per i nodi abbiamo le seguenti euristiche:

- equispaziati, $u_0 = \dots = u_p = 0$, $u_{p+j} = \frac{j}{n-p+1} \quad j = 1, \dots, n-p$
 $u_{m-p} = \dots = u_m = 1$
- medie di parametri, $u_0 = \dots = u_p = 0$

$$U_{m-p} = \dots = U_m = 1, \quad U_{p+j} = \frac{1}{p} \sum_{i=1}^{j+p-1} t_i$$

Oss Se i nodi sono scelti in questo modo, il sistema lineare per i P_i è TN e a banda con semilarghezza d: banda < p, quindi se p è piccolo si può risolvere con eliminazione di Gauss

Un'altra tecnica è il "metodo universale" in cui scegliamo i nodi equispaziati e $t_j = \arg \max N_{j,p}(t)$

Oss

- il metodo non dipende dai data points
- in pratica i t_j vengono presi come approssimazioni dei punti di max
- vale l'invarianza affine, infatti preso il sistema

$Q = NP$, data una trasformazione affine $x^T \rightsquigarrow x^T M + v^T$
e detto $u = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^d$ otteniamo

$$QM = NPM$$

$$QM + u v^T = NPM + u v^T$$

$$\frac{QM + u v^T}{\tilde{Q}} = \frac{N(PM + u v^T)}{\tilde{P}}$$

Poiché $Nu = u$

Vediamo adesso il caso d: interpolazioni d: superfici (chiamato anche interpolazione globale)

Dati:

- datapoints $Q_{ij} \in \mathbb{R}^d$ con $0 \leq i \leq n \quad 0 \leq j \leq m$
- (p, q) coppia d: grad.

vogliamo trovare una superficie B-spline $S(u, v)$ di grad. (p,q) che interpoli i data points nell'ordine dato

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) P_{ij}$$

Come prima dobbiamo determinare i vettori dei nodi, i valori dei parametri e i punti P_{ij} di controllo

Per ogn. j fissato applico un metodo unidimensionale per la scelta dei parametri a Q_{0j}, \dots, Q_{mj} e trovo parametr. $r_0^{(j)}, \dots, r_m^{(j)}$

Poi calcolo le medie al variare di j

$$r_i = \frac{1}{n+1} \sum_{j=0}^m r_i^{(j)} \quad j = 0, \dots, m$$

Con la stessa tecnica trovo i parametr. s_0, \dots, s_n nell'altra direzione

Per i punti di controllo P_{ij} imponiamo la condizione di interpolazione

$$\begin{aligned} Q_{ij} = S(r_i, s_j) &= \sum_{h=0}^n \sum_{k=0}^m N_{h,p}(r_i) N_{k,q}(s_j) P_{hk} = \\ &= \sum_{h=0}^n N_{h,p}(r_i) \underbrace{\sum_{k=0}^m N_{k,q}(s_j) P_{hk}}_{=: A_{h,j} \in \mathbb{R}^d} = \\ &= \sum_{h=0}^n N_{h,p}(r_i) A_{h,j} \end{aligned}$$

Quindi la condizione di interpolazione può essere scomposta in due problemi unidimensionali:

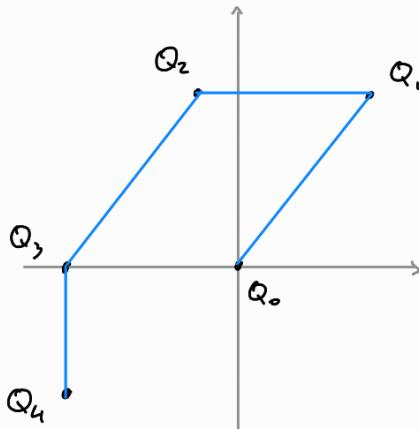
Prima risolvo per ogn. j il problema $Q_{ij} = \sum_{h=0}^n N_{h,p}(r_i) A_{h,j}$.

Not. al: $A_{h,j}$ risolvo per ogn. h il problema $A_{h,j} = \sum_{k=0}^m N_{k,q}(s_j) P_{hk}$

Ex (interpolazione di curve B-spline)

$$d=2, \quad p=3, \quad n=4$$

$$Q_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad Q_1 = \begin{pmatrix} 3 \\ 4 \end{pmatrix}, \quad Q_2 = \begin{pmatrix} -1 \\ 4 \end{pmatrix}, \quad Q_3 = \begin{pmatrix} -4 \\ 0 \end{pmatrix}, \quad Q_4 = \begin{pmatrix} -4 \\ -3 \end{pmatrix}$$



Sol Utilizziamo la chordal length per la scelta dei parametri:

$$t_0 = 0, \quad \ell = 5 + 4 + 5 + 3 = 17$$

$$t_1 = \frac{5}{17}, \quad t_2 = \frac{9}{17}, \quad t_3 = \frac{14}{17}, \quad t_4 = 1$$

Calcoliamo i nodi con il criterio della media dei parametri:

$$U_0 = U_1 = U_2 = U_3 = 0$$

$$U_4 = \frac{1}{3}(t_1 + t_2 + t_3) = \frac{1}{3} \frac{28}{17} = \frac{28}{51}$$

$$U_5 = U_6 = U_7 = U_8 = 1$$

Per determinare i punti di controllo bisogna risolvere

$$Q = NP$$

$$\text{con } N = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ N_{03}(5/17) & N_{13}(5/17) & N_{23}(5/17) & N_{33}(5/17) & 0 \\ N_{03}(9/17) & N_{13}(9/17) & N_{23}(9/17) & N_{33}(9/17) & 0 \\ 0 & N_{13}(14/17) & N_{23}(14/17) & N_{33}(14/17) & N_{43}(14/17) \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Come esercizio si può ricavare esplicitamente i P_i e verificare che la curva ottenuta interpolya i Q_i .

Sol 2 Utilizziamo il metodo universale

$$U_0 = U_1 = U_2 = U_3 = 0 \quad U_4 = \frac{1}{2}$$

$$U_5 = U_6 = U_7 = U_8 = 1$$

I t_i vengono scelti come punti di massimo delle $N_{i,p}$ e i punti di controllo si ottengono risolvendo il sistema lineare

Approssimazione globale

Dati: $Q_0, \dots, Q_n \in \mathbb{R}^d$ data points, p grado e $p < h \leq n$
cerco una curva B-spline $\mathcal{C}(t) = \sum_{i=0}^n N_{i,p}(t) P_i$ di grado p
che approssimi i data points nel senso dei minimi quadrati.

Supponiamo di aver scelto t_0, \dots, t_n e nodi U_0, \dots, U_m ($m = h + p + 1$)
e determiniamo i punti di controllo P_0, \dots, P_h che minimizzano

$$\text{la funzione } f(P_0, \dots, P_h) = \sum_{k=0}^n |Q_k - \mathcal{C}(t_k)|^2$$

$$\begin{aligned} \text{Generalmente si assume } P_0 &= \mathcal{C}(0) = Q_0, & t_0 &= 0 \\ P_h &= \mathcal{C}(1) = Q_h, & t_n &= 1 \end{aligned}$$

Quindi la funzione da minimizzare diventa

$$\begin{aligned} f(P_1, \dots, P_{h-1}) &= \sum_{k=1}^{n-1} |Q_k - \mathcal{C}(t_k)|^2 = \sum_{k=1}^{n-1} \left| Q_k - \sum_{i=0}^h N_{i,p}(t_k) P_i \right|^2 = \\ &= \sum_{k=1}^{n-1} \left| Q_k - \underbrace{N_{0p}(t_k)P_0 + N_{hp}(t_k)P_h}_{=: R_k \text{ (noto)}} - \sum_{i=1}^{h-1} N_{i,p}(t_k) P_i \right|^2 = \end{aligned}$$

$$\begin{aligned}
&= \sum_{k=1}^{n-1} \left| R_k - \sum_{i=1}^{h-1} N_{ip}(t_k) P_i \right|^2 = \\
&= \sum_{k=1}^{n-1} \left(R_k - \sum_{i=1}^{h-1} N_{ip}(t_k) P_i \right)^T \left(R_k - \sum_{i=1}^{h-1} N_{ip}(t_k) P_i \right) = \\
&= \sum_{k=1}^{n-1} \left(R_k^T R_k - 2 \sum_{i=1}^{h-1} N_{ip}(t_k) R_k^T P_i + \left(\sum_{i=1}^{h-1} \right)^T \left(\sum_{i=1}^{h-1} \right) \right)
\end{aligned}$$

Per minimizzare calcoliamo le derivate rispetto a tutti i punti di controllo e poniamo uguali a 0

$$\begin{aligned}
0 &= \frac{\partial J(P_1, \dots, P_h)}{\partial P_k} = \sum_{k=1}^{n-1} \left(-2 N_{kp}(t_k) R_k + 2 \left(\sum_{i=1}^{h-1} N_{ip}(t_k) P_i \right) N_{kp}(t_k) \right) \\
\Rightarrow \underbrace{\sum_{k=1}^{n-1} N_{kp}(t_k) R_k}_{=: \tilde{R}_p \text{ noto}} &= \sum_{k=1}^{n-1} N_{kp}(t_k) \left(\sum_{i=1}^{h-1} N_{ip}(t_k) P_i \right)
\end{aligned}$$

Questa condizione si riscrive come un sistema lineare (per ex) della forma $(N^T N) P = R$

$$\text{dove } P = \begin{bmatrix} P_{1,1} & \dots & P_{1,d} \\ \vdots & & \vdots \\ P_{h-1,1} & \dots & P_{h-1,d} \end{bmatrix} = \begin{bmatrix} P_1 \\ \vdots \\ P_{h-1} \end{bmatrix}, \quad R = \begin{bmatrix} \tilde{R}_1 \\ \vdots \\ \tilde{R}_{n-1} \end{bmatrix}$$

$$N = \begin{bmatrix} N_{1,p}(t_1) & \dots & N_{h-1,p}(t_1) \\ \vdots & & \vdots \\ N_{1,p}(t_{n-1}) & \dots & N_{h-1,p}(t_{n-1}) \end{bmatrix} \in \mathbb{R}^{(n-1) \times (h-1)} \Rightarrow N^T N \in \mathbb{R}^{(h-1) \times (h-1)}$$

Se i nodi sono scelti con il metodo delle medie, $N^T N$ è a banda, simmetrica e definita pos

