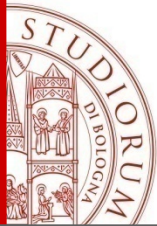


Curve spline e NURBS

Tool-box nurbs



Curve NURBS

$$C(t) = \sum_{i=1}^{m+K} P_i R_{i,m}(t) \quad t \in [a, b]$$

Le funzioni e curve NURBS sono una generalizzazione razionale delle funzioni e curve spline;

le $R_{i,m}$ sono le funzioni RB-spline (Rational B-spline) di ordine m (dove $\text{grado} = m - 1$)

P_i sono i punti di controllo

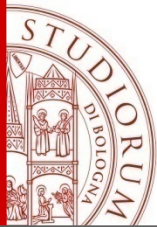
Il tool-box nurbs che andiamo a presentare permette quindi di gestire curve nurbs, ma anche come loro caso particolare curve spline sia 2D che 3D e superfici.

Curve NURBS

$$C(t) = \sum_{i=1}^{m+K} P_i R_{i,m}(t) \quad t \in [a, b]$$

- m è l'ordine (grado + 1) della curva
- k è il numero di nodi ($k+1$ è il numero di tratti)
- m_i sono le molteplicità dei nodi e $K = \sum_{i=1}^k m_i$
- t_i sono i nodi della partizione estesa e sono $K+2m$
- $m+K$ è il numero totale dei punti di controllo (dim di S)
- $R_{i,m}$ sono le funzioni RB-spline di ordine m
- P_i sono i punti di controllo

Ciascuna $R_{i,m}$ è non nulla solo in un certo intervallo (t_i, t_{i+m}) , detto supporto, così che la curva ha controllo locale



Geometria NURBS

Nel **tool-box nurbs** la geometria viene memorizzata in una structure array di Matlab;

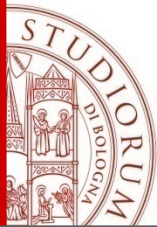
Una **structure array** è un tipo di dato che raggruppa le informazioni usando dei contenitori detti **campi**.

Ogni campo può contenere qualsiasi tipo di informazione.

L'accesso ai dati di un campo si effettua usando la sintassi

NomeStruttura.NomeCampo

Una structure array può essere allocata e inizializzata prima di usarla, oppure come la variabili o gli array può essere definita nel momento che la si usa:



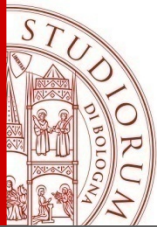
Geometria NURBS

Vediamo un paio di **Esempi**:

```
%inizializzata quando si usa  
crv.form='B-NURBS';  
crv.dim=7;
```

```
%allocata e inizializzata prima di usarla  
crv=struct('form','B-NURBS','dim',7);
```

Entrambe producono una structure di nome `crv` con due campi rispettivamente **form** e **dim**; il primo campo contiene un valore stringa di caratteri, il secondo un valore numerico.



Geometria NURBS

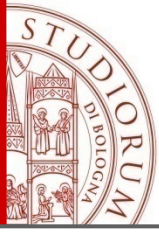
Un **cell array** è un tipo di dato con dei contenitori di informazioni indicizzati chiamati celle, dove ogni cella può contenere un qualunque tipo di informazione.

I cell array solitamente sono liste di caratteri, un misto di stringhe e numeri o array di numeri di differenti lunghezze.

Un cell array può essere allocato o allocato e inizializzato con i comandi:

```
C=cell(size1,size2, ... )           % alloca un array C di celle
```

```
C={1,2,3; 'text', 3, {11;22;33}} %alloca un array C 2x3 di celle  
e la cella di indici (2,3) è a sua volta un array 3x1 di celle
```



Geometria NURBS

Per accedere al contenuto delle celle si usano le graffe.

Esempio:

$C\{1,1\}$ è l'elemento numerico 1

$C\{2,1\}$ è la stringa 'text'

$C\{2,3\}$ è un array 3x1 di celle e per accedere a questi elementi si deve scrivere:

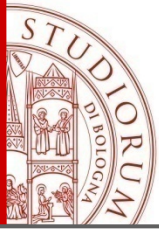
$C\{2,3\}\{1\}$ per 11

$C\{2,3\}\{2\}$ per 22

$C\{2,3\}\{2\}$ per 33

Esempio:

$D=\{ \};$ % alloca un array vuoto



Geometria NURBS

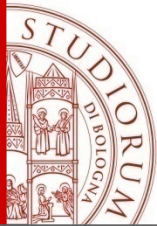
Le entità geometriche nel tool-box nurbs sono descritte in una struttura che contiene tutte le informazioni necessarie.

