NURBS surfaces

CAD Surfaces

NURBS surfaces

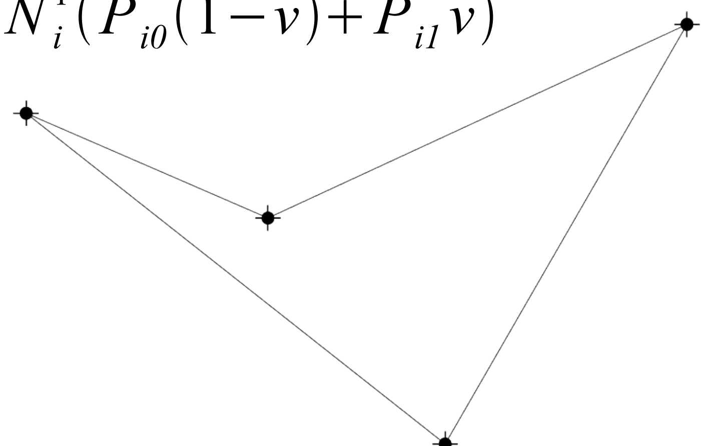
- Basic surfaces
  - Biliear patch
  - Ruled surfaces
  - Extruded surfaces
  - Coons patch
- Advanced surface algorithms
  - Generalized revolution surfaces
  - Profiled surfaces
- Geometric modelling and B-REP topology
- Open questions

Basic surfaces

- Bilinear patches
  - Through 4 points, we want to build a surface supported by the 4 straight lines joining the points.
    $P_{00}, P_{01}, P_{11}, P_{10}$
  - The surface has the following expression :
    $$S(u,v)=P_{00}(1-u)(1-v)+P_{01}(1-u)v+P_{10}u(1-v)+P_{11}uv$$
  - Hence the transformation into a B-spline : $S(u,v)=\sum_{i=0}^{1} N_i^1(P_{i0}(1-v)+P_{i1}v)$

$$\left.\begin{aligned}N_0^1(u)&=1-u\\ N_1^1(u)&=u\end{aligned}\right\} U=\{0,0,1,1\}$$

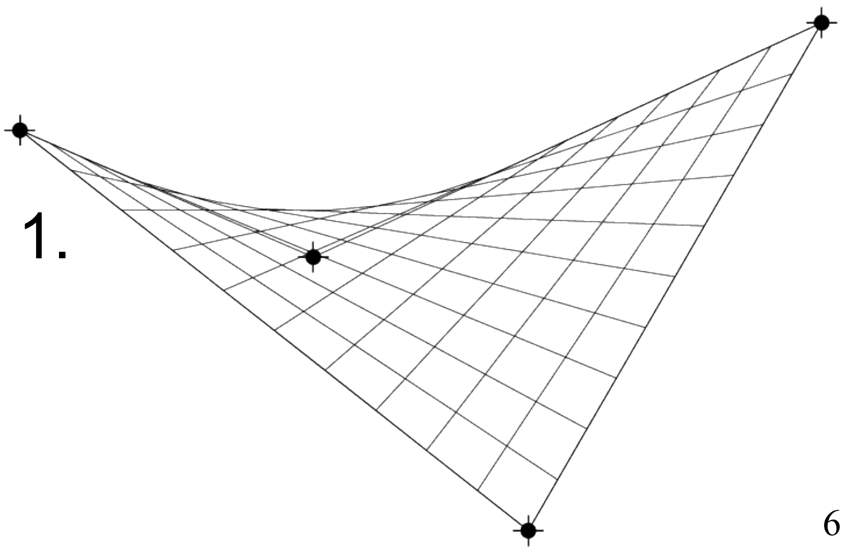
$$\left.\begin{aligned}N_0^1(v)&=1-v\\ N_1^1(v)&=v\end{aligned}\right\} V=\{0,0,1,1\}$$



- Bilinear square
  - Bézier surface of degree 1 in each direction

$$S^w(u,v) = \sum_{i=0}^{1} \sum_{j=0}^{1} N_i^1(u) N_j^1(v) P_{ij}^w$$

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



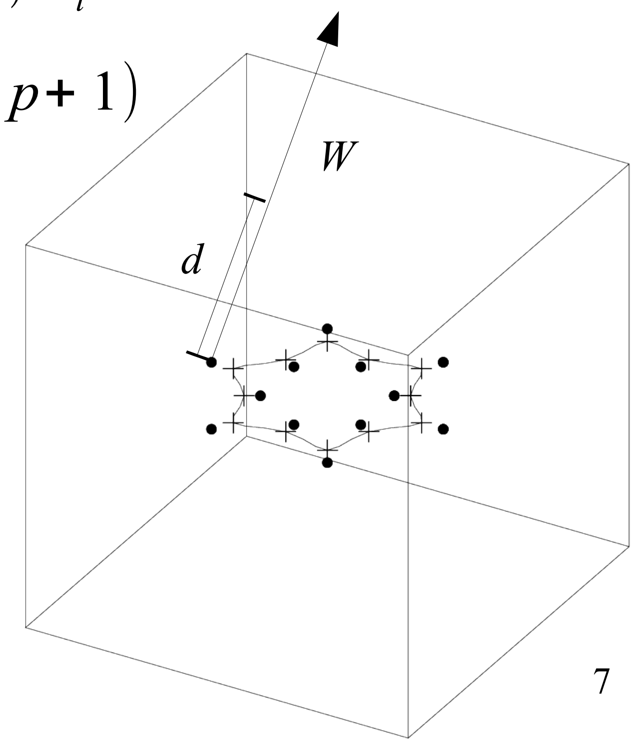
  - The weights w_i are equal to 1.
  - The surface is polynomial (non-rational)

- Extruded surfaces
  - Let C be a NURBS curve of degree p , of nodal sequence U, possibly closed, with $n+1$ control points:

$$C^w(u)=\sum_{i=0}^{n} N_i^p(u)\, P_i^w \quad C(u)=\sum_{i=0}^{n} R_i^p(u)\, P_i$$

$$U=\{u_0,\cdots,u_r\} \qquad (r+1 \text{ nodes with } r=n+p+1)$$

  - We want to extrude this curve along a unit vector W, for a length d.
  - What is the expression of the resulting surface as a NURBS ?



- Extruded surfaces

In 3D : $S(u,v)=\sum_{i=0}^{n}\sum_{j=0}^{m} R_{ij}^{p,q}(u,v)P_{ij}=\sum_{i=0}^{n} R_i^p(u)(P_i+vdW)$ $\quad W_i^w=\begin{pmatrix} W w_i \\ 0 \end{pmatrix}$

Using homog. coord.

$$S^w(u,v)=\sum_{i=0}^{n}\sum_{j=0}^{m} N_i^p(u)N_j^q(v)P_{ij}^w=\sum_{i=0}^{n} N_i^p(u)(P_i^w+vdW_i^w)$$

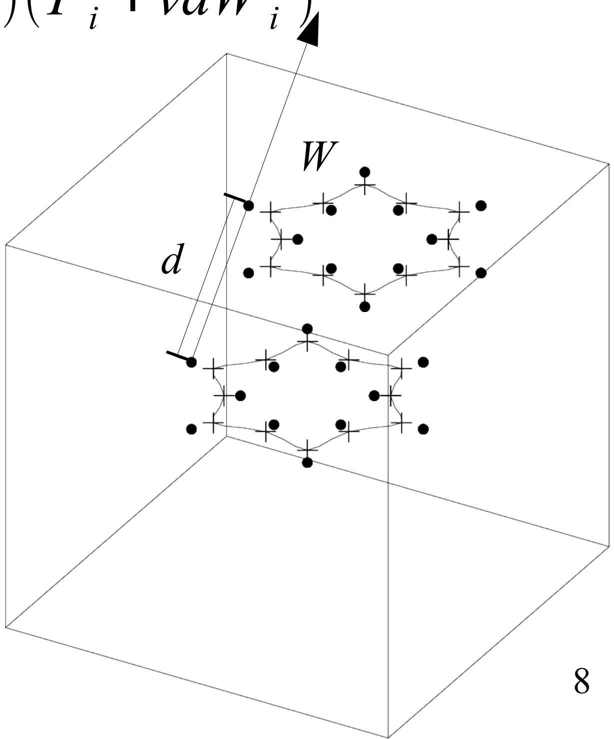
$$S^w(u,v)=\sum_{i=0}^{n} N_i^p(u)((1-v)P_{i0}^w+vP_{i1}^w)$$

$$P_{i0}^w=P_i^w$$

$$P_{i1}^w=P_i^w+dW_i^w$$

$$S^w(u,v)=\sum_{i=0}^{n} N_i^p(u)\sum_{j=0}^{1} N_j^1(v)P_{ij}^w$$

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



- Extruded surfaces

$$S^w(u,v)=\sum_{i=0}^{n}\sum_{j=0}^{1} N_i^p(u)\,N_j^1(v)\,P_{ij}^w$$

$$U=\{u_0,\cdots,u_r\} \qquad P_{i0}^w=P_i^w$$

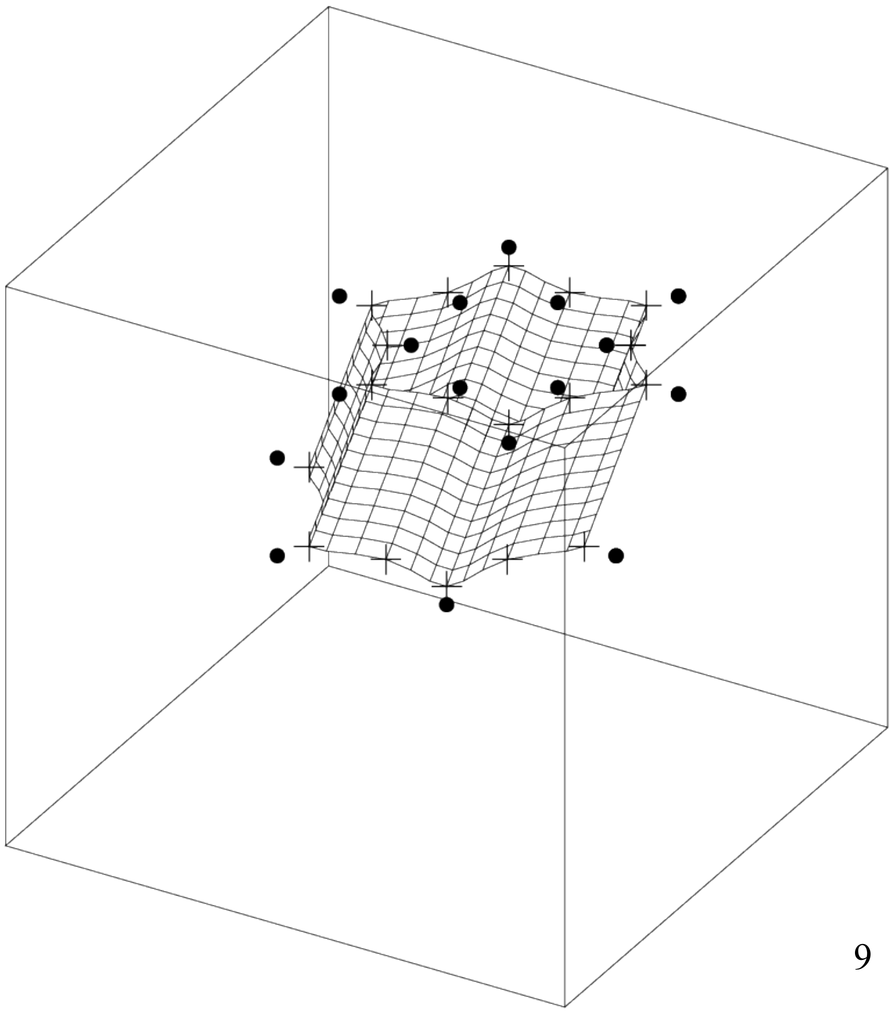
$$V=\{0,0,1,1\} \qquad P_{i1}^w=P_i^w+dW_i^w$$