Per quanto riguarda l'approssimazione globale per superficie, dati datapoints Q_{ij} , $0 \leq i \leq n$, $0 \leq j \leq m$, fissati grad: (p, q) e fissati (h, k) con $p < h \leq n$, $q < k \leq m$, supponiamo di aver scelto i parametri $\{r_\alpha\}_{\alpha=0,\dots,h}$ e $\{s_\beta\}_{\beta=0,\dots,k}$ e vettori dei nodi $U \in V$ cerchiamo punti di controllo $\{P_{\alpha\beta}\}_{\substack{\alpha=0,\dots,h \\ \beta=0,\dots,k}}$ che definiscono la superficie

$$S(u, v) = \sum_{\alpha=0}^h \sum_{\beta=0}^k N_{\alpha, p}(u) N_{\beta, q}(v) P_{\alpha\beta}$$

in modo tale da minimizzare la quantità

$$\mathcal{J}(P) = \sum_{i=0}^m \sum_{j=0}^n |Q_{ij} - S(r_i, s_j)|^2$$

Possibili approcci:

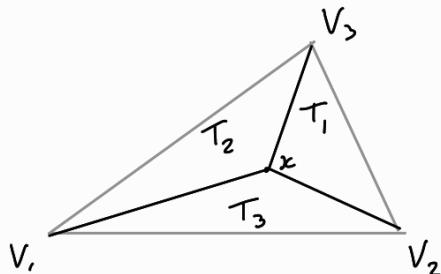
- calcolo le derivate e le pongo uguali a 0
- (soluzione approssimata) spezzo il problema bivariato in problemi univariati come per l'interpolazione

Patch triangolare di Bézier

Useremo i polinomi di Bernstein bivariati lavorando in coordinate barientriche. Definiamo quindi le coordinate barientriche

Sia T un triangolo di vertici V_1, V_2 e V_3 non degenere

Sia X un punto nel piano (non necessariamente nel triangolo)



con T , triangolo di vertici X, V_2, V_3

T_2 V_1, X, V_3

T_3 V_1, V_2, X

Allora esistono e sono unici: $\tau_1, \tau_2, \tau_3 \in \mathbb{R}$ (coordinate baricentriche di X) tali che $\tau_1 + \tau_2 + \tau_3 = 1$ e $X = \tau_1 V_1 + \tau_2 V_2 + \tau_3 V_3$

In particolare se x è interno al triangolo T ha coordinate baricentriche positive. Vale $V_1 = (1, 0, 0)$, $V_2 = (0, 1, 0)$ e $V_3 = (0, 0, 1)$

Geometricamente $\tau_i = \frac{\text{area } T_i}{\text{area } T}$ (aree con segno)

Infatti le due condizioni sui τ_i si possono riscrivere come

$$\begin{bmatrix} 1 & 1 & 1 \\ V_{1x} & V_{2x} & V_{3x} \\ V_{1y} & V_{2y} & V_{3y} \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} 1 \\ X_x \\ X_y \end{bmatrix} \quad \begin{array}{l} \text{dove } X = (X_x, X_y) \\ V_i = (V_{ix}, V_{iy}) \end{array}$$

coordinate cartesiane

Risolvendo con Cramer abbiamo la condizione sulle aree

Def Fissati p intero positivo e T come sopra, definiamo i

POLINOMI DI BERNSTEIN TRIANGOLARI come

$$B_{ijk}^{(p)}(X) = \frac{p!}{i! j! k!} \tau_1^i \tau_2^j \tau_3^k \quad \text{con } i+j+k=p, \quad X = (\tau_1, \tau_2, \tau_3)$$

coordinate baricentriche

Equivalentemente

$$B_{ij}^{(p)}(X) = \frac{p!}{i! j! (p-i-j)!} \tau_1^i \tau_2^j (1-\tau_1-\tau_2)^{(p-i-j)}$$

Es ($p=2$)

$$B_{2,0,0}^{(2)}(X) = \tau_1^2 \quad B_{1,1,0}^{(2)}(X) = 2\tau_1 \tau_2$$

$$B_{0,2,0}^{(2)}(X) = \tau_2^2 \quad B_{1,0,1}^{(2)}(X) = 2\tau_1 \tau_3$$

$$B_{0,0,2}^{(2)}(X) = \tau_3^2 \quad B_{0,1,1}^{(2)}(X) = 2\tau_2 \tau_3$$

Proprietà

- I polinomi di Bernstein bivariati sono una base dei polinomi di grado $\leq p$

Dim Cominciamo contando i $B_{i,j,k}^{(p)}$

Per contarli dobbiamo contare in quanti modi possiamo scegliere i, j e k tali che $i+j+k=p$

Con stars and bars ottieniamo $\binom{p+2}{2}$ modi, quindi i $B_{i,j,k}^{(p)}$ sono $\binom{p+2}{2}$ che è esattamente la dimensione dello spazio

Per l'indipendenza lineare consideriamo i polinomi con K e T_3 impliciti, e chiamiamo $U = \frac{T_1}{(1-T_1-T_2)}$ e $V = \frac{T_2}{(1-T_1-T_2)}$
abbiamo $B_{i,j,k}^{(p)}(x) = \frac{p!}{i! j! (p-i-j)!} U^i V^j (1-U-V)^p$

Se io avessi una combinazione lineare nulla

$$\sum_{\substack{i,j \\ i+j \leq p}} C_{i,j} B_{i,j}^{(p)}(x) = 0$$

$$\Rightarrow \sum_{\substack{i,j \\ i+j \leq p}} C_{i,j} \frac{p!}{i! j! (p-i-j)!} U^i V^j = 0$$

e $C_{i,j}=0 \forall i,j$ perché combinazione lineare nulla

di un riscalamento della base canonica

□

- Non-negatività nel triangolo di riferimento
- Partizione dell'unità (sviluppo del trinomio o proprietà delle aree)

- Ricorrenza:

$$B_{i,j,k}^{(p)}(x) = \tau_1 B_{i-1,j,k}^{(p-1)}(x) + \tau_2 B_{i,j-1,k}^{(p-1)}(x) + \tau_3 B_{i,j,k-1}^{(p-1)}(x)$$

- Elevamento del grado

$$\begin{aligned} B_{i,j,k}^{(p)}(x) &= (\tau_1 + \tau_2 + \tau_3) B_{i,j,k}^{(p)}(x) = \\ &= \frac{i+1}{p+1} B_{i+1,j,k}^{(p+1)}(x) + \frac{j+1}{p+1} B_{i,j+1,k}^{(p+1)}(x) + \frac{k+1}{p+1} B_{i,j,k+1}^{(p+1)}(x) \end{aligned}$$

- Derivate direzionali:

le derivate sono direzionali rispetto ad un vettore dato da $X-Y$ (da spazio affine a spazio vettoriale) avente coordinate che sommano a 0. Detto $u = (\delta_1, \delta_2, \delta_3)$ con $\delta_1 + \delta_2 + \delta_3 = 0$

$$D_u[B_{i,j,k}^{(p)}(x)] = p \left[\delta_1 B_{i-1,j,k}^{(p-1)}(x) + \delta_2 B_{i,j-1,k}^{(p-1)}(x) + \delta_3 B_{i,j,k-1}^{(p-1)}(x) \right]$$

Inoltre

$$D_u^{(r)}[B_{i,j,k}^{(p)}(V_1)] = 0 \quad \text{per } r = 1, \dots, p-i-1$$

$$D_u^{(r)}[B_{i,j,k}^{(p)}(V_2)] = 0 \quad \text{per } r = 1, \dots, p-j-1$$

$$D_u^{(r)}[B_{i,j,k}^{(p)}(V_3)] = 0 \quad \text{per } r = 1, \dots, p-k-1$$

- Caratterizzazione dei punti di massimo

$$\left[D_u^{(r)}[B_{i,j,k}^{(p)}(x)] = 0 \quad \forall u \right] \Leftrightarrow x = \frac{i V_1 + j V_2 + k V_3}{p} =: \sum_i i V_i$$

Dim Per la formula della derivata, $\forall u = (\delta_1, \delta_2, \delta_3)$ t.c. $\sum \delta_i = 0$

$$\begin{aligned} 0 = D_u^{(r)}[B_{i,j,k}^{(p)}(x)] &= \delta_1 \frac{(p-1)!}{(i-1)!j!k!} \tau_1^{i-1} \tau_2^j \tau_3^k + \\ &+ \delta_2 \frac{(p-1)!}{i!(j-1)!k!} \tau_1^i \tau_2^{j-1} \tau_3^k + \delta_3 \frac{(p-1)!}{i!j!(k-1)!} \tau_1^i \tau_2^j \tau_3^{k-1} \end{aligned}$$

Quindi semplificando, $\forall u = (\delta_1, \delta_2, \delta_3)$ t.c. $\sum \delta_i = 0$

$$\frac{\delta_1}{jk} T_2 T_3 + \frac{\delta_2}{ik} T_1 T_3 + \frac{\delta_3}{ij} T_1 T_2 = 0$$

da cui si ottiene semplicemente la tesi:

□

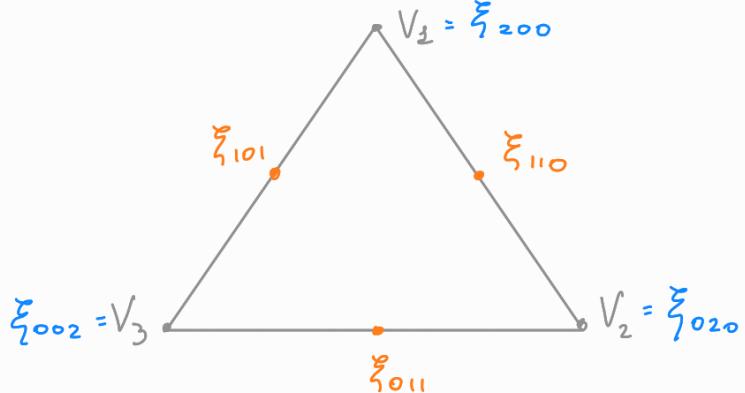
Le ξ_{ijk} sono le equivalenti delle ascisse di Greville

Per $p=2$ abbiamo

$$\xi_{200} = V_1 \quad \xi_{110} = \frac{V_1 + V_2}{2}$$

$$\xi_{020} = V_2 \quad \xi_{101} = \frac{V_1 + V_3}{2}$$

$$\xi_{002} = V_3 \quad \xi_{011} = \frac{V_2 + V_3}{2}$$



Superficie di Bézier triangolare

Fissato un grado p e punti di controllo P_{ijk} definiamo la superficie di Bézier triangolare da essa definita come

$$S(x) = \sum_{\substack{i,j,k \\ i+j+k=p}} B_{ijk}^{(p)}(x) P_{ijk}$$

Per le curve di Bézier sappiamo che possiamo sempre rappresentare il grafico di un polinomio come curva di Bézier