Se chiamiamo **geom** la structure di una geometria NURBS, i campi principali saranno:

geom.order : un vettore con l'ordine in ogni direzione;

geom.knots: un cell array con la/le partizione/i nodale/i estese;

geom.number: numero di funzioni base in ogni direzione;



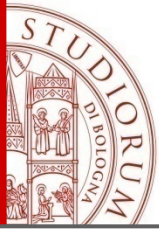
Geometria NURBS

geom.coefs: array con i punti di controllo in coordinate omogenee; per una curva la dim è $(4, n1)$, per una superficie $(4, n1, n2)$ e per un volume $(4, n1, n2, n3)$.

Con riferimento a **geom.coefs** i punti di controllo (CP) sono a 4 coordinate

$$\begin{bmatrix} x*w \\ y*w \\ z*w \\ w \end{bmatrix}$$

e per una curva avremo $n1$ CP, per una superficie $n1 \times n2$ CP, per un volume $n1 \times n2 \times n3$ CP.

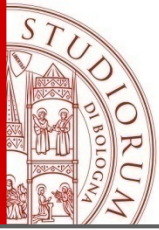


Geometria NURBS

Sempre con riferimento a **geom.coefs** i punti di controllo (CP) sono a 4 coordinate e le prime tre contengono le coordinate dei CP moltiplicate per il peso;

I punti di controllo (CP) sono a 4 coordinate anche per geometrie 2D;

Nel caso di curve o superfici spline i coefficienti peso w sono sempre 1.

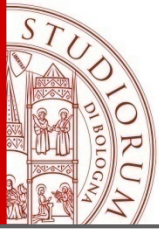


Esercizio A.1

Testo. Utilizzare il comando `crv=nrblne(...)` per generare una curva NURBS che connetta i punti $(0,0)^T$ e $(2,1)^T$ con un segmento retto. Si usi `help nrblne` per la sintassi. Quali informazioni fornisce la `crv` della struttura NURBS che si è creata? Esaminare `crv.coefs` e `crv.knots`

Soluzione:

```
%script main_A1.m  
crv = nrblne ([0 0], [2 1]);  
nrbctrlplot ( crv );  
crv  
crv.coefs  
crv.knots
```



Esercizio B.1

Testo. Per creare una curva NURBS di ordine 3, è possibile utilizzare il comando `nrbmak(...)`. Si usino gli stessi punti di controllo di A.2, ma si adatti il vettore dei nodi in modo da definire una curva quadratica.

Soluzione:

```
%script main_B1.m
```

```
ctrl=[0 2 4 6;
```

```
      0 1 0 2;
```

```
      0 0 0 0;
```

```
      1 1 1 1];
```

```
crv2 = nrbmak (ctrl, [0 0 0 0.5 1 1 1]);
```

```
nrbctrlplot ( crv2 );
```

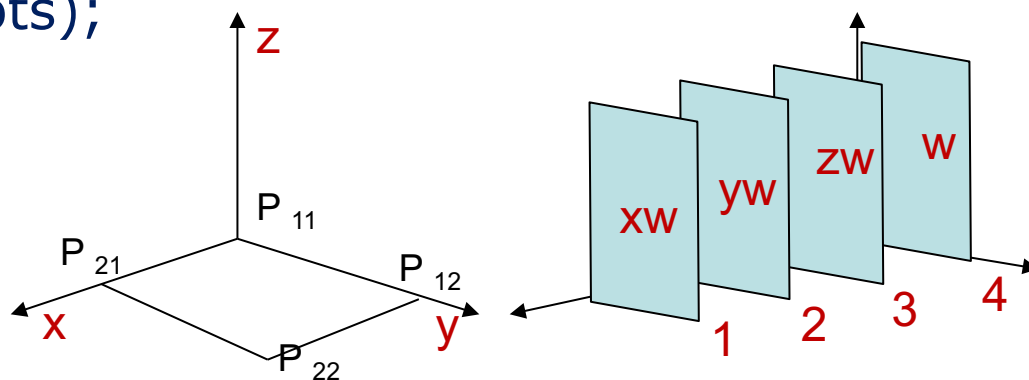
Esercizio C.1

Testo. Come le curve, anche le superfici possono essere create con il comando `nrbmak(..,..)`. Lo si usi per creare il quadrato (piano) unitario con punti di controllo nei quattro vertici.