$$W_i^w=\begin{pmatrix} W\,w_i \\ 0 \end{pmatrix}$$

$$S(u,v)=\sum_{i=0}^{n}\sum_{j=0}^{1} R_{ij}^{p,1}(u,v)\,P_{ij}$$

$$P_{i0}=P_i \qquad w_{i0}=w_i$$

$$P_{i1}=P_i+dW \qquad w_{i1}=w_i$$



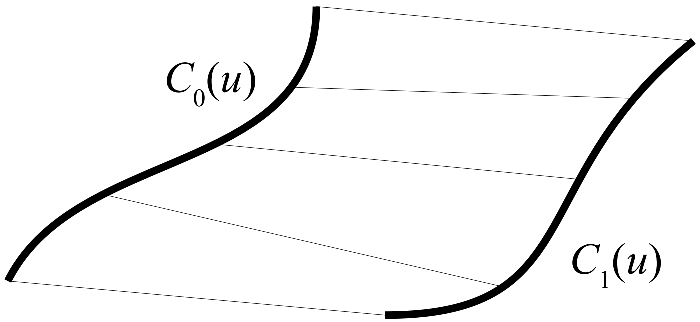
- Ruled surfaces
  - We have two curves

$$C_0^w(u)=\sum_{i=0}^{n_0} N_i^{p_0}(u)\,P_{i0}^w \qquad C_1^w(u)=\sum_{i=0}^{n_1} N_i^{p_1}(u)\,P_{i1}^w$$

$$C_0(u)=\sum_{i=0}^{n_0} R_i^{p_0}(u)\,P_{i0} \qquad C_1(u)=\sum_{i=0}^{n_1} R_i^{p_1}(u)\,P_{i1}$$

$$U_0=\{u_{00},\cdots,u_{r0}\} \qquad U_1=\{u_{01},\cdots,u_{r1}\}$$

  - We want a ruled surface in the direction v, i.e a linear interpolation between $C_0(u)$ and $C_1(u)$.



Computer Aided Design

NURBS surfaces

- Ruled surfaces
  - There are conditions on the curves $C_0(u)$ and $C_1(u)$.
    - Same parametrization (compatible nodal sequences)

$$\left.\begin{array}{l} U_0 = U_1 = U \\ p_0 = p_1 = p \end{array}\right\} \quad n_0 = n_1 = n \quad \Rightarrow \quad \left\{\begin{array}{l} C_0^w(u) = \sum_{i=0}^{n} N_i^p(u) P_{i0}^w \\ C_1^w(u) = \sum_{i=0}^{n} N_i^p(u) P_{i1}^w \end{array}\right.$$

Identical shape functions

  - The surface is then expressed simply

$$S^w(u,v) = (1-v) C_0^w(u) + v C_1^w(u)$$

$$S^w(u,v) = \sum_{j=0}^{1} N_j^1(v) C_j^w(u)$$

thus,

$$S^w(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{1} N_i^p(u) N_j^1(v) P_{ij}^w$$

$$S(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{1} R_{ij}^{p,1}(u,v) P_{ij11}$$

NURBS surfaces

- What to do if conditions on the curves $C_0(u)$ and $C_1(u)$ are not met ?

  1 – Make sure that the parametric interval matches

  - Affine transformation of one of the parameters (see chapter 3)

  2 – Degree elevation towards the highest degree $= \max(p_0, p_1)$

  - Transformation into a set of Bézier curves by node saturation (chap. 4)
  - Degree elevation for each Bézier curves with Forrest's relations (chap. 3)
  - Deletion of multiple nodes (chap. 4)

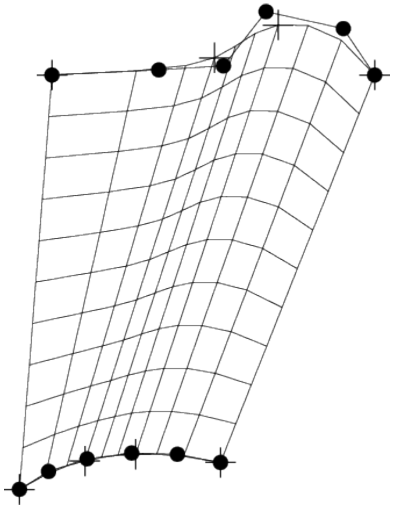
  3 – Node insertion (chap. 4)

  - Nodes of $C_0(u)$ not found in $C_1(u)$ are introduced in $C_1(u)$ and reciprocally

- These operations do not alter the geometry of the support curves
  - Excepted the parametrization if point (1) is not satisfied