Analogamente, dato un polinomio $q(x) = \sum_{i+j+k=p} q_{ijk} B_{ijk}^{(p)}(x)$, possiamo rappresentare il suo grafico come superficie di Bézier

Proprietà

- Inviluppo convesso

- Invarianza affine
- Algoritmo di de Casteljau
per ricorrenza vale

$$\begin{aligned}
 S(x) &= \sum_{i+j+k=p} B_{ijk}^{(p)}(x) P_{ijk}^{(0)} = \\
 &= \sum_{i+j+k=p} \left(\tau_1 B_{i-1,j,k}^{(p-1)}(x) + \tau_2 \dots + \tau_3 \dots \right) P_{ijk}^{(0)} = \\
 &\text{solito tricck d: termini nulli e shift} \\
 &= \sum_{i+j+k=p-1} B_{ijk}^{(p-1)}(x) P_{ijk}^{(1)}
 \end{aligned}$$

con $P_{ijk}^{(1)} = \tau_1 P_{i+1,j,k}^{(0)} + \tau_2 P_{i,j+1,k}^{(0)} + \tau_3 P_{i,j,k+1}^{(0)}$

Iterando riusciamo a calcolare esplicitamente $S(x)$

- Elevamento a potenza

$$\begin{aligned}
 S(x) &= \sum_{i+j+k=p} B_{ijk}^{(p)}(x) P_{ijk}^{(0)} = \sum_{i+j+k=p+1} B_{ijk}^{(p+1)}(x) \hat{P}_{ijk}^{(0)} \\
 \text{con } \hat{P}_{ijk}^{(0)} &= \frac{i}{p+1} P_{i-1,j,k}^{(0)} + \frac{j}{p+1} P_{i,j-1,k}^{(0)} + \frac{k}{p+1} P_{i,j,k-1}^{(0)}
 \end{aligned}$$

- Derivate: preso $u = (\delta_1, \delta_2, \delta_3)$ con $\sum \delta_i = 0$ vale

$$D_u [S(x)] = p \sum_{i+j+k=p-1} \left(\delta_1 P_{i+1,j,k}^{(0)} + \delta_2 P_{i,j+1,k}^{(0)} + \delta_3 P_{i,j,k+1}^{(0)} \right) B_{ijk}^{(p-1)}(x)$$

Nel vertice abbiamo

$$D_u [S(V_1)] = p \left(\delta_1 P_{p,0,0}^{(0)} + \delta_2 P_{p-1,1,0}^{(0)} + \delta_3 P_{p-1,0,1}^{(0)} \right)$$

e analogamente per gli altri vertici

Raccordi di patch di Bézier triangolare

Consideriamo due triangoli adiacenti:

e due superfici definite su T e \tilde{T} che

senza perdita di generalità possiamo supporre avere lo stesso grado

$$S(x) = \sum_{i+j+k=p} B_{ijk}^{(p)}(x) P_{ijk}$$

$$\tilde{S}(x) = \sum_{i+j+k=p} B_{ijk}^{(p)}(x) \tilde{P}_{ijk}$$

dove indicano $X = (x_1, x_2, x_3)$ le coordinate baricentriche rispetto a T e $\tilde{X} = (\tilde{x}_1, \tilde{x}_2, \tilde{x}_3)$ le coordinate baricentriche rispetto a \tilde{T}

Se X appartiene al segmento V_1V_2 vale $X = (x_1, x_2, 0) = (\tilde{x}_1, \tilde{x}_2, 0)$

quindi $x_1 = \tilde{x}_1$ e $x_2 = \tilde{x}_2$. Allora

$$S(x) = \sum_{i+j=p} B_{ij0}^{(p)}(x) P_{ij0} = \sum_{i+j=p} \frac{p!}{i!j!} x_1^i x_2^j P_{ij0}$$

$$\tilde{S}(x) = \sum_{i+j=p} B_{ij0}^{(p)}(\tilde{x}) \tilde{P}_{ij0} = \sum_{i+j=p} \frac{p!}{i!j!} \tilde{x}_1^i \tilde{x}_2^j \tilde{P}_{ij0} = \sum_{i+j=p} \frac{p!}{i!j!} x_1^i x_2^j \tilde{P}_{ij0}$$

Imporre $X \in V_1V_2 \Rightarrow S(x) = \tilde{S}(x)$ (ovvero la condizione di rapporto C^0)

vuo d: re imporre $P_{ij0} = \tilde{P}_{ij0} \quad \forall i, j \text{ t.c. } i+j=p$

1/12

Determiniamo le condizioni per il raccordo C^1 , ovvero per cui:

valga che ∇ trasversale a V_1V_2 (cioè $\nabla \neq 0$ e ∇ non $\parallel V_1V_2$)

s: abb: a $D_\nu S(x) = D_\nu \tilde{S}(x) \quad \forall x \in V_1V_2$

Ricordiamo la formula (detto $\nu = (\delta_1, \delta_2, \delta_3)$ dove $\delta_1 + \delta_2 + \delta_3 = 0$)

$$D_\nu S(x) = p \sum_{i+j+k=p-1} (\delta_1 P_{i+1,j,k} + \delta_2 P_{i,j+1,k} + \delta_3 P_{i,j,k+1}) B_{ijk}^{(p-1)}(x)$$

Sia $v = (\delta_1, \delta_2, \delta_3) = (\tilde{\delta}_1, \tilde{\delta}_2, \tilde{\delta}_4)$ rispetto a T e \tilde{T}

e $X \in V_1 V_2$ da cui $X = (x_1, x_2, 0)$ rispetto a T e \tilde{T} (cioè $x_1 = \tilde{x}_1$, $x_2 = \tilde{x}_2$)

Oss Poiché la condizione di raccordo C^1 è controllata solo su punti X con coordinate $x_3 = \tilde{x}_4 = 0$ quindi tutti i polinomi con $k > 0$ si annulleranno (contenendo la terza coordinata), quindi possiamo restringere la sommatoria ai termini con $k=0$

Possiamo riscrivere la formula della derivata nel seguente modo:

$$D_v S(X) = p \sum_{i+j=p-1} (\delta_1 P_{i+1,j,0} + \delta_2 P_{i,j+1,0} + \delta_3 P_{i,j,1}) \frac{(p-1)!}{i! j!} x_1^i x_2^j$$

$$D_v \tilde{S}(X) = p \sum_{i+j=p-1} (\tilde{\delta}_1 P_{i+1,j,0} + \tilde{\delta}_2 P_{i,j+1,0} + \tilde{\delta}_4 \tilde{P}_{i,j,1}) \frac{(p-1)!}{i! j!} x_1^i x_2^j$$

$\hat{\quad}$ utilizzando $\tilde{P}_{i,j,0} = P_{i,j,0}$ per continuità C^0

Chiedere $D_v S(X) = D_v \tilde{S}(X)$ è equivalente a chiedere che tutti i termini della sommatoria coincidano, quindi:

$$\forall i+j=p-1 \quad (\delta_1 - \tilde{\delta}_1) P_{i+1,j,0} + (\delta_2 - \tilde{\delta}_2) P_{i,j+1,0} + \delta_3 P_{i,j,1} + \tilde{\delta}_4 \tilde{P}_{i,j,1} = 0$$

Osserviamo che $\delta_3 \neq 0$ altrimenti v sarebbe parallelo a $V_1 V_2$, e analogamente $\tilde{\delta}_4 \neq 0$. Allora otteniamo

$$\frac{\delta_1 - \tilde{\delta}_1}{\tilde{\delta}_4} P_{i+1,j,0} + \frac{\delta_2 - \tilde{\delta}_2}{\tilde{\delta}_4} P_{i,j+1,0} + \frac{\delta_3}{\tilde{\delta}_4} P_{i,j,1} = \tilde{P}_{i,j,1}$$

$\mu_1 \qquad \mu_2 \qquad \mu_3$

In particolare poiché $v = \delta_1 V_1 + \delta_2 V_2 + \delta_3 V_3 = \tilde{\delta}_1 V_1 + \tilde{\delta}_2 V_2 + \tilde{\delta}_4 V_4$ quindi (μ_1, μ_2, μ_3) sono le coordinate barientriche di V_4

rispetto a T . La condizione di raccordo C^1 è quindi:

$$\tilde{P}_{i,j,\pm} = \mu_1 P_{i+1,j,0} + \mu_2 P_{i,j+1,0} + \mu_3 P_{i,j,\pm} \quad \forall i+j=p-1$$

A parole, la condizione C^0 ci impone come scegliere i punti $\tilde{P}_{i,j,0}$, la condizione C^1 ci impone come scegliere i punti $\tilde{P}_{i,j,\pm}$

Ex Consideriamo la patch quadratica data dai seguenti punti:

$$P_{020} = \begin{pmatrix} 0 \\ 6 \\ 0 \end{pmatrix}$$

$$P_{011} = \begin{pmatrix} 0 \\ 3 \\ 0 \end{pmatrix} \quad P_{110} = \begin{pmatrix} 3 \\ 3 \\ 6 \end{pmatrix}$$

$$P_{002} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad P_{101} = \begin{pmatrix} 3 \\ 0 \\ 0 \end{pmatrix} \quad P_{200} = \begin{pmatrix} 6 \\ 0 \\ 0 \end{pmatrix}$$

- Se a $X = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, calcoliamo $S(X)$ usando de Casteljau

$$P_{100}^{(1)} = \frac{1}{3} P_{200}^{(0)} + \frac{1}{3} P_{110}^{(0)} + \frac{1}{3} P_{101}^{(0)} = \begin{pmatrix} 4 \\ 1 \\ 5 \end{pmatrix}$$

$$P_{010}^{(1)} = \frac{1}{3} P_{110}^{(0)} + \frac{1}{3} P_{020}^{(0)} + \frac{1}{3} P_{011}^{(0)} = \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix}$$

$$P_{001}^{(1)} = \frac{1}{3} P_{101}^{(0)} + \frac{1}{3} P_{011}^{(0)} + \frac{1}{3} P_{002}^{(0)} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

$$P_{000}^{(2)} = \frac{1}{3} P_{100}^{(1)} + \frac{1}{3} P_{010}^{(1)} + \frac{1}{3} P_{001}^{(1)} = \begin{pmatrix} 2 \\ 2 \\ \frac{7}{3} \end{pmatrix}$$