Soluzione:

```
coefs = cat(3,[0 0;  
              0 1],[1 1;  
                  0 1]);  
knots = {[0 0 1 1] [0 0 1 1]};  
plane = nrbmak(coefs,knots);  
nrbplot(plane, [2 2]);  
nrbctrlplot(plane);  
plane.coefs
```

```
plane.coefs(:, :, 1) =    0    0  
                        0    1  
                        0    0  
                        1    1  
plane.coefs(:, :, 2) =    1    1  
                        0    1  
                        0    0  
                        1    1
```



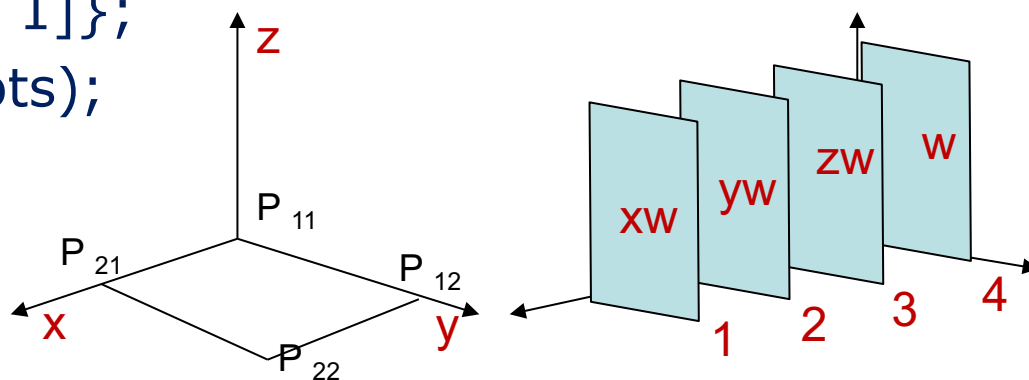
Esercizio C.1

Testo. Come le curve, anche le superfici possono essere create con il comando `nrbmak(..,..)`. Lo si usi per creare il quadrato (piano) unitario con punti di controllo nei quattro vertici.

Soluzione:

```
CX=[0 0; 1 1];
CY=[0 1; 0 1];
coefs(1, :, :)=CX;
coefs(2, :, :)=CY;
knots = {[0 0 1 1] [0 0 1 1]};
plane = nrbmak(coefs,knots);
nrbplot(plane, [2 2]);
nrbctrlplot(plane);
```

```
plane.coefs(:, :, 1) =    0    1
                        0    0
                        0    0
                        1    1
plane.coefs(:, :, 2) =    0    1
                        1    1
                        0    0
                        1    1
```



Esercizio C.4

Testo. Come le superfici, anche i volumi possono essere creati con il comando `nrbmak(...)`. Lo si usi per creare il cubo unitario con punti di controllo negli otto vertici.

Soluzione:

```
CX = zeros(2,2,2);
```

```
CX(:, :, 1)=[0 0; 1 1];
```

```
CX(:, :, 2)=[0 0; 1 1];
```

```
CY = zeros(2,2,2);
```

```
CY(:, :, 1)=[0 1; 0 1];
```

```
CY(:, :, 2)=[0 1; 0 1];
```

```
CZ = zeros(2,2,2);
```

```
CZ(:, :, 1)=[0 0; 0 0];
```

```
CZ(:, :, 2)=[1 1; 1 1];
```

```
CW=ones(size(CZ));
```

```
coefs(1, :, :, :)=CX.*CW;
```

```
coefs(2, :, :, :)=CY.*CW;
```

```
coefs(3, :, :, :)=CZ.*CW;
```

```
coefs(4, :, :, :)=CW;
```

```
knots = {[0 0 1 1] [0 0 1 1] [0 0 1 1] };
```

```
plane = nrbmak(coefs,knots);
```

```
nrbplot(plane, [2 2]);
```

```
nrbctrlplot(plane);
```