NURBS surfaces

- Some examples of ruled surfaces

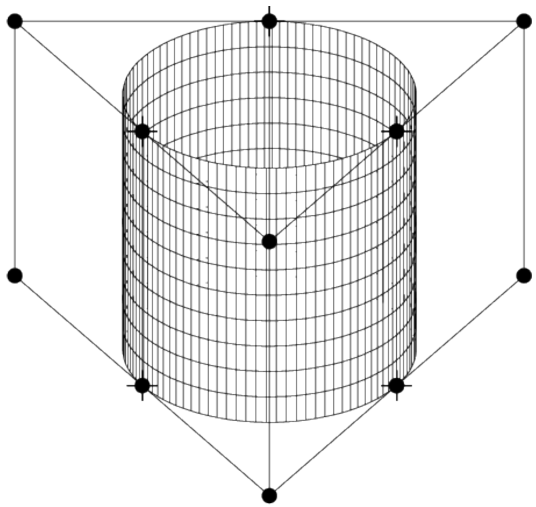


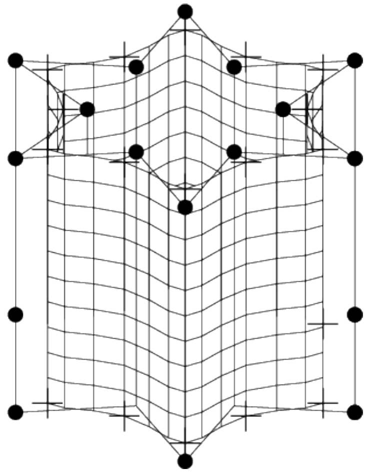
$$U=\{0,0,0,0,1,2,3,3,3,3\} \quad p=3$$

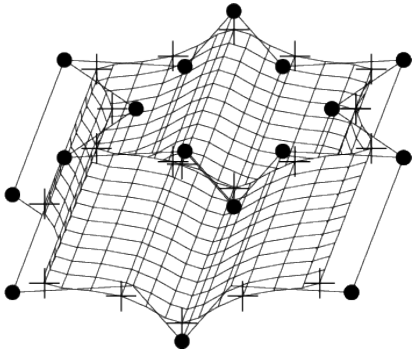
$$V=\{0,0,1,1\} \quad q=1$$

NURBS surfaces

- Cylinders





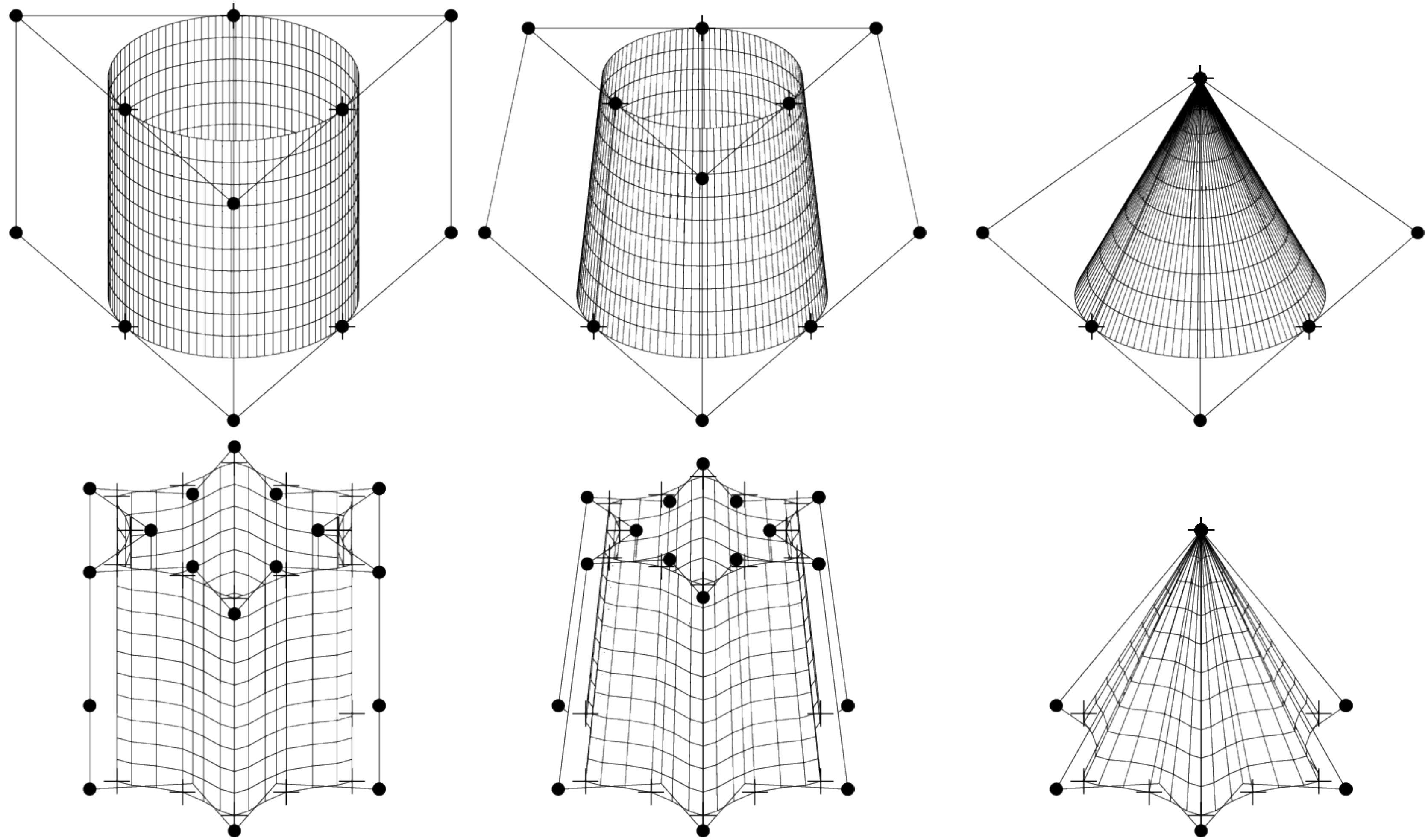


$$U=\{-3,-2,-1,0,\cdots,13,14,15\} \quad p=3$$

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

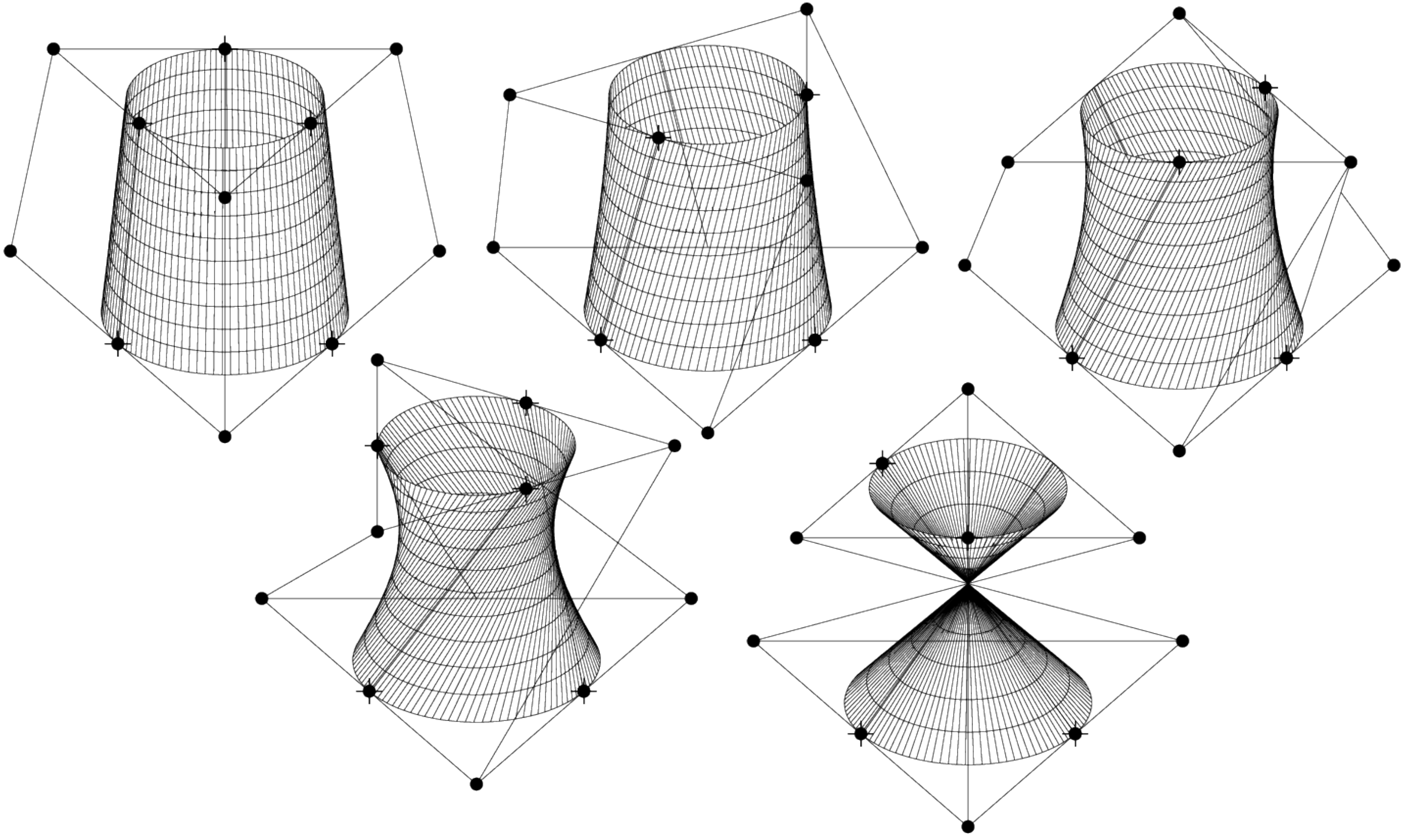
$$V=\{0,0,1,1\} \quad q=1$$

- Cones



NURBS surfaces

- Hyperboloids

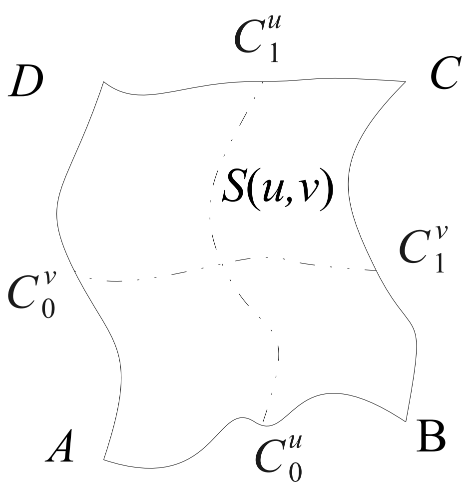


- Coons patches
  - Can we represent a Coons patch exactly with a NURBS surface ?
    - 4 boundary curves
    - Compatible ; i.e. NURBS :

      - of same nodal sequence and same degree two by two

      - nodal sequences yield curves with parameters contained between 0 and 1 (for more simplicity)

      - whose extremities are matching two by two
    - Curves C^u of nodal sequence U, degree p, n control points P^u_{ij} for C^u_j
    - Curves C^v of nodal sequence V, degree q, m control points P^v_{ij} for C^v_j



NURBS surfaces

- Coons patch = assembly of ruled surfaces

$$S_1(u,v)=(1-v)C_0^u(u)+vC_1^u(u)$$
$$S_2(u,v)=(1-u)C_0^v(v)+uC_1^v(v)$$
$$S_3(u,v)=(1-u)(1-v)A+u(1-v)B+v(1-u)D+uvC$$

- If the boundary curves are compatible NURBS curves, we can represent S_1, S_2 and S_3 as NURBS surfaces...
- Is the sum $S(u,v)=S_1(u,v)+S_2(u,v)-S_3(u,v)$ a NURBS as well ?

NURBS surfaces

- The surfaces S_1 et S_2 are ruled surfaces :

$$S_1(u,v)=(1-v)C_0^u(u)+v\,C_1^u(u)$$

$$S_2(u,v)=(1-u)C_0^v(v)+u\,C_1^v(v)$$

$$S_1(u,v)=\sum_{i=0}^{n}\sum_{j=0}^{1}N_i^p(u)N_j^1(v)P_{ij}^1$$

$$S_2(u,v)=\sum_{i=0}^{1}\sum_{j=0}^{m}N_i^1(u)N_j^q(v)P_{ij}^2$$

$$U_1=U$$
$$V_1=\{0,0,1,1\}$$

$$U_2=\{0,0,1,1\}$$
$$V_2=V$$

$$P_{ij}^1=P_{ij}^u$$

$$P_{ij}^2=P_{ji}^v$$

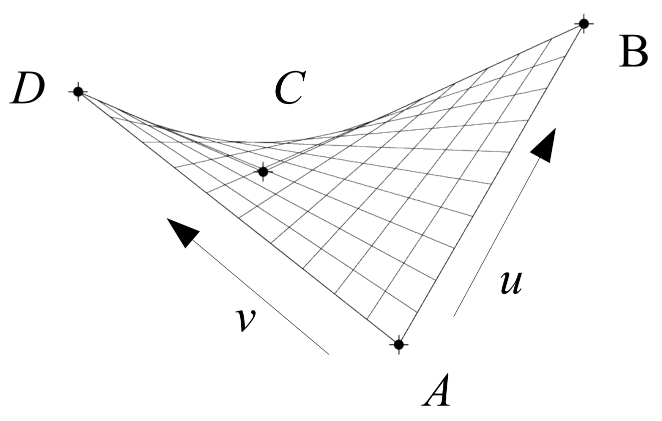
NURBS surfaces

- The surface S_3 is a bilinear patch

$$S_3(u,v)=\sum_{i=0}^{1}\sum_{j=0}^{1} N_i^1(u)\,N_j^1(v)\,P_{ij}^3$$

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

$$P_{00}^3=A$$
$$P_{10}^3=B$$
$$P_{01}^3=D$$
$$P_{11}^3=C$$



NURBS surfaces

- The « sum » between several NURBS is possible (it is a linear combination ; cf. partition of unity & affine invariance)

$$S(u,v)=S_1(u,v)+S_2(u,v)-S_3(u,v)$$

$$P_{ij} \quad =? \quad P_{ij}^1 \quad + \quad P_{ij}^2 \quad - \quad P_{ij}^3$$

- But ....

  - No conformity of the surfaces (different # of CP)

  - Different shape functions (because nodal sequences are different)

Computer Aided Design

NURBS surfaces

- The « sum » between several NURBS is possible (it is a linear combination ; cf. partition of unity & affine invariance) – if they are similar.

$$S(u,v) = S_1(u,v) + S_2(u,v) - S_3(u,v)$$

P_{ij}	P_{ij}^1	P_{ij}^2	P_{ij}^3
$U^* = ?$	$U_1 = U$	$U_2 = \{0,0,1,1\}$	$U_3 = \{0,0,1,1\}$
$V^* = ?$	$V_1 = \{0,0,1,1\}$	$V_2 = V$	$V_3 = \{0,0,1,1\}$
$p^* = ?$	$p_1 = p$	$p_2 = 1$	$p_3 = 1$
$q^* = ?$	$q_1 = 1$	$q_2 = q$	$q_3 = 1$

- Nodal sequences must correspond.

NURBS surfaces

- The « sum » between several NURBS is possible (it is a linear combination ; cf. partition of unity & affine invariance) – if they are similar.

$U^* = U$
$V^* = V$
$p^* = p$
$q^* = q$

$U_1 = U$
$V_1 = \{0, 0, 1, 1\}$
$p_1 = p$
$q_1 = 1$

$U_2 = \{0, 0, 1, 1\}$
$V_2 = V$
$p_2 = 1$
$q_2 = q$

$U_3 = \{0, 0, 1, 1\}$
$V_3 = \{0, 0, 1, 1\}$
$p_3 = 1$
$q_3 = 1$

- degree elevation $\rightarrow q_1 = q$ | degree elevation $\rightarrow p_2 = p$ | degree elevation $\rightarrow p_3 = p\,,\ q_3 = q$
- Then, node insertions $\rightarrow V_1 = V$ | $\rightarrow U_2 = U$ | $\rightarrow U_3 = U\,,\ V_3 = V$

NURBS surfaces

- Each operation (degree elevation or node insertion) adds control points so as to make “compatible” surfaces
  - Finally, one can write

$$S(u,v)=S_1(u,v)+S_2(u,v)-S_3(u,v)$$

$$P_{ij}^{*} \quad = \quad P_{ij}^{1*} \quad + \quad P_{ij}^{2*} \quad - \quad P_{ij}^{3*}$$

$U^*=U$	$U_1^*=U$	$U_2^*=U$	$U_3^*=U$
$V^*=V$	$V_1^*=V$	$V_2^*=V$	$V_3^*=V$
$p^*=p$	$p_1^*=p$	$p_2^*=p$	$p_3^*=p$
$q^*=q$	$q_1^*=q$	$q_2^*=q$	$q_3^*=q$

NURBS surfaces

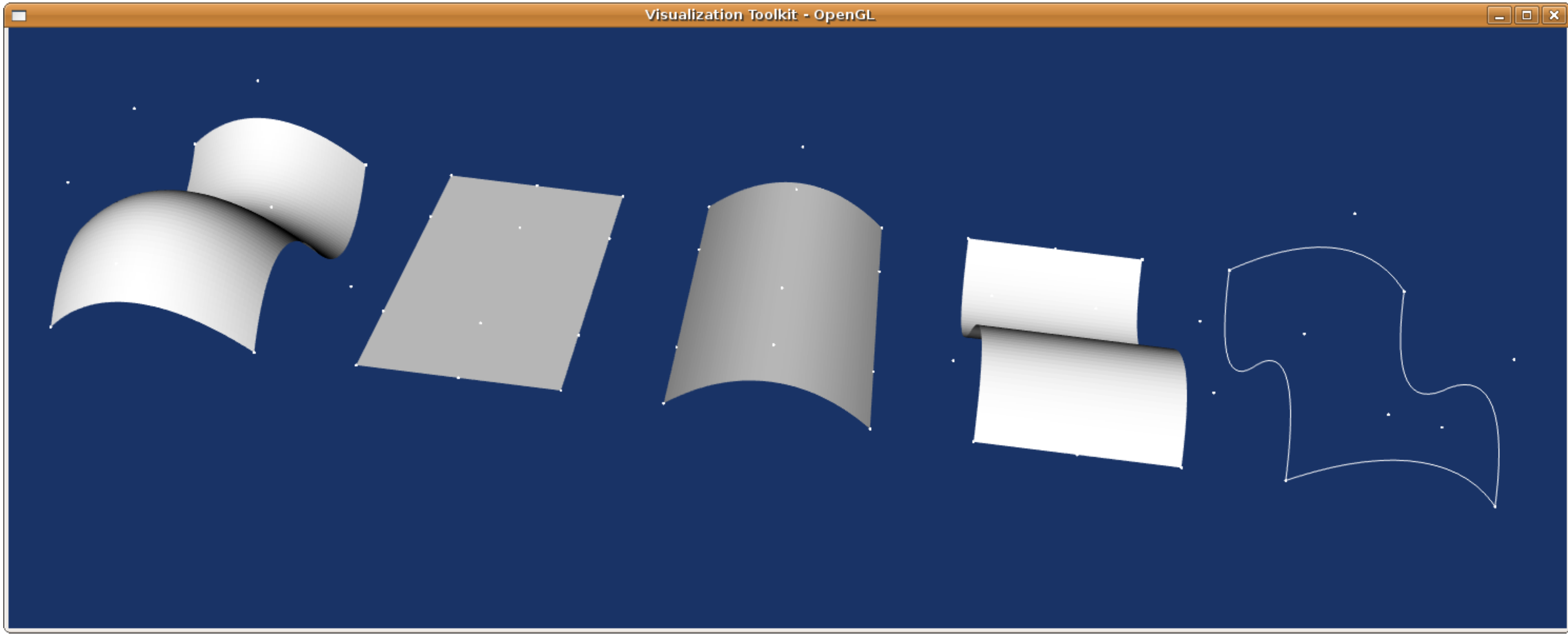
- Degree elevation (in u or v) of a surface whose nodal sequence is that of a Bézier curve :
  - Identical to the degree elevation ease of a Bézier curve
  - Forrest relations written on the set of control points

$$
\begin{aligned}
&\text{for } j=0\cdots q\\
&Q_{0j}=P_{0j}\\
&\text{for } i=1\cdots p \quad Q_{ij}=P_{i-1,j}+\frac{(p+1-i)}{(p+1)}(P_{ij}-P_{i-1,j})\\
&Q_{p+1,j}=P_{pj}
\end{aligned}
$$