- Estendiamo la superficie per mezzo di un'altra patch

Consideriamo (ad esempio) $V_4 = \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix}$ in coordinate barientriche

rispetto al triangolo T su cui è definita la prima patch

Applicando le formule viste per i punti di controllo della seconda patch otteniamo:

$$C^0 \Rightarrow \tilde{P}_{200} = P_{200}, \quad \tilde{P}_{110} = P_{110}, \quad \tilde{P}_{020} = P_{020}$$

$$C^1 \Rightarrow \tilde{P}_{101} = P_{200} + P_{110} - P_{101} = \begin{pmatrix} 6 \\ 3 \\ 15 \end{pmatrix}$$

$$\tilde{P}_{011} = P_{110} + P_{020} - P_{011} = \begin{pmatrix} 3 \\ 6 \\ 6 \end{pmatrix}$$

Il punto \tilde{P}_{002} può essere scelto a piacere

Oss questo formalismo si estende alle spline (Powell-Sabin) e alle forme razionali (NURPS)

Ad esempio, la versione razionale di una patch di Bezier

ha la forma

$$R(x) = \frac{\sum_{i+j+k=p} w_{ijk} P_{ijk} B_{ijk}^{(p)}(x)}{\sum_{i+j+k=p} w_{ijk} B_{ijk}^{(p)}(x)}$$

Metodi di suddivisione

Vogliamo presentare un metodo alternativo di presentazione delle curve

Es (metodo di Chaikin - George Chaikin 1974)

Supponiamo di avere una successione di punti $\{P_i\}_{i=0}^n$

nel piano che collegiamo con una spezzata

Su ogni segmento $P_i P_{i+1}$ definiamo due nuovi punti:

$$Q_i = \frac{3}{4} P_i + \frac{1}{4} P_{i+1} \quad \text{e} \quad R_i = \frac{1}{4} P_i + \frac{3}{4} P_{i+1}$$

Considero come nuova successione di punti quella ottenuta interlacciando le successioni $\{Q_i\}_{i=0}^{n-1}$ e $\{R_i\}_{i=0}^{n-1}$, ovvero

$$Q_0, R_0, Q_1, R_1, Q_2, R_2, \dots, Q_{n-1}, R_{n-1}$$

Posso iterare questo procedimento e si dimostra che la curva ottenuta al limite è la curva B-spline di grado 2 con punti di controllo $\{P_i\}_{i=0}^n$ e nodi uniformi.

All'atto pratico (es: rappresentazione di un'approssimazione della curva su uno schermo con numero finito di pixel) il procedimento converge molto velocemente ad approssimazioni decenti.

Dimostriamo che la curva al limite è proprio la B-spline.

Dm Consideriamo la curva B-spline quadratica con punti di controllo $\{P_0, \dots, P_n\}$ e nodi equispaziali $\{u_j\}_{j=0, \dots, m}$.

Sia zh l'ampiezza di ciascun knot span e inseriamo un nuovo nodo nel punto medio di ciascun knot span, applicando le formule già viste per calcolare i nuovi punti di controllo (normalmente stiamo facendo "knot insertion" ripetute)

$$Q_i = (1-\alpha_i)P_{i-1} + \alpha_i P_i \quad \text{dove} \quad \alpha_i = \frac{\tilde{u} - u_i}{u_{i+p} - u_i}, \quad k-p+1 \leq i \leq k$$

↑
nuovo punto di controllo
dopo la knot insertion di \tilde{u} nel knot span $[u_k, u_{k+1}]$

Nel nostro caso con $p=2$, per ogni nuovo nodo \tilde{u} inserito devo inserire due nuovi punti di controllo ($k-1 \leq i \leq k$)

Per $i=k$ avremo $\alpha_k = \frac{\tilde{u} - u_k}{u_{k+2} - u_k} = \frac{h}{4h} = \frac{1}{4}$ (avendo scelto \tilde{u} come punto medio di $[u_k, u_{k+1}]$ ed essendo tuttavia knot span amp: zh)

$$\text{Per } i=k-1 \text{ avremo } \alpha_{k-1} = \frac{\tilde{u} - u_{k-1}}{u_{k+1} - u_{k-1}} = \frac{3h}{4h} = \frac{3}{4}$$

Allora abbiamo

$$Q_k = \frac{3}{4} P_{k-1} + \frac{1}{4} P_k, \quad Q_{k-1} = \frac{1}{4} P_{k-2} + \frac{3}{4} P_{k-1}$$

Quindi: l'inserimento di un nuovo nodo nel punto medio di ogni knot span equivale ad un passo del metodo di Chaikin

□

2/12

Ri prendiamo il problema della suddivisione di curve. Vogliamo suddividere le **B-spline cardinali**, ovvero le B-spline in cui tutti i nodi (anche gli estremi che non avranno più molteplicità) sono equispazati. Definiamo quindi:

$$B_0(t) = \begin{cases} 1 & \text{se } t \in [0, 1] \\ 0 & \text{altr.ment.} \end{cases}, \quad B_0^{(k)}(t) = B_0(t-k), \quad k \in \mathbb{Z}$$

convoluzione →

$$\text{Definiamo poi: } B_1(t) = B_0(t) * B_0(t), \text{ ovvero } B_1(t) = \int_{\mathbb{R}} B_0(s) B_0(t-s) ds$$

$$\text{e analogamente } B_1^{(k)}(t) = B_1(t-k)$$

$$\text{In generale } B_\ell(t) = \underbrace{B_0(t) * \dots * B_0(t)}_{\ell+1 \text{ fattori}} = *^\ell B_0(t) = B_0(t) * B_{\ell-1}(t)$$

$$B_\ell^{(k)}(t) = B_\ell(t-k)$$

$B_\ell(t)$ ha regolarità $C^{\ell-1}$

Formula di raffinamento (o suddivisione) per B-spline

$$B_\ell(t) = \frac{1}{2^\ell} \sum_{k=0}^{\ell+1} \binom{\ell+1}{k} B_\ell(2t-k) \quad \ell \geq 0$$

Dim per induzione (forte) su ℓ

$\ell=0$ Per il caso base la tesi è $B_0(t) = B_0(2t) + B_0(2t-1)$

che segue osservando che $B_0(t) = \mathbb{1}_{[0,1]}(t)$,

$$B_0(2t) = \mathbb{1}_{[0,1]}(2t) = \mathbb{1}_{[0,1/2]}(t) \quad \text{e} \quad B_0(2t-1) = \mathbb{1}_{[1/2,1]}(t)$$

$\ell \Rightarrow \ell+1$ Per definizione abbiamo $B_\ell(t) = *^\ell B_0(t)$

Per ipotesi induktiva possiamo sostituire $B_0(t)$

Utilizziamo ora le seguenti proprietà della convoluzione:

1] linearità:

2] commutatività

3] time scaling & time shift

$$f(2t) * g(2t) = \frac{1}{2}(f * g)(2t)$$

$$f(t-i) * g(t-j) = (f * g)(t-i-j)$$

Allora

$$\begin{aligned} B_\ell(t) &= *^\ell B_0(t) = *^\ell (B_0(2t) + B_0(2t-1)) \\ &= \sum_{k=0}^{\ell+1} \binom{\ell+1}{k} \underbrace{(B_0(2t) * \dots * B_0(2t))}_{\ell+1-k \text{ fattori}} * \underbrace{(B_0(2t-1) * \dots * B_0(2t-1))}_{k \text{ fattori}} = \\ &= \sum_{k=0}^{\ell+1} \binom{\ell+1}{k} \underbrace{\frac{1}{2^{\ell-k}} B_{\ell-k}(2t)}_{\substack{\uparrow \\ \text{fattori}}} * \underbrace{\frac{1}{2^{k-1}} B_{k-1}(2t-k)}_{\substack{\uparrow \\ \text{fattori}}} = \\ &= \frac{1}{2^{\ell-1}} \sum_{k=0}^{\ell+1} \binom{\ell+1}{k} \frac{1}{2} B_\ell(2t-k) \\ &= \frac{1}{2^\ell} \sum_{k=0}^{\ell+1} \binom{\ell+1}{k} B_\ell(2t-k) \end{aligned}$$

□

Supponiamo di aver fissato dei punti di controllo $\{P_i\}_{i \in I}$

(l'ambiguità su I è voluta per non perdere di generalità, all'atto pratico sarà o \mathbb{Z} o un sottoinsieme finito)

Definiamo la curva $C(t) = \sum_i P_i B_\ell^{(i)}(t)$

Vettorialmente possiamo scrivere $C(t) = [B_e(t)]^T P$ con $P = [P_i]_{i \in I}$
e $[B_e(t)] = [B_e^{(i)}(t)]_{i \in I}$

Sia $s_k = \frac{1}{2^k} \binom{l+1}{k}$ per l fissato, allora possiamo riscrivere la formula di raffinamento come

$$[B_e(t)]^T = [B_e(2t)]^T S, \quad S =$$

La matrice S è detta **MATRICE GLOBALE**

DI **SUDDIVISIONE**, è stocastica per colonne e infinita ma ha la proprietà

di finite column height, ovvero a colonne con un numero finito di entrate con "altezza" (della parte non nulla) delle colonne limitata

$$S = \begin{bmatrix} \vdots & \vdots & \vdots \\ 0 & 0 & 0 \\ s_0 & 0 & 0 \\ s_1 & 0 & 0 \\ s_2 & s_0 & 0 \\ s_3 & s_1 & 0 \\ s_4 & s_2 & s_0 \\ \vdots & \vdots & \vdots \\ s_{e-1} & s_{e-1} & s_{e-3} \\ 0 & s_e & s_{e-2} \\ 0 & s_{e+1} & s_{e-1} \\ 0 & 0 & s_e \\ 0 & 0 & s_{e+1} \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \end{bmatrix}$$

Possiamo quindi scrivere $C(t) = [B_e(2t)]^T SP$, quindì SP fornisce i nuovi punti di controllo dopo un passo di raffinamento