- The nodal sequence is then augmented
- Node insertions in a B-Spline surface
  - see chapter 5

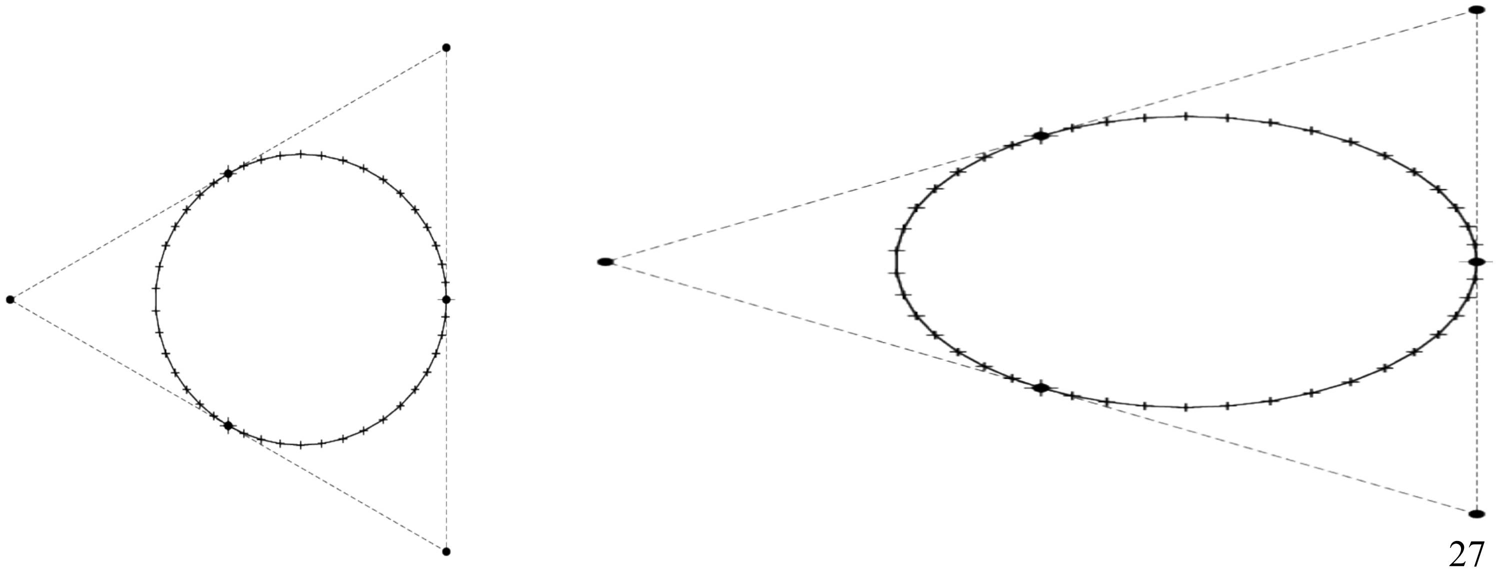
Computer Aided Design

NURBS surfaces



NURBS surfaces

- Global modification of curves / surfaces
  - Affine transformation of control points
  - The affine invariance assures us that the resulting curve is what we want.
    - Ex. Ellipse from a circle – scaling in a single direction.



(some) advanced algorithms

NURBS surfaces

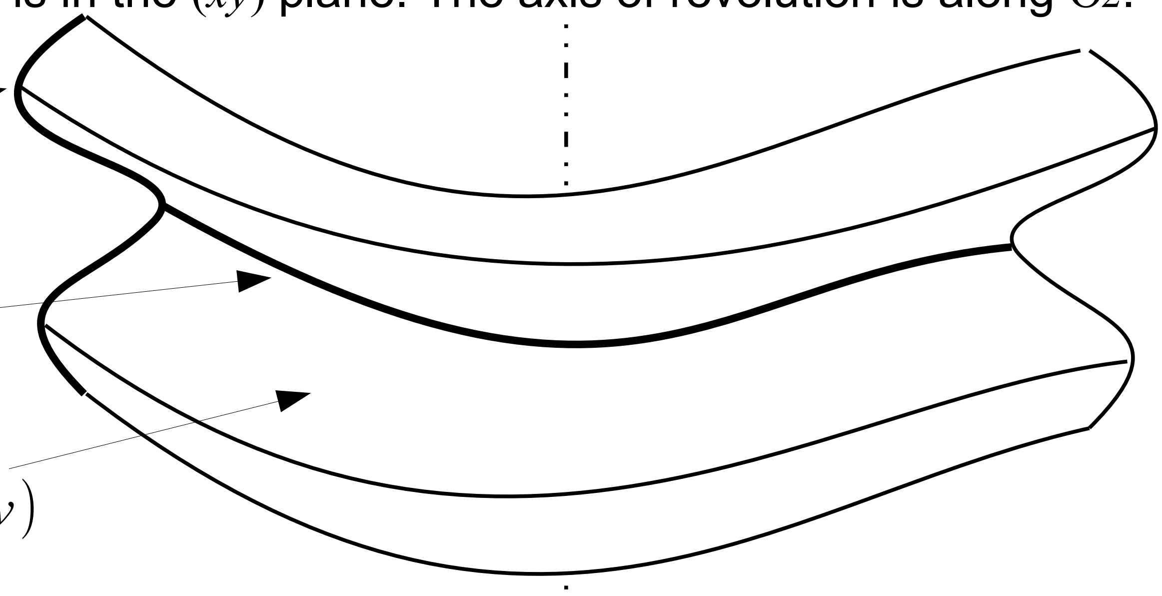
- Profiled surfaces

a) Generalization of the surface of revolution

- Each point of a generating curve (the profile curve) follows a trajectory whose radius is defined by a second curve (the trajectory curve)
- We assume without loss of generality that $P(u)$ is in the (xz) plane , and that $T(v)$ is in the (xy) plane. The axis of revolution is along Oz.

$$P(u)=\begin{pmatrix} x^p(u) \\ 0 \\ z^p(u) \end{pmatrix}$$

$$T(v)=\begin{pmatrix} x^t(v) \\ y^t(v) \\ 0 \end{pmatrix}$$

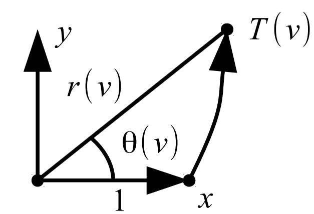


$$S(u,v)$$

NURBS surfaces

- Generalization of surfaces of revolution
  - Lets transform T to polar coordinates : it corresponds to a simple rotation around z + a uniform scaling in x-y (not z) :

$$T(v)=\begin{pmatrix} x^t(v) \\ y^t(v) \\ 0 \end{pmatrix}=\begin{pmatrix} r(v)\cos\theta(v) \\ r(v)\sin\theta(v) \\ 0 \end{pmatrix}$$



  - The related transformation matrix is therefore :

$$M(v)=S(v)\cdot R(v)=\begin{pmatrix} r & 0 & 0 \\ 0 & r & 0 \\ 0 & 0 & 1 \end{pmatrix}\cdot\begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}=\begin{pmatrix} r\cos\theta & -r\sin\theta & 0 \\ r\sin\theta & r\cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

  - Let's apply this to P :

$$P(u)=\begin{pmatrix} x^p(u) \\ 0 \\ z^p(u) \end{pmatrix} \rightarrow S(u,v)=M(v)\cdot P(u)=\begin{pmatrix} x^p(u)\cdot r(v)\cos\theta(v) \\ x^p(u)\cdot r(v)\sin\theta(v) \\ z^p(u) \end{pmatrix}=\begin{pmatrix} x^p(u)\cdot x^t(v) \\ x^p(u)\cdot y^t(v) \\ z^p(u) \end{pmatrix}$$

NURBS surfaces

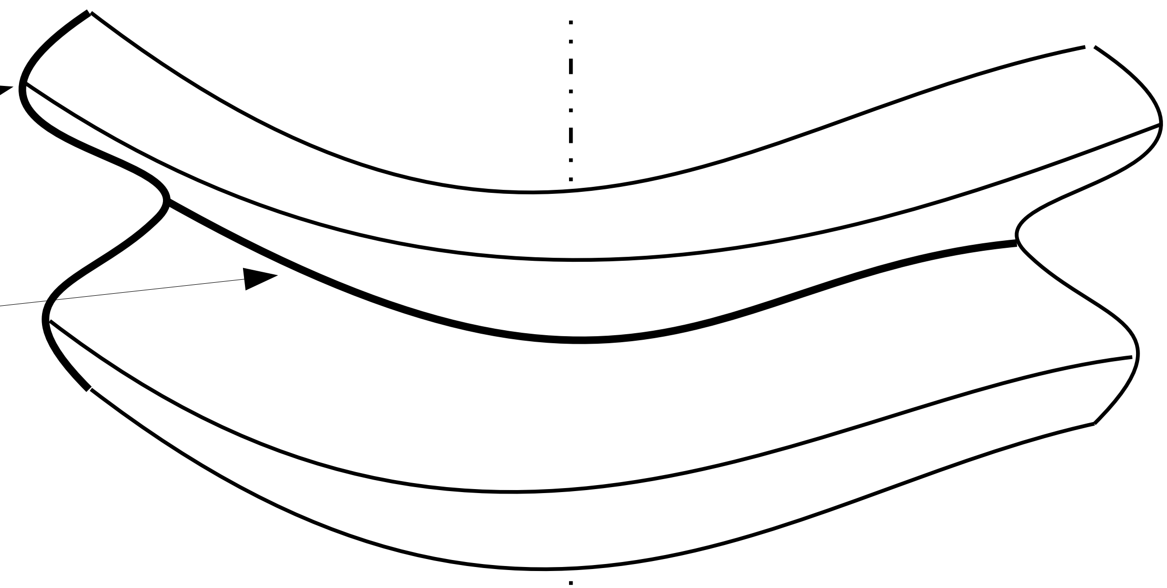
- Generalization of surfaces of revolution
  - The analytical expression of the surface is therefore simply:

$$S(u,v)=\begin{pmatrix} x^p(u)\cdot x^t(v) \\ x^p(u)\cdot y^t(v) \\ z^p(u) \end{pmatrix}$$

  - Can we express it as a NURBS ?

$$P(u)=\begin{pmatrix} x^p(u) \\ 0 \\ z^p(u) \end{pmatrix}$$

$$T(v)=\begin{pmatrix} x^t(v) \\ y^t(v) \\ 0 \end{pmatrix}$$



NURBS surfaces

- Generalization of surfaces of revolution
  - New control points are located with reference to the z axis
  - We have to deal with homogeneous coordinates