Abbiamo quindi $P^{(0)} = P$ e $P^{(j)} = SP^{(j-1)} = S^j P$

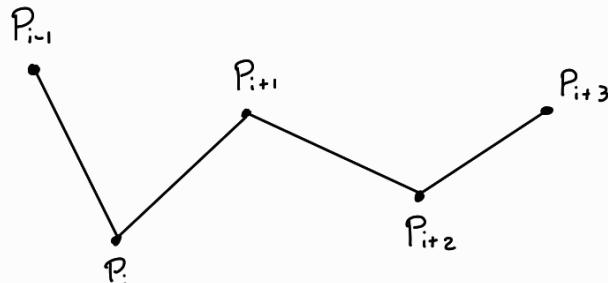
Per via della forma delle righe di S abbiamo due formule di suddivisione per i punti di controllo

$$P_{2i+1}^{(j+1)} = \sum_h s_{2i+1,h} P_h^{(j)} = \sum_h s_{2(i-h)+1} P_h^{(j)}$$

$$P_{2i}^{(j+1)} = \sum_h s_{2i,h} P_h^{(j)} = \sum_h s_{2(i-h)} P_h^{(j)}$$

Es

$$\boxed{l=1}$$

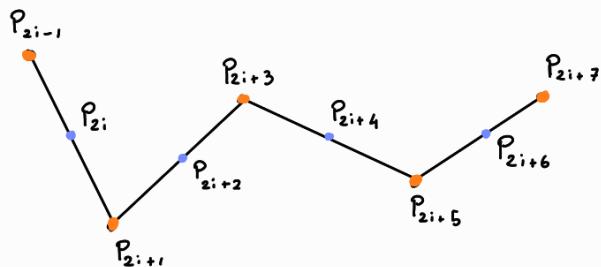


$$S_k = \frac{1}{2} \binom{2}{k} \quad \text{con } k=0,1,2 \quad \text{quindi: } S_0 = \frac{1}{2}, \quad S_1 = 1 \quad \text{e} \quad S_2 = \frac{1}{2}$$

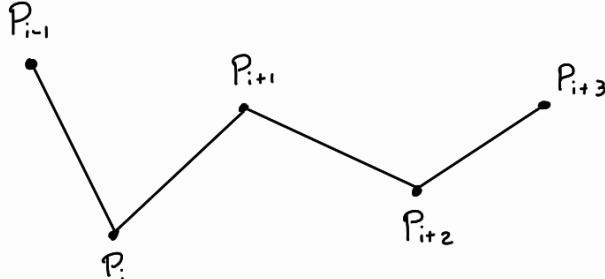
Le due formule sono

| $P_{2i}^{(j+1)} = \frac{1}{2} P_i^{(j)} + \frac{1}{2} P_{i-1}^{(j)}$

| $P_{2i+1}^{(j+1)} = P_i^{(j)}$



$\ell=3$



$$S_k = \frac{1}{8} \binom{4}{k}$$

$$S_0 = \frac{1}{8} \quad S_1 = \frac{4}{8}$$

$$S_2 = \frac{6}{8} \quad S_3 = \frac{4}{8}$$

$$S_4 = \frac{1}{8}$$

Le formule diventano

| $P_{2i}^{(j+1)} = \frac{1}{8} P_i^{(j)} + \frac{6}{8} P_{i+1}^{(j)} + \frac{1}{8} P_{i+2}^{(j)}$

| $P_{2i+1}^{(j+1)} = \frac{4}{8} P_i^{(j)} + \frac{4}{8} P_{i+1}^{(j)}$



$\ell=2$

$$S_k = \frac{1}{4} \binom{3}{k}, \quad \text{cioè} \quad S_0 = \frac{1}{4}, \quad S_1 = \frac{3}{4}, \quad S_2 = \frac{3}{4}, \quad S_3 = \frac{1}{4}$$

Allora otteniamo esattamente il metodo di Chaikin

Es (FOUR-POINT SCHEME)

$$P_{2i}^{(j+1)} = P_i^{(j)}$$

$$P_{2i+1}^{(j+1)} = -\frac{1}{16} P_{i-1}^{(j)} + \frac{9}{16} P_i^{(j)} + \frac{9}{16} P_{i+1}^{(j)} - \frac{1}{16} P_{i+2}^{(j)}$$

S: dimostra che questo metodo converge ad una curva C^2 (non spline)

Una matrice infinita S stocastica per righe con finite column height definisce uno schema di suddivisione (che può convergere o non convergere)

15/12

Nel caso delle B-spline abbiamo visto che la matrice di suddivisione ci fornisce una strategia di raffinamento, $[B(2t)]^T = [B(t)]^T S$

Notazione

- vettore differenze di punti consecutivi:

sia Δ matrice tc $\Delta_{ij} = \begin{cases} -1 & \text{se } j=i \\ 1 & \text{se } j=i+1 \\ 0 & \text{altrimenti} \end{cases}$ $\Delta = \begin{bmatrix} 1 & & & \\ -1 & \ddots & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix}$

Allora $\Delta P^j = \Delta \begin{bmatrix} P_i^j \\ P_{i+1}^j \\ \vdots \end{bmatrix}$ e $(\Delta P^j)_i = P_{i+1}^j - P_i^j$

- le norme utilizzate sono tutte norme infinito, in particolare:

- $\|P\| = \sup_i \|P_i\|$ (vettore di punti)
- $\|S\| = \sup_j \sum_i |S_{ij}|$ (matrice)
- $\|f\| = \sup_t \|f(t)\|$ (funzione)

- Indichiamo con $P^j(t)$ la spezzata che congiunge i punti al passo j -esimo ... $P_i^j, P_{i+1}^j, P_{i+2}^j, \dots$

Vale $P(t) = \underbrace{[B_1(t)]^T}_{\substack{\downarrow \\ \text{vettore dei punti}}} P = \sum_k B_1^{(k)}(t) P_k = \sum_k B_1(t-k) P_k$
 vettore della B-spline cardinal: d: grado \pm

Prop (condizione di convergenza per regole di suddivisione)

Se $\exists c > 0$, $0 < \gamma < 1$, j_0 tc $\forall j \geq j_0 \quad \|\Delta P^j\| \leq c \gamma^j$

allora $P^j(t)$ converge ad una certa curva $P^\infty(t)$

Dim Sia S la matrice di suddivisione del metodo considerato e sia S_1 la matrice di suddivisione delle B-spline di grado 1.

Abbiamo visto che le colonne di S_1 sono della forma $(\dots 0 \frac{1}{2} \frac{1}{2} 0 \dots)^T$ e sono shiftate in basso di 2 posti rispetto alla precedente.

$$\text{Sia } u = (\dots 1 1 1 \dots)^T$$

Oss

- $(S - S_1)u = 0$ poiché $Su = S_1u = u$
- Vale $S - S_1 = D \cdot \Delta$

dove D è la matrice definita da $D_{ij} = -\sum_{k=i}^j (S - S_1)_{ik}$.

$$\begin{aligned} \|P^{j+1}(t) - P^j(t)\| &= \|B_1(2^{j+1}t) \underbrace{P^{j+1}}_1 - \underbrace{B_1(2^j t) P^j}_1\| = \\ &= \|B_1(2^{j+1}t) \underbrace{SP^j}_1 - \underbrace{B_1(2^{j+1}t) S_1 P^j}_1\| = \\ &= \|B_1(2^{j+1}t) (S - S_1) P^j\| = \\ &= \|B_1(2^{j+1}t) D \Delta P^j\| \leq \\ &\leq 1 \cdot \|D\| \cdot \|\Delta P^j\| \leq (\text{definitivamente, cioè } j \geq j_0) \\ &\leq \|D\| c \delta^j \end{aligned}$$

Allora la successione $\{P^j(t)\}_j$ converge (è d'Cauchy?) ad un limite $P^\infty(t)$. La convergenza delle curve $P^j(t)$ a $P^\infty(t)$ è uniforme perché $\|P^j(t) - P^\infty(t)\| \leq \|D\| \frac{c \delta^j}{1-\gamma}$ (è cioè di succ. geometrica), quindi $P^\infty(t)$ è una curva continua.

□

In pratica: data S (ovvero le regole di suddivisione) come possiamo dire se il metodo converge (ovvero se valgono le ipotesi della proposizione)?

Vale $\Delta P^{j+1} = \Delta S P^j$ ed esiste una matrice D (non quella della prop.) t.c. $\Delta S = D \Delta$ quindi $\Delta P^{j+1} = D \Delta P^j$. Se $\|D\| \leq 1$ allora il metodo converge.

Es (metodo di suddivisione per spline cubiche)

$$P_{2i+1}^{j+1} = \frac{1}{8}(4P_i^j + 4P_{i+1}^j)$$

$$P_{2i}^{j+1} = \frac{1}{8}(P_{i-1}^j + 6P_i^j + P_{i+1}^j)$$

Per quanto riguarda le differenze ottengono

$$\begin{aligned} (\Delta P^{j+1})_{2i} &= P_{2i+1}^{j+1} - P_{2i}^{j+1} = \\ &= \frac{1}{8}(4P_i^j + 4P_{i+1}^j - P_{i-1}^j - 6P_i^j - P_{i+1}^j) = \\ &= \frac{1}{8}(3P_{i+1}^j - 2P_i^j - P_{i-1}^j) = \\ &= \frac{1}{8}(3(P_{i+1}^j - P_i^j) + (P_i^j - P_{i-1}^j)) \end{aligned}$$

$$\begin{aligned} (\Delta P^{j+1})_{2i+1} &= P_{2i+2}^{j+1} - P_{2i+1}^{j+1} = \\ &= \frac{1}{8}(P_i^j + 6P_{i+1}^j + P_{i+2}^j - 4P_i^j - 4P_{i+1}^j) = \\ &= \frac{1}{8}(P_{i+2}^j + 2P_{i+1}^j - 3P_i^j) = \\ &= \frac{1}{8}((P_{i+2}^j - P_{i+1}^j) + 3(P_{i+1}^j - P_i^j)) \end{aligned}$$

Quindi la matrice D ha la seguente forma

$$D = \frac{1}{8} \begin{bmatrix} \dots & 0 & 3 & 1 & 0 & 0 & \dots \\ \dots & 0 & 0 & 1 & 3 & 0 & \dots \\ & & & & \vdots & & \end{bmatrix} \Rightarrow \|D\| = \frac{1}{2}$$

Subdivision surfaces

Drawbacks of NURBS surfaces:

- ▶ A single NURBS surface can only represent simple topologies (disk, cylinder, torus).
- ▶ Many NURBS patches are needed to model complex geometry.
- ▶ When deforming a surface made of NURBS patches, cracks may arise at the seams. (Serious drawback in animation).

Idea behind subdivision: start with a coarse mesh and define a smooth curve or surface as the limit of a sequence of successive refinements.

Refinements are performed according to **subdivision rules**.