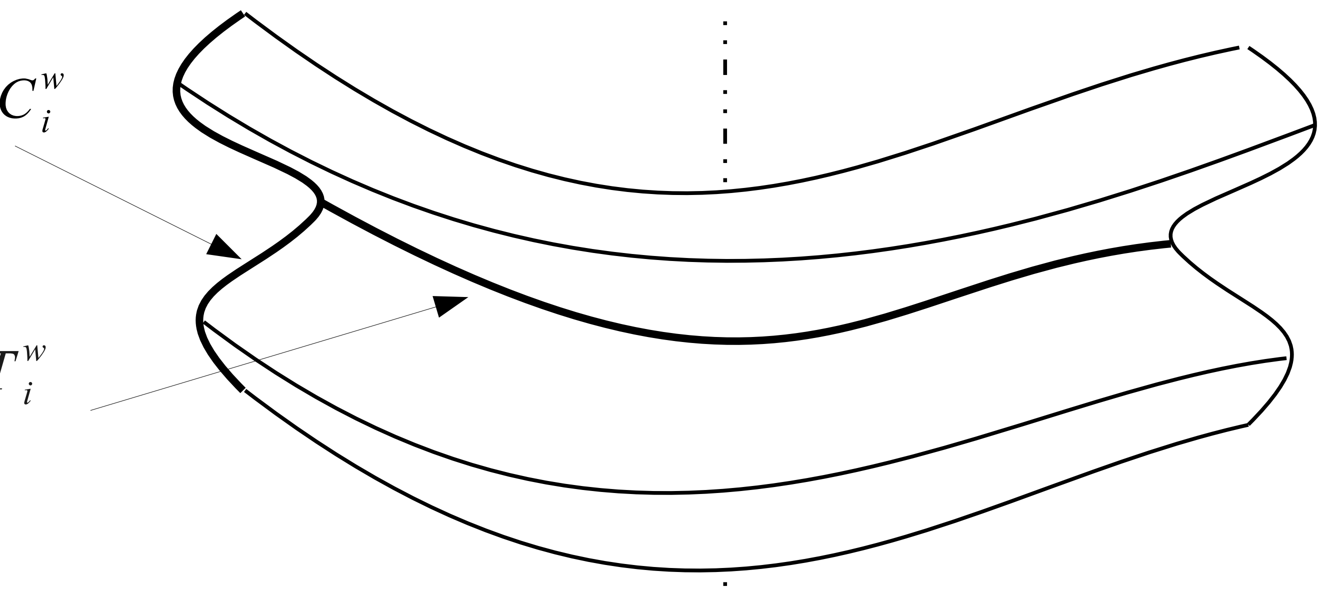
$$S^w(u,v)=\sum_{i=0}^{n}\sum_{j=0}^{m} N_i^p(u)\,N_j^q(v)\,P_{ij}^w$$

$$C^w(u)=\sum_{i=0}^{n} N_i^p(u)\,C_i^w$$

$$U=\{u_0,\cdots,u_r\}$$

$$T^w(v)=\sum_{i=0}^{m} N_i^q(v)\,T_i^w$$

$$V=\{v_0,\cdots,v_s\}$$



Computer Aided Design

NURBS surfaces

$$S(u,v)=\begin{pmatrix} x^p(u)\cdot x^t(v) \\ x^p(u)\cdot y^t(v) \\ z^p(u) \end{pmatrix} \equiv \begin{pmatrix} x^p(u)x^t(v)w^p(u)w^t(v) \\ x^p(u)y^t(v)w^p(u)w^t(v) \\ z^p(u)w^p(u)w^t(v) \\ w^p(u)w^t(v) \end{pmatrix}$$

$$C^w(u)=\sum_{i=0}^{n} N_i^p(u)C_i^w=\begin{pmatrix} x^p(u)w^p(u) \\ 0 \\ z^p(u)w^p(u) \\ w^p(u) \end{pmatrix}=\begin{pmatrix} \sum_{i=0}^{n} N_i^p(u)x_i^p w_i^p \\ 0 \\ \sum_{i=0}^{n} N_i^p(u)z_i^p w_i^p \\ \sum_{i=0}^{n} N_i^p(u)w_i^p \end{pmatrix}$$

n+1 control points

$$T^w(v)=\sum_{j=0}^{m} N_j^q(v)T_j^w=\begin{pmatrix} x^t(v)w^t(v) \\ y^t(v)w^t(v) \\ 0 \\ w^t(v) \end{pmatrix}=\begin{pmatrix} \sum_{j=0}^{m} N_j^q(v)x_j^t w_j^t \\ \sum_{j=0}^{m} N_j^q(v)y_j^t w_j^t \\ 0 \\ \sum_{j=0}^{m} N_j^q(v)w_j^t \end{pmatrix}$$

m+1 control points

- Determination of the CPs

$$x^p(u)x^t(v)w^p(u)w^t(v)$$
$$=\sum_{i=0}^{n} N_i^p(u)x_i^p w_i^p\cdot\sum_{j=0}^{m} N_j^q(v)x_j^t w_j^t$$
$$=\sum_{i=0}^{n}\sum_{j=0}^{m} N_i^p(u)N_j^q(v)x_i^p w_i^p x_j^t w_j^t$$

Same for the other coordinates :

$$S^w(u,v)=\sum_{i=0}^{n}\sum_{j=0}^{m} N_i^p(u)N_j^q(v)\begin{pmatrix} x_i^p x_j^t w_i^p w_j^t \\ x_i^p y_j^t w_i^p w_j^t \\ z_i^p w_i^p w_j^t \\ w_i^p w_j^t \end{pmatrix}$$

$(n+1).(m+1)$ control points

NURBS surfaces

- Initial data

$$C^w(u)=\sum_{i=0}^{n} N_i^p(u)\,C_i^w \qquad C_i^w=\begin{pmatrix} x_i^p w_i^p \\ 0 \\ z_i^p w_i^p \\ w_i^p \end{pmatrix}$$

$$U=\{u_0,\cdots,u_r\}$$

$$T^w(v)=\sum_{i=0}^{m} N_i^q(v)\,T_i^w \qquad T_j^w=\begin{pmatrix} x_j^t w_j^t \\ y_j^t w_j^t \\ 0 \\ w_j^t \end{pmatrix}$$

$$V=\{v_0,\cdots,v_s\}$$

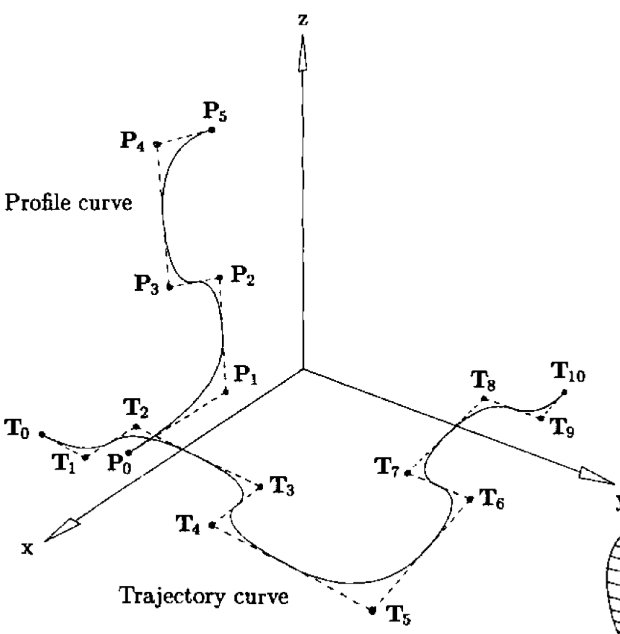
- The surface is expressed :

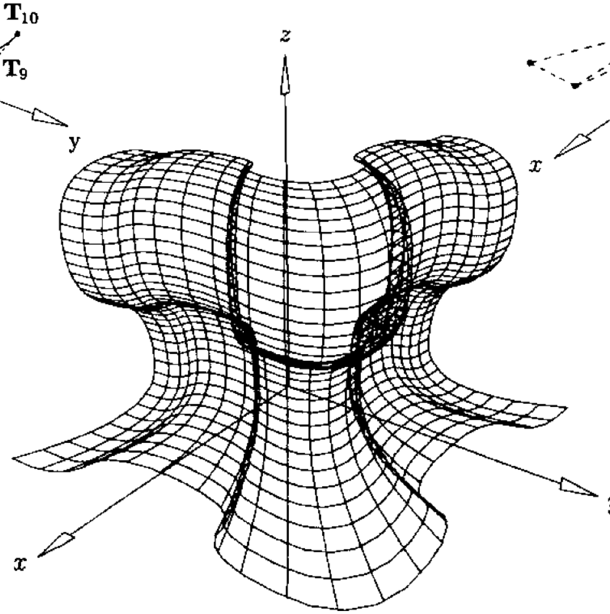
$$S^w(u,v)=\sum_{i=0}^{n}\sum_{j=0}^{m} N_i^p(u)\,N_j^q(v)\,P_{ij}^w \qquad P_{ij}^w=\begin{pmatrix} x_i^p x_j^t w_i^p w_j^t \\ x_i^p y_j^t w_i^p w_j^t \\ z_i^p w_i^p w_j^t \\ w_i^p w_j^t \end{pmatrix}$$

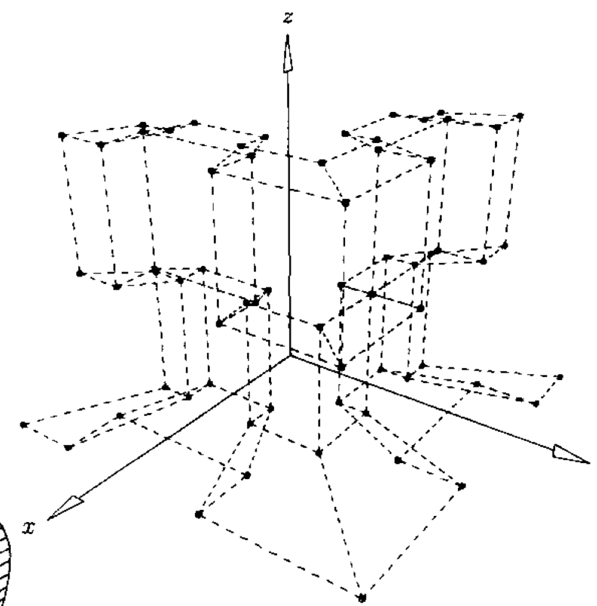
$$U=\{u_0,\cdots,u_r\} \qquad V=\{v_0,\cdots,v_s\}$$

Computer Aided Design

NURBS surfaces





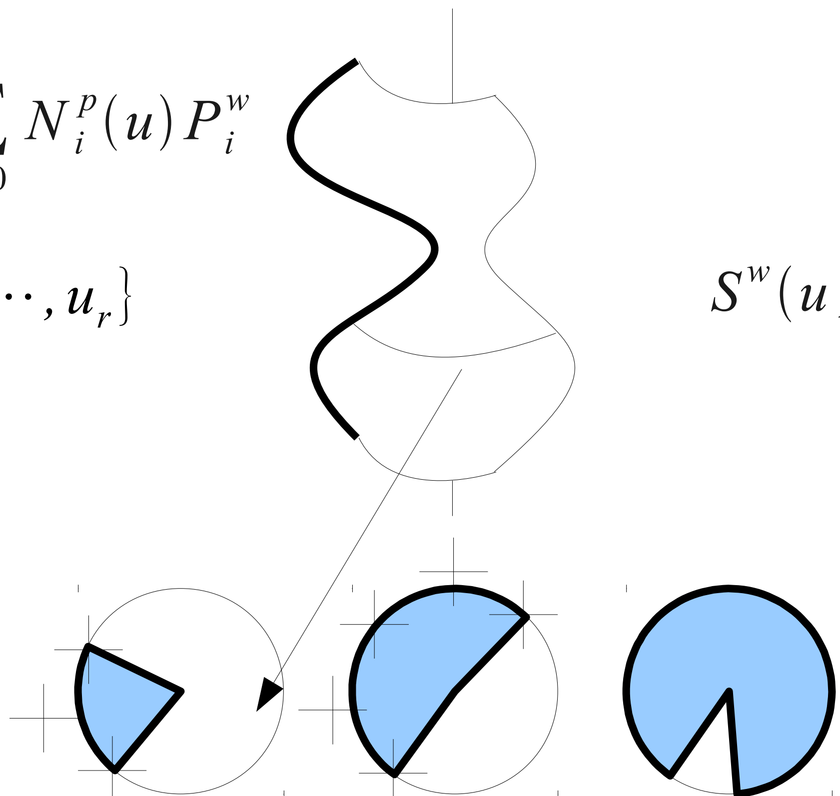


L. Piegl « the NURBS book »

- Surface of revolution
  - Let us have a curve (generating curve) that we want to revolve around an axis W, by a certain angle α.