Subdivision surfaces

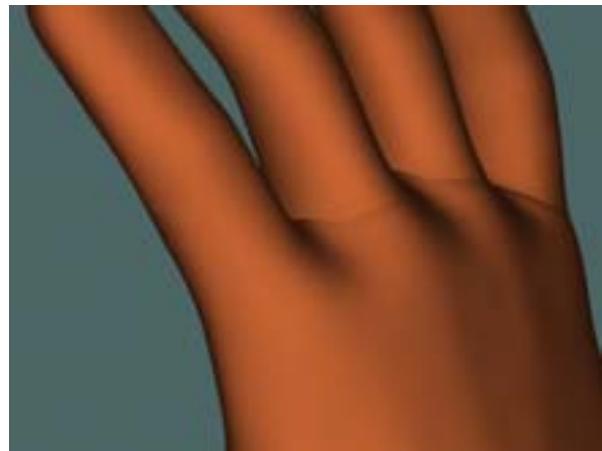
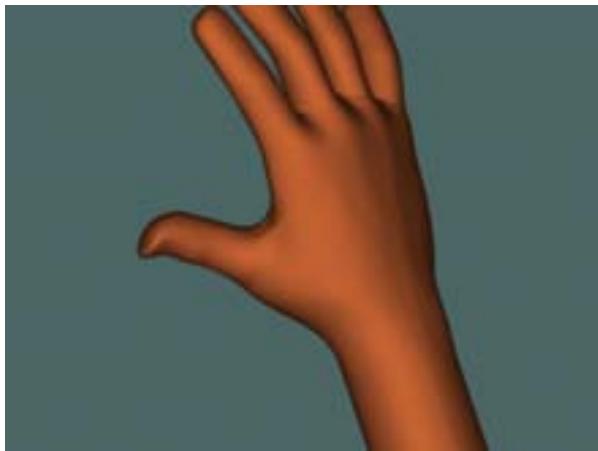
The first significant application of subdivision surfaces in animation appears in Pixar's short film *Geri's Game*.



Geri's Game won the 1997 Academy Award for Best Animated Short Film.

Subdivision surfaces

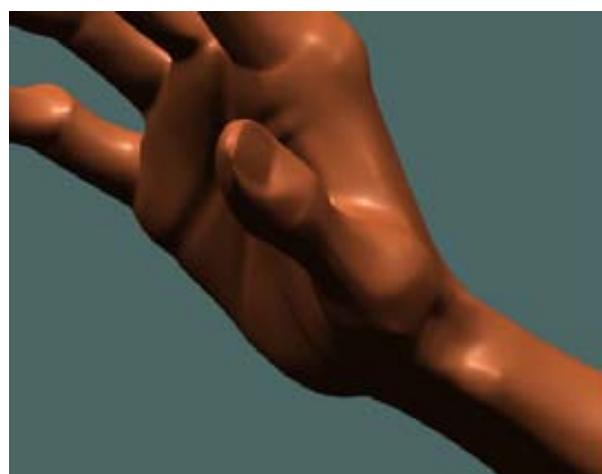
Look at Woody's hand (NURBS):



Woody is a character from Pixar's *Toy Story* (1995).

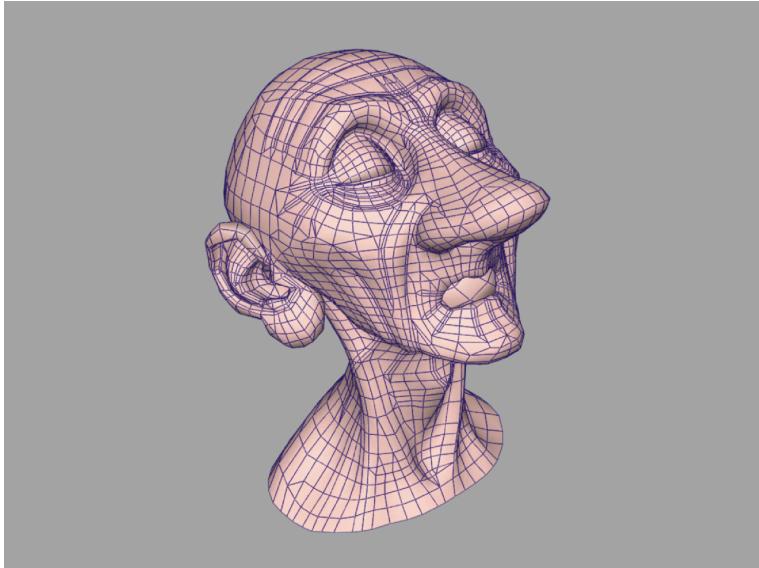
Subdivision surfaces

Compare with Geri's hand:



Subdivision surfaces

Subdivision surfaces were used to model Geri's head, hands and clothes.



Control mesh for Geri's head, created by digitizing a full-scale model sculpted out of clay. Taken from *Subdivision Surfaces in Character Animation*, by Tony DeRose, Michael Kass and Tien Truong.

Subdivision surfaces

Goals of subdivision schemes:

- ▶ **Efficiency**: compute new points cheaply.
- ▶ **Compact support**: each point influences the shape of the final surface only over a small, finite region.
- ▶ **Local definition**: the subdivision rules should only depend on “close” points.
- ▶ **Affine invariance**: if an affine transformation is applied to the original set of points, the resulting surface should undergo the same transformation.
- ▶ **Simplicity**: there should be a small number of subdivision rules, and determining the rules should preferably be an offline process.
- ▶ **Continuity**: the resulting surfaces should have “nice” continuity properties (be \mathcal{C}^1 , for instance).

Meshes

Meshes are defined by a set of vertices, a set of edges and a set of faces. We assume that

- ▶ each pair of vertices is connected by at most one edge,
- ▶ each vertex has valence at least 2, *ovvero almeno 2 spigoli incidenti*
- ▶ each face has valence at least 3.

Note that faces are not necessarily planar.

A boundary edge is an edge that does not belong to two distinct faces. When dealing with boundaries, one generally assumes that each boundary edge belongs to exactly one face, and no vertex belongs to three distinct boundary edges.

Subdivision surfaces

Subdivision allows us to build surfaces of **arbitrary topology**.

Topology means:

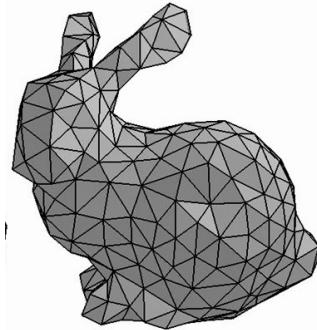
- ▶ the topological genus of the mesh and of the resulting surface,
- ▶ the topology of the mesh, in particular the degrees of vertices.

These two notions are related by the properties of the Euler characteristic. For instance:

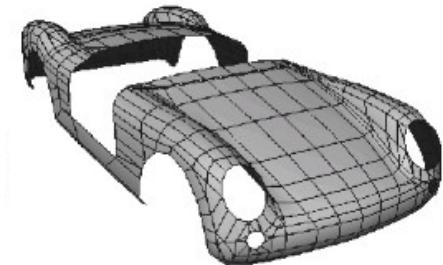
- ▶ if we want a quadrilateral control mesh with all vertices of degree 4, then the resulting shape is necessarily homeomorphic to the plane, the cylinder or the torus,
- ▶ but if we allow for vertices of different degrees, we can build surfaces in a wider range of topological classes.

Meshes

Most subdivision methods are defined on **triangular meshes**:



or on **quadrilateral meshes**:

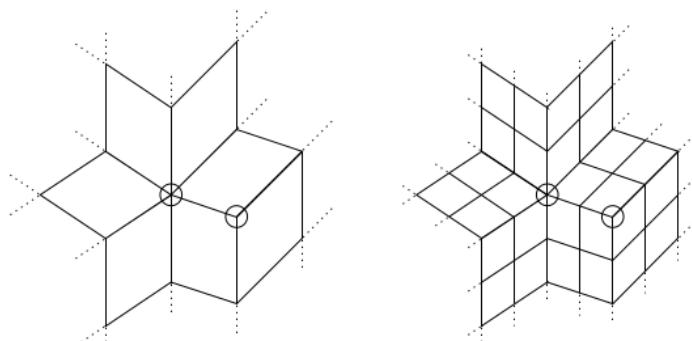


Let us assume for simplicity that we work on surfaces without boundary.

Valence

- ▶ The **degree** or **valence** of a vertex V in a mesh is the number of edges that have V as an endpoint (or, equivalently, the number of adjacent vertices).
- ▶ The **standard valence** of a vertex is 6 for triangular meshes and 4 for quadrilateral meshes.
- ▶ A vertex whose valence is not standard is called **extraordinary**.

(in ita: "eccezionali")



Subdivision rules

- ▶ For each subdivision scheme, transition from one subdivision level to the next is described by **subdivision rules**. Such rules compute points on the finer level as linear combinations of points on the coarser level.
- ▶ There may be separate rules for different vertex valences: one rule for **standard vertices** and one or more rules for **extraordinary vertices**. *valenza straordinaria*
- ▶ In case of **boundary** or **creases** (sharp features in the surface): more rules for boundary points.

Regular quad-mesh subdivision

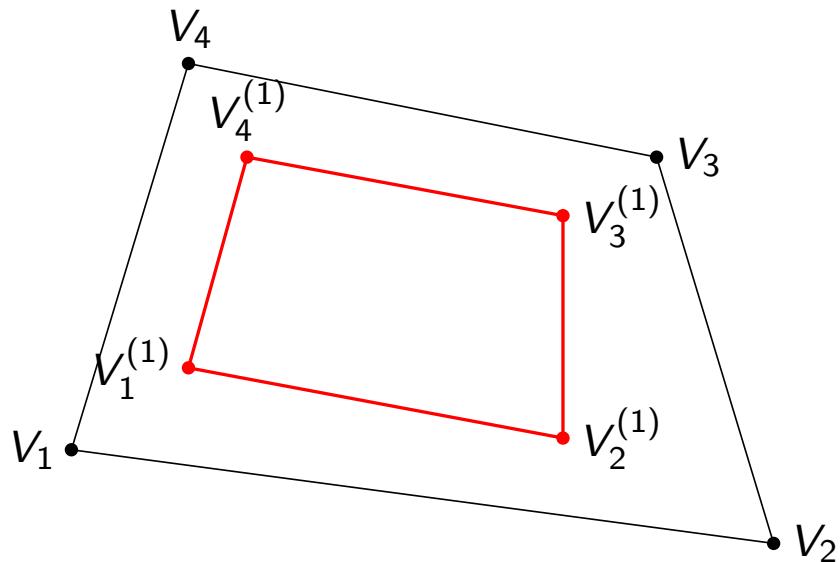
- ▶ Start from a quad-mesh: all faces are quadrilaterals, each vertex has 4 adjacent edges and 4 adjacent faces.
- ▶ Let V_1, V_2, V_3, V_4 be the vertices of a given face.
- ▶ Refinement: for $i = 1, 2, 3, 4$ replace V_i with

$$V_i^{(1)} = \frac{3}{16} V_{i-1} + \frac{9}{16} V_i + \frac{3}{16} V_{i+1} + \frac{1}{16} V_{i+2} \quad \text{mod } 4.$$

- ▶ The new mesh is still quadrilateral.
- ▶ The method converges to biquadratic B-splines.

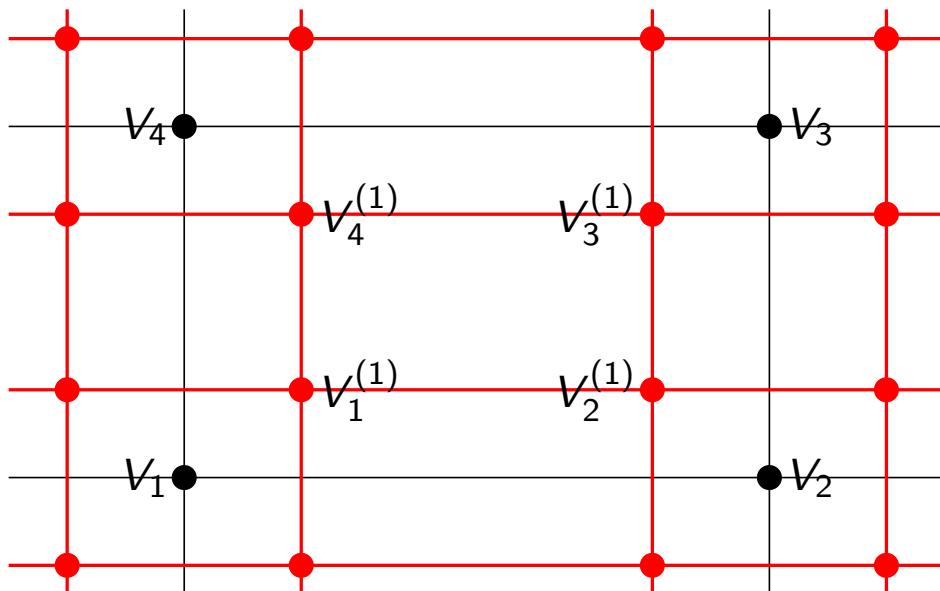
Regular quad-mesh subdivision

$$\begin{bmatrix} V_1^{(1)} \\ V_2^{(1)} \\ V_3^{(1)} \\ V_4^{(1)} \end{bmatrix} = \frac{1}{16} \begin{bmatrix} 9 & 3 & 1 & 3 \\ 3 & 9 & 3 & 1 \\ 1 & 3 & 9 & 3 \\ 3 & 1 & 3 & 9 \end{bmatrix} \cdot \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix}$$



Regular quad-mesh subdivision

Old vertices are removed:



Doo-Sabin subdivision

The previous rule is a special case of Doo-Sabin subdivision:

- ▶ Start from an arbitrary mesh.
- ▶ For each face: let n be its valence and define new vertices as

$$V_i^{(1)} = \sum_{j=1}^n \alpha_{i,j} V_j,$$

with

$$\alpha_{i,j} = \begin{cases} \frac{n+5}{4n} & i = j, \\ \frac{3+2 \cos \frac{2\pi(i-j)}{n}}{4n} & i \neq j. \end{cases}$$

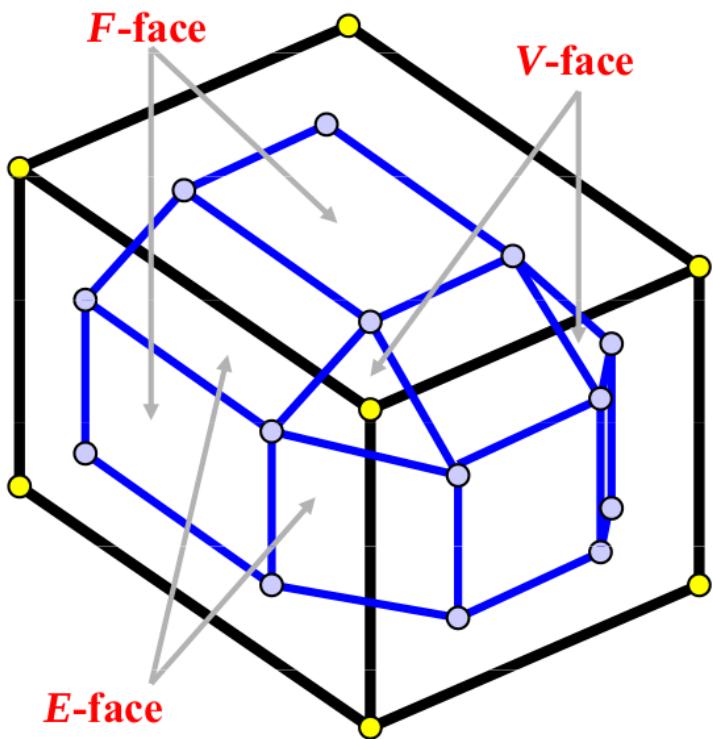
D. Doo and M. Sabin, *Behaviour of recursive division surface near extraordinary points*, Computer Aided Design 10 (1978), 356-360.

Doo-Sabin subdivision

Three types of faces in the new mesh:

- ▶ **F-faces**: found by cyclically connecting the $V_i^{(1)}$ of each original face,
- ▶ **E-faces**: for each pair of faces sharing a common edge, consider the new face produced by the four $V_i^{(1)}$ lying on the lines that connect the centroid of each face to the edge endpoints,
- ▶ **V-faces**: found by connecting the $V_i^{(1)}$ that surround each original vertex.

Doo-Sabin subdivision



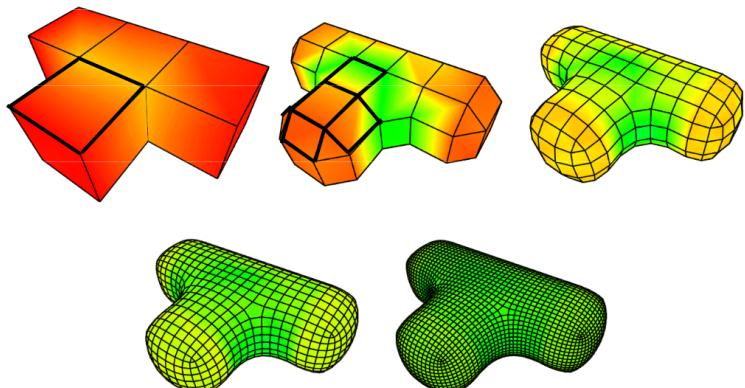
Some figures for Doo-Sabin and Catmull-Clark are from C.-K. Shene, *Introduction to Computing with Geometry*.

Doo-Sabin subdivision

The algorithm produces four-sided faces, with the exception of

- ▶ F-faces generated by initial faces that are not four-sided,
- ▶ V-faces generated by extraordinary vertices.

After the first step, all vertices have valence 4.



- ▶ Doo-Sabin limit surfaces are G^1 everywhere and are “mostly” biquadratic B-splines (exception: extraordinary vertices).
- ▶ The centroid of any original face will be on the limit surface.

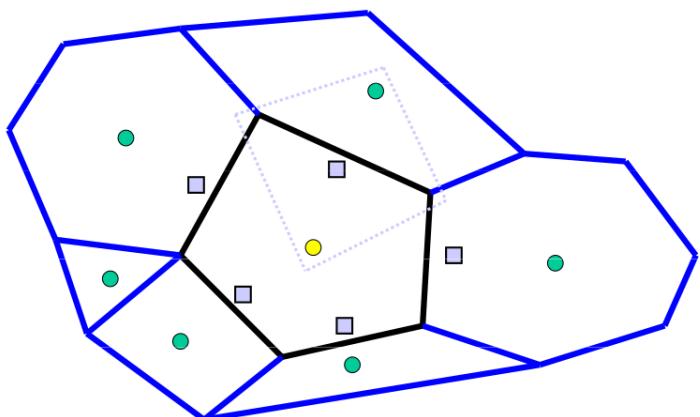
Catmull-Clark subdivision

- ▶ Catmull-Clark subdivision is a generalization of bicubic B-splines to arbitrary topology (just like Doo-Sabin generalizes biquadratic B-splines).
- ▶ The limit surface is C^1 for extraordinary points (valence $\neq 4$) and C^2 everywhere else. It is globally G^2 .
- ▶ Idea: compute a new face point for each face, a new edge point for each edge and a new vertex point for each vertex.

E. Catmull and J. Clark, *Recursively generated B-spline surfaces on arbitrary topological meshes*, Computer Aided Design 10 (1978), 350-355.

Catmull-Clark subdivision

- ▶ Each face point (small disk) is the centroid of a face.
- ▶ Each edge point (small square) is the average of the two endpoints of an edge and of the two face points of that edge's adjacent faces.



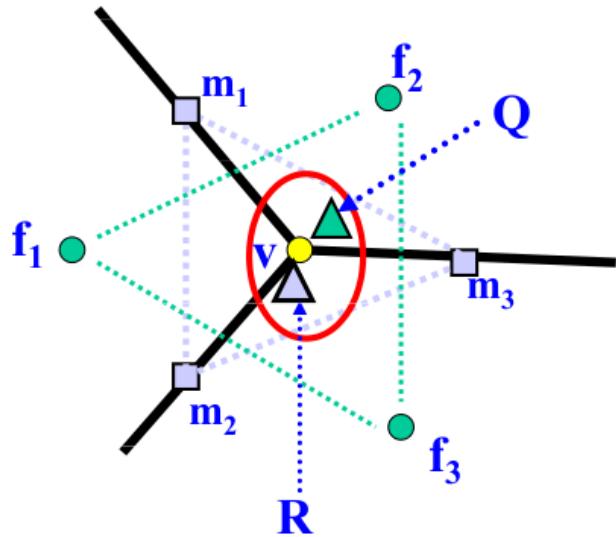
Catmull-Clark subdivision

For each vertex v compute a new vertex point

$$v' = \frac{1}{n}Q + \frac{2}{n}R + \frac{n-3}{n}v,$$

where

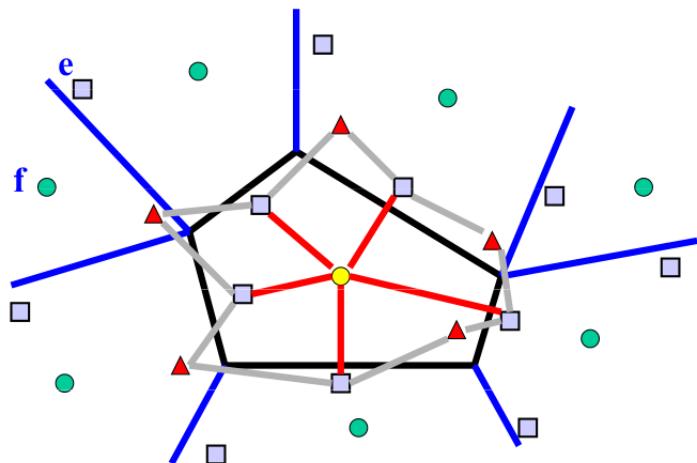
- ▶ n is the valence of v ,
- ▶ Q is the average of all new face points adjacent to v ,
- ▶ R is the average of all midpoints of incident edges of v .