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

$$U=\{u_0,\cdots,u_r\}$$



$$S^w(u,v)=\sum_{j=0}^{m} N_j^2(v)\,Q_j^w(u)$$

m=2 if $\alpha<=2\pi/3$ (1 segment, 3 CP)

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

m=4 if $2\pi/3<\alpha<=4\pi/3$ (2segments, 5 CP)

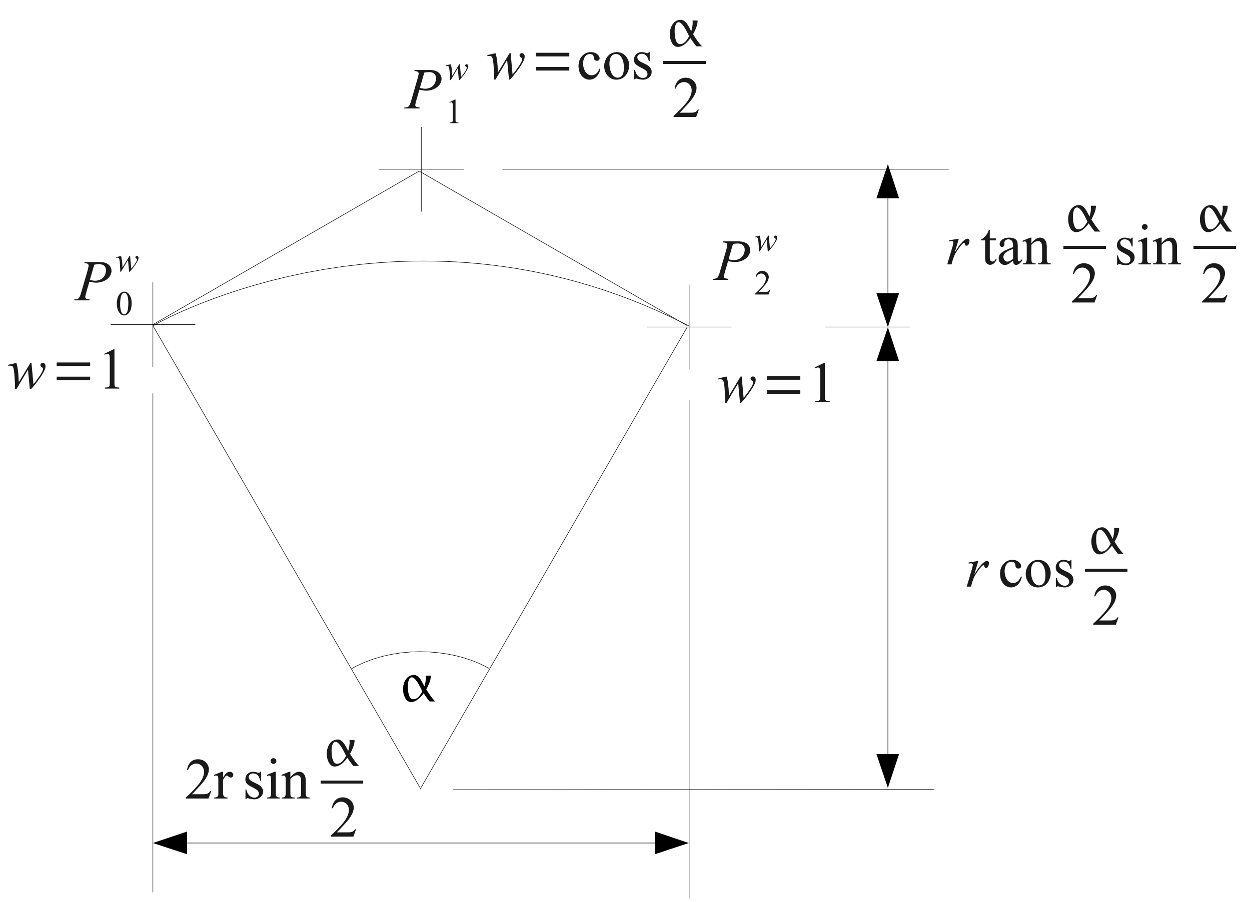
$$V=\{0,0,0,1,1,2,2,2\}$$

m=6 if $4\pi/3<\alpha<=2\pi$ (3segments, 7 CP)

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

NURBS surfaces

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



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

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

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

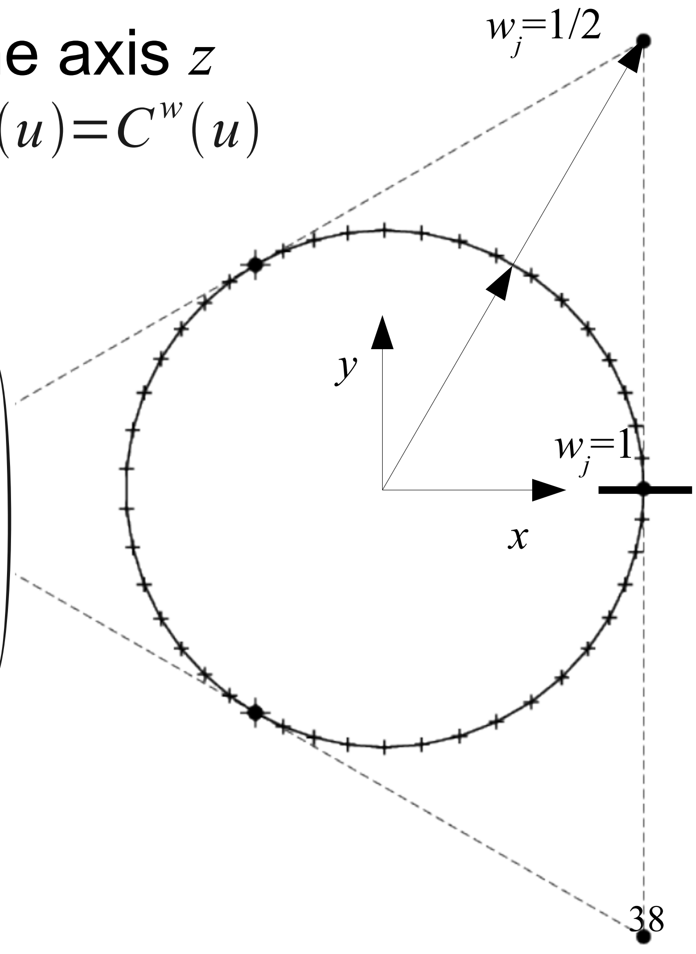
NURBS surfaces

- Without loss of generality, let's assume that
  - $\alpha=2\pi$
  - A rotation axis coincident with the axis z
  - Curve C lies in the plane xz : $Q_0^w(u)=C^w(u)$
- Computation of the points $Q_j^w(u)$

$$Q_0^w(u)=\begin{pmatrix} x(u)\cdot w(u) \\ 0\cdot w(u) \\ z(u)\cdot w(u) \\ w(u) \end{pmatrix} \quad Q_1^w(u)=\begin{pmatrix} 2x\cos\pi/3\cdot w\cdot 1/2 \\ 2x\sin\pi/3\cdot w\cdot 1/2 \\ z\cdot w\cdot 1/2 \\ w\cdot 1/2 \end{pmatrix}$$

$$Q_2^w(u)=\begin{pmatrix} x\cos 2\pi/3\cdot w \\ x\sin 2\pi/3\cdot w \\ z\cdot w \\ w \end{pmatrix}$$

etc...



Computer Aided Design

NURBS surfaces

- Definition as a NURBS

$$S^w(u,v)=\sum_{j=0}^{m} N_j^2(v)\,Q_j^w(u)=\sum_{j=0}^{m} N_j^2(v)\sum_{i=0}^{n} N_i^p(u)\,P_{ij}^w$$

Rotation + scaling of the curve = Rotation + scaling of control points of the curve

  - The operation is possible because NURBS curves are invariant by affine transformations

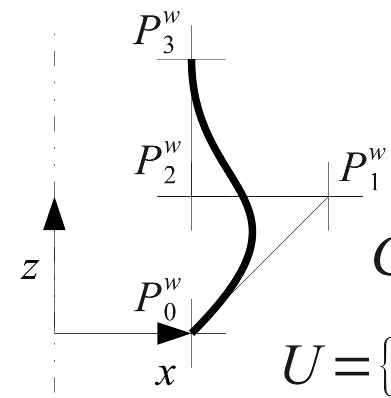
$$S^w(u,v)=\sum_{i=0}^{n}\sum_{j=0}^{m} N_i^p(u)\,N_j^2(v)\,P_{ij}^w$$

Computer Aided Design

NURBS surfaces

- Example - revolution of 90° of a curve around the axis z :

$$P_0^w = \begin{pmatrix}1\\0\\0\\1\end{pmatrix} \quad P_1^w = \begin{pmatrix}2\\0\\1\\1\end{pmatrix} \quad P_2^w = \begin{pmatrix}1\\0\\1\\1\end{pmatrix} \quad P_3^w = \begin{pmatrix}1\\0\\2\\1\end{pmatrix}$$

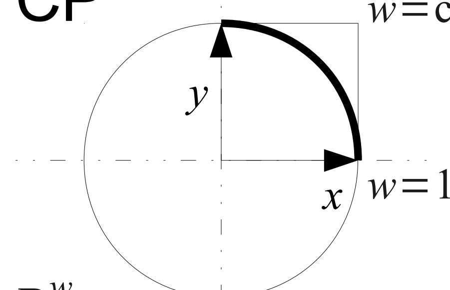


$$C^w(u) = \sum_{i=0}^{3} N_i^3(u) P_i^w$$

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

- Calculation of circle's parameters
- Rotation / scaling of CP

$$w = \cos\frac{\alpha}{2} = \frac{\sqrt{2}}{2}$$



$$S^w(u,v) = \sum_{i=0}^{3} \sum_{j=0}^{2} N_i^3(u) N_j^2(v) P_{ij}^w$$

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

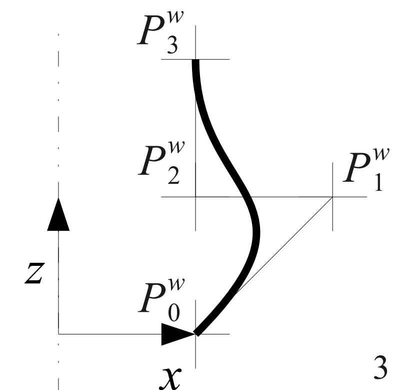
NURBS surfaces

- Example - revolution of 90° of a curve around the axis z :

$$P^w_{00}=\begin{pmatrix}1\\0\\0\\1\end{pmatrix} P^w_{10}=\begin{pmatrix}2\\0\\1\\1\end{pmatrix} P^w_{20}=\begin{pmatrix}1\\0\\1\\1\end{pmatrix} P^w_{30}=\begin{pmatrix}1\\0\\2\\1\end{pmatrix}$$

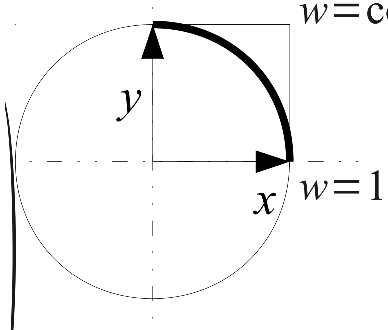
$$P^w_{02}=\begin{pmatrix}0\\1\\0\\1\end{pmatrix} P^w_{12}=\begin{pmatrix}0\\2\\1\\1\end{pmatrix} P^w_{22}=\begin{pmatrix}0\\1\\1\\1\end{pmatrix} P^w_{32}=\begin{pmatrix}0\\1\\2\\1\end{pmatrix}$$

$$P^w_{01}=\begin{pmatrix}w\\w\\0\\w\end{pmatrix} P^w_{11}=\begin{pmatrix}2w\\2w\\w\\w\end{pmatrix} P^w_{21}=\begin{pmatrix}w\\w\\w\\w\end{pmatrix} P^w_{31}=\begin{pmatrix}w\\w\\2w\\w\end{pmatrix}$$



$$S^w(u,v)=\sum_{i=0}^{3}\sum_{j=0}^{2} N_i^3(u)\,N_j^2(v)\,P^w_{ij}$$

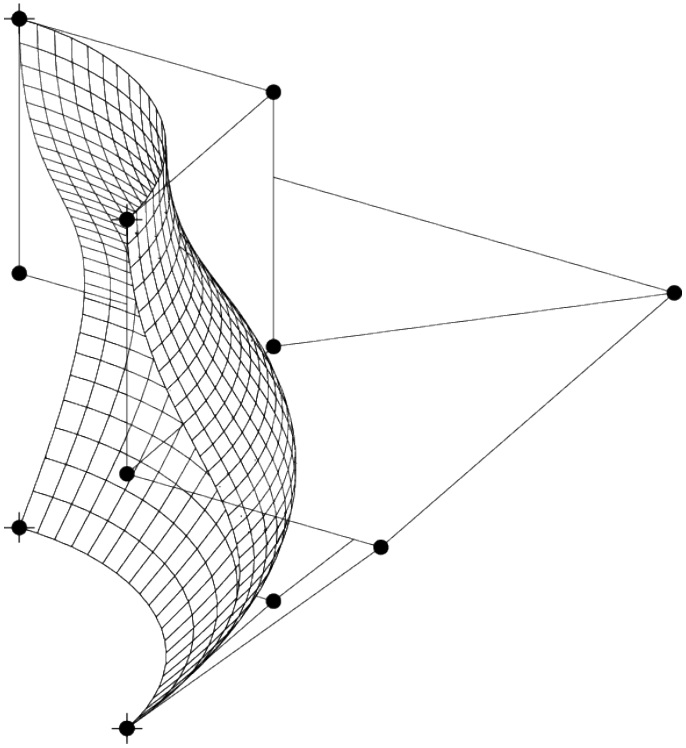
$$w=\cos\frac{\alpha}{2}=\frac{\sqrt{2}}{2}$$



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

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

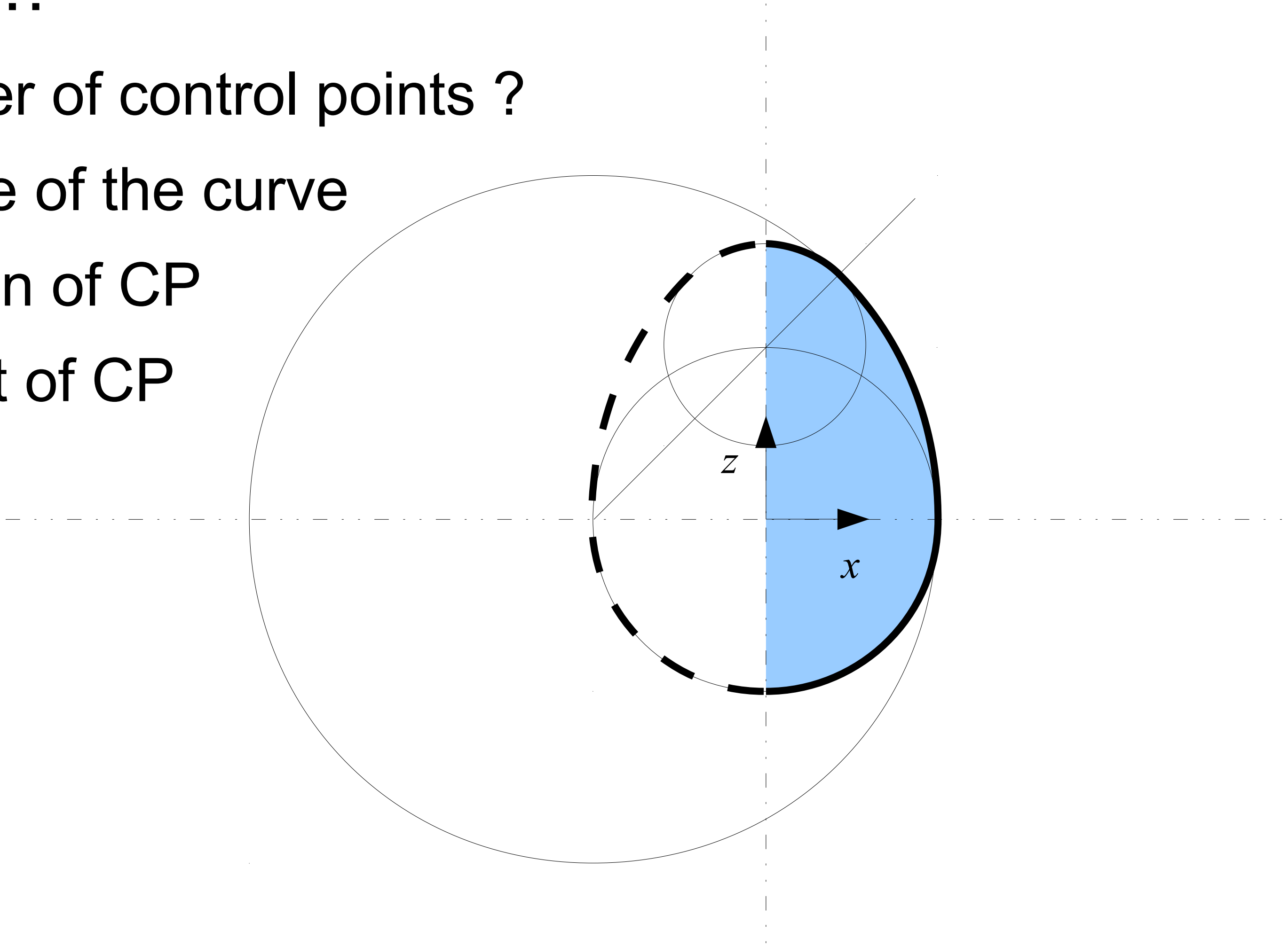
NURBS surfaces



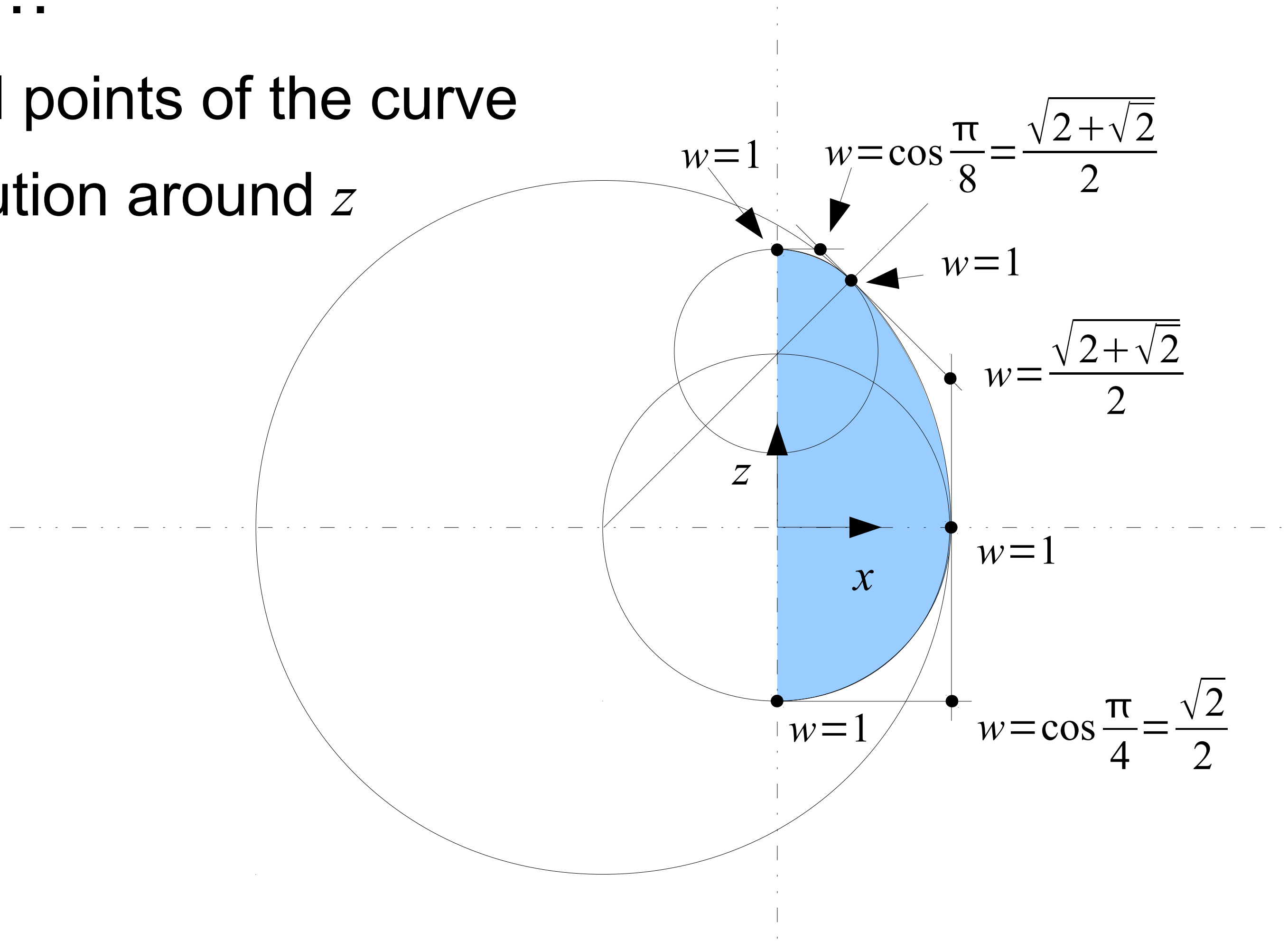
Computer Aided Design

NURBS surfaces

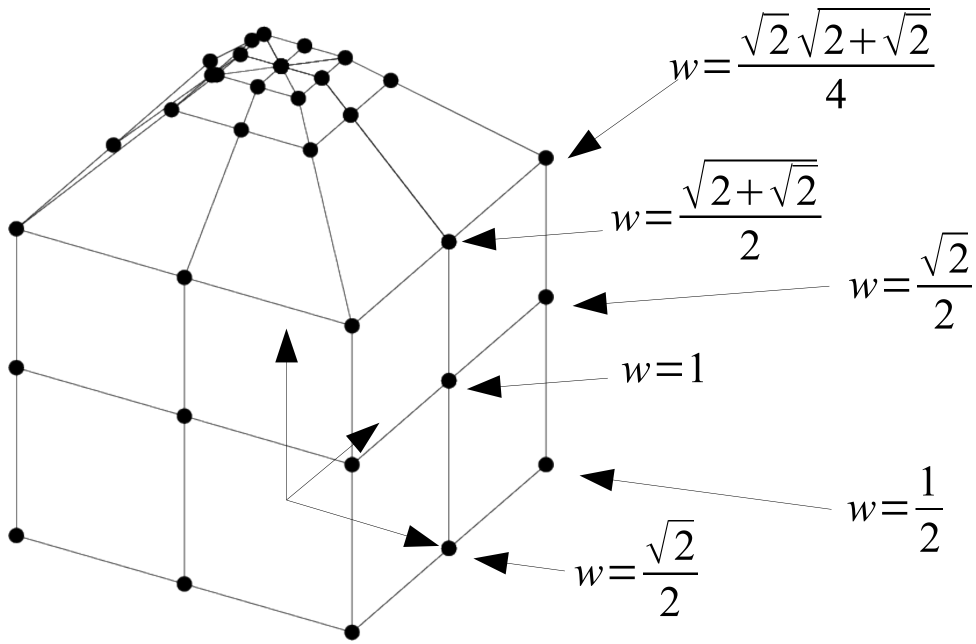
- An egg …
  - Number of control points ?
  - Degree of the curve
  - Position of CP
  - Weight of CP