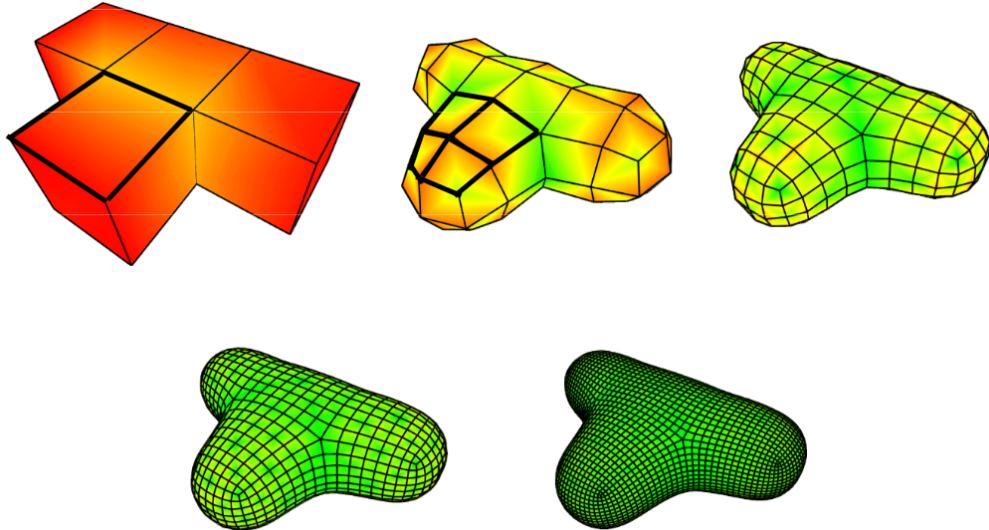
Catmull-Clark subdivision

Now let's connect the new vertices.

- ▶ For each face, connect its face point to each edge point.
- ▶ Connect each new vertex v' (associated with old vertex v) to the two edge points of the edges incident to v .

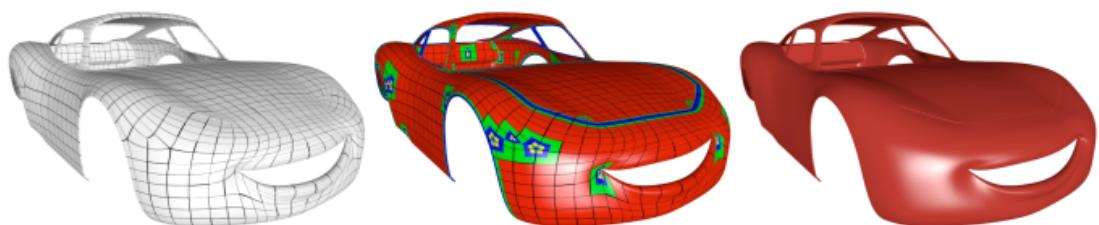


Catmull-Clark subdivision



Catmull-Clark subdivision

An example:



Input base mesh (left), subdivision patch structure (center), rendered final model (right). ©Disney/Pixar.

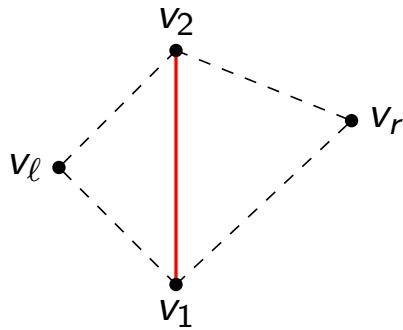
M. Niessner, C. Loop, M. Meyer and T. DeRose, *Feature Adaptive GPU Rendering of Catmull-Clark Subdivision Surfaces*, ACM Transactions on Graphics 31(1), 2012.

Loop subdivision

- ▶ Algorithm proposed by **Charles Loop** in 1987.
- ▶ Based on a **triangular mesh**.
- ▶ Computes a new edge point e for each edge and a new vertex for each vertex.

The edge point e associated with edge v_1v_2 is computed as

$$e = \frac{3}{8}(v_1 + v_2) + \frac{1}{8}(v_\ell + v_r).$$



Charles Loop: Smooth Subdivision Surfaces Based on Triangles, M.S. Mathematics thesis, University of Utah, 1987.

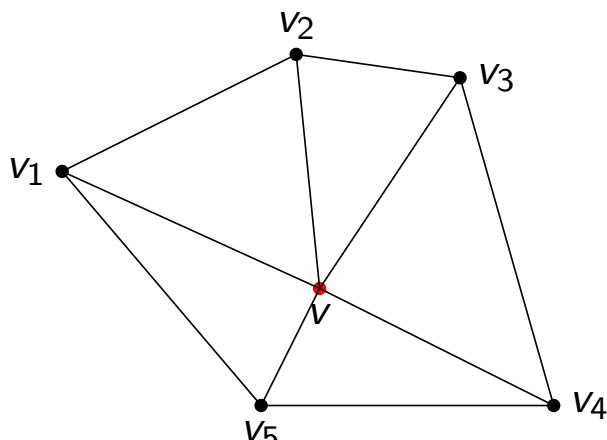
Loop subdivision

Now let v be a vertex, n its valence, v_1, \dots, v_n its adjacent vertices. The new vertex v' associated with v is

$$v' = (1 - n\alpha)v + \alpha \sum_{j=1}^n v_j$$

with

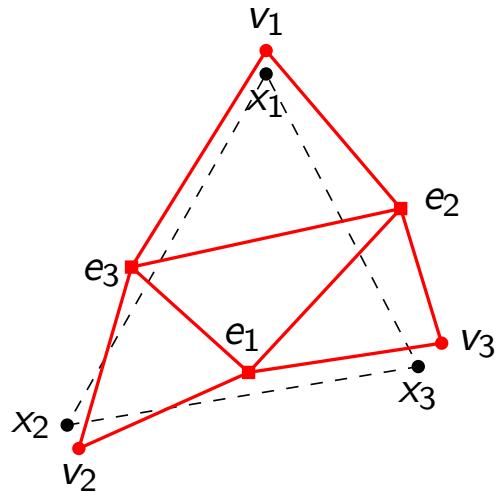
$$\alpha = \begin{cases} \frac{3}{16} & n = 3 \\ \frac{1}{n} \left(\frac{5}{8} - \left(\frac{3}{8} + \frac{1}{4} \cos \frac{2\pi}{n} \right)^2 \right) & n > 3 \end{cases}$$



Loop subdivision

How do we connect the new vertices?

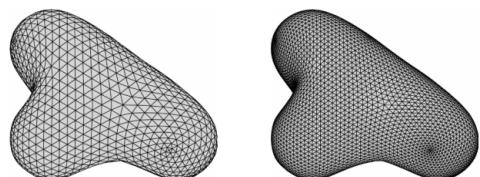
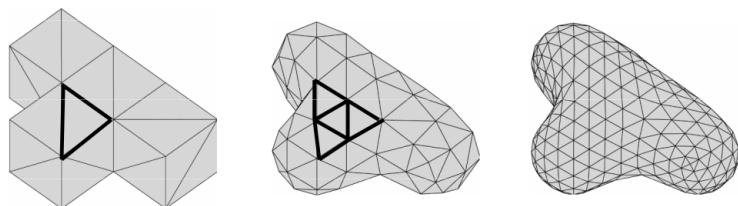
- ▶ Consider a triangle of vertices x_1, x_2, x_3 in the “old” mesh.
- ▶ Let v_1, v_2, v_3 be the new vertex points (v_i associated with x_i).
- ▶ Let e_1, e_2, e_3 be the new edge points (e_i associated with $x_{i+1}x_{i+2} \bmod 3$).
- ▶ Define new edges so as to have triangles $v_i e_{i+1} e_{i+2} \bmod 3$ and $e_1 e_2 e_3$.



Loop subdivision

- ▶ If all vertices have valence 6, the limit surface is C^2 (Bézier triangles). It is only G^1 at extraordinary vertices.
- ▶ A vertex with n adjacent vertices v_1, \dots, v_n converges to

$$v_\infty = \frac{3 + 8(n-1)\alpha}{3 + 8n\alpha} + \frac{8\alpha}{3 + 8n\alpha} \sum_{j=1}^n v_j.$$



Classification of subdivision schemes

Primal/Dual

- ▶ Primal: faces are split into sub-faces (e.g., Loop, Catmull-Clark).
- ▶ Dual: vertices are split into multiple vertices (e.g., Doo-Sabin).

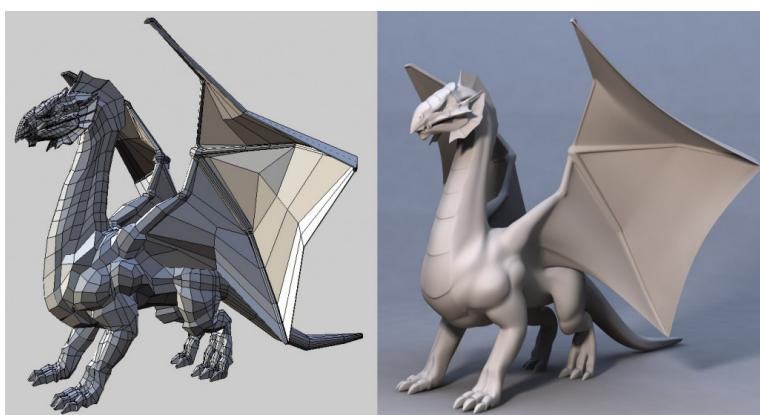
Approximating/Interpolating

- ▶ Approximating: control points are not interpolated (e.g., Loop, Catmull-Clark).
- ▶ Interpolating: control points are interpolated.

Stationary/Non-stationary

- ▶ Stationary: the same subdivision matrix is used at each iteration.
- ▶ Non-stationary: different subdivision matrices are used.

Software



Free software based on surface subdivision includes:

- ▶ Wings3d (www.wings3d.com): subdivision modeler. Uses Baumgart's winged-edge data structure.
- ▶ OpenSubdiv: subdivision API by Pixar and Microsoft.
- ▶ Blender (www.blender.org): 3D creation (modeling, animation, rendering... also uses OpenSubdiv).