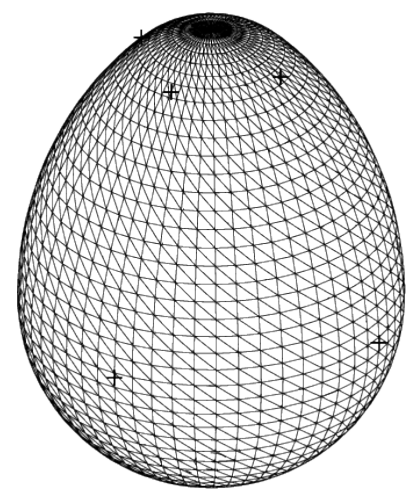


- An egg …
  - control points of the curve
  - Revolution around z



NURBS surfaces





- Profiled surfaces

b) profile with a controlled section obtained by sweeping

- same scheme :

  - curved trajectory
  - Section curve
  - with an **orientation matrix**: $M(v)$

- The “analytic” surface is

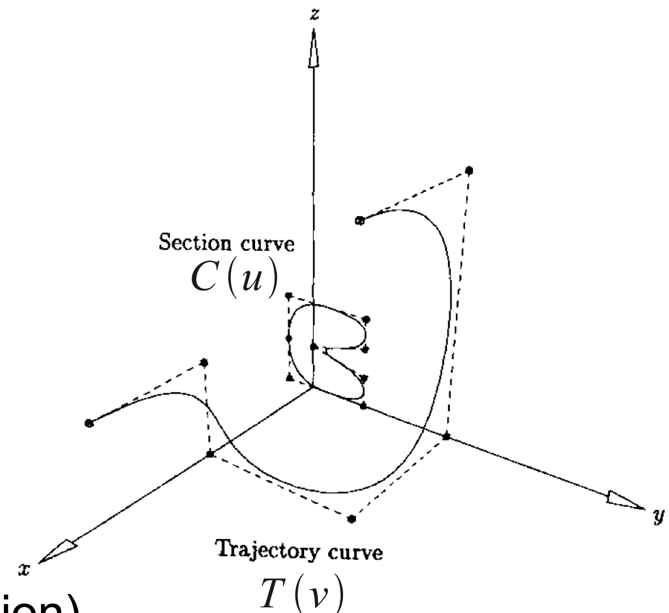
$$S(u,v) = T(v) + M(v)\,C(u)$$

- Two possibilities

  1- $M(v)$ is an identity (constant orientation)

  2- $M(v)$ depends on the trajectory

  In these two cases, $M(v)$ does **not** correspond to a generalized rotation (no fixed axis of rotation)



L. Piegl « the NURBS book »

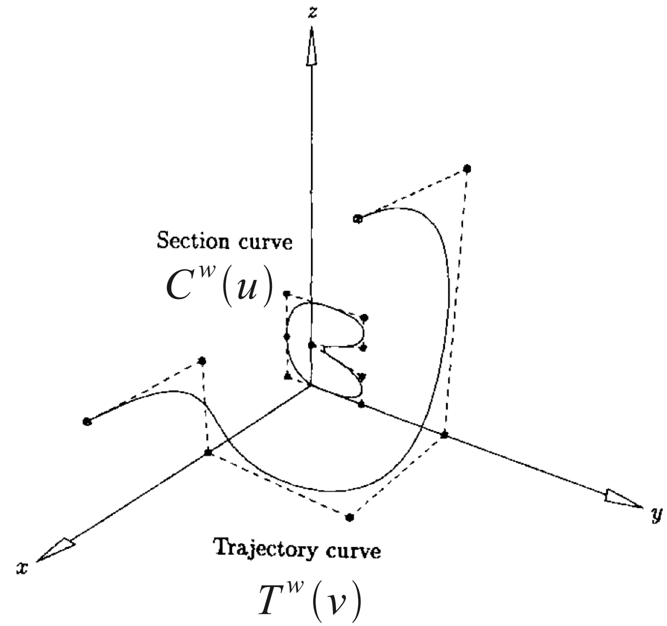
NURBS surfaces

- Case 1 : when $M(v)$ is an identity : $S(u,v)=T(v)+C(u)$

The section is simply moved without changing the orientation.

$$C^w(u)=\sum_{i=0}^{n} N_i^p(u)\,C_i^w=\begin{pmatrix}\sum_{i=0}^{n} N_i^p(u)\,x_i^c w_i^c\\ \vdots\\ \sum_{i=0}^{n} N_i^p(u)\,w_i^c\end{pmatrix}$$

$$T^w(v)=\sum_{i=0}^{m} N_i^q(v)\,T_i^w=\begin{pmatrix}\sum_{i=0}^{m} N_i^q(u)\,x_i^t w_i^t\\ \vdots\\ \sum_{i=0}^{n} N_i^q(u)\,w_i^t\end{pmatrix}$$



NURBS surfaces

$$S(u,v)=T(v)+C(u)$$

$$C^w(u)=\sum_{i=0}^{n} N_i^p(u)\,C_i^w=\begin{pmatrix}\sum_{i=0}^{n} N_i^p(u)\,x_i^c w_i^c\\ \vdots\\ \sum_{i=0}^{n} N_i^p(u)\,w_i^c\end{pmatrix}\qquad T^w(v)=\sum_{i=0}^{m} N_j^q(v)\,T_i^w=\begin{pmatrix}\sum_{j=0}^{m} N_i^q(v)\,x_i^t w_i^t\\ \vdots\\ \sum_{j=0}^{n} N_j^q(v)\,w_i^t\end{pmatrix}$$

$$x^p(u)+x^t(v)=\frac{\sum_{i=0}^{n} N_i^p(u)\,x_i^c w_i^c}{\sum_{i=0}^{n} N_i^p(u)\,w_i^c}+\frac{\sum_{j=0}^{m} N_j^q(v)\,x_j^t w_j^t}{\sum_{j=0}^{m} N_j^q(v)\,w_j^t}$$

$$=\frac{\sum_{i=0}^{n} N_i^p(u)\,x_i^c w_i^c\cdot\sum_{j=0}^{m} N_j^q(v)\,w_j^t+\sum_{i=0}^{n} N_i^p(u)\,w_i^c\cdot\sum_{j=0}^{m} N_j^q(v)\,x_j^t w_j^t}{\sum_{i=0}^{n} N_i^p(u)\,w_i^c\cdot\sum_{j=0}^{m} N_j^q(v)\,w_j^t}$$

$$=\frac{\sum_{i=0}^{n}\sum_{j=0}^{m} N_i^p(u)\,N_j^q(v)\,x_i^c w_i^c w_j^t+\sum_{i=0}^{n}\sum_{j=0}^{m} N_i^p(u)\,N_j^q(v)\,x_j^t w_i^c w_j^t}{\sum_{i=0}^{n}\sum_{j=0}^{m} N_i^p(u)\,N_j^q(v)\,w_i^c w_j^t}=\frac{\sum_{i=0}^{n}\sum_{j=0}^{m} N_i^p(u)\,N_j^q(v)\,(x_i^c+x_j^t)\,w_i^c w_j^t}{\sum_{i=0}^{n}\sum_{j=0}^{m} N_i^p(u)\,N_j^q(v)\,w_i^c w_j^t}$$

$n.m$ homogeneous coordinates of control points

Associated weight

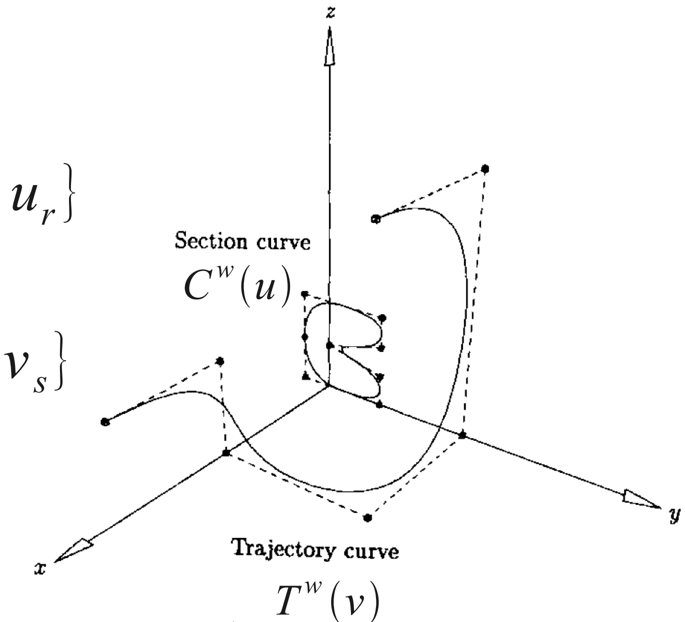
NURBS surfaces

- Case 1 : $M(v)$ is an identity : $S(u,v)=T(v)+C(u)$

$$C^w(u)=\sum_{i=0}^{n} N_i^p(u)\,C_i^w \qquad U=\{u_0,\cdots,u_r\}$$

$$T^w(v)=\sum_{i=0}^{m} N_i^q(v)\,T_i^w \qquad V=\{v_0,\cdots,v_s\}$$

$$S^w(u,v)=\sum_{i=0}^{n}\sum_{j=0}^{m} N_i^p(u)\,N_j^q(v)\,P_{ij}^w$$



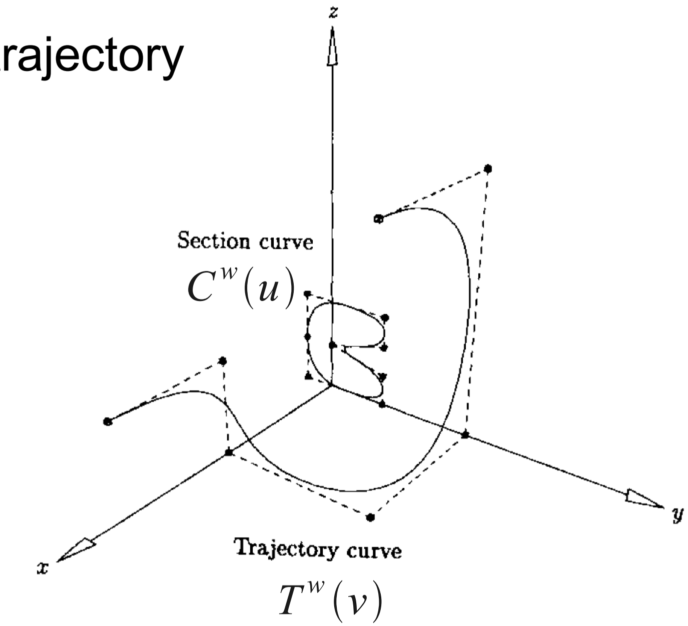
$$C_i^w=\begin{pmatrix} x_i^c w_i^c \\ z_i^c w_i^c \\ z_i^c w_i^c \\ w_i^c \end{pmatrix} \quad T_j^w=\begin{pmatrix} x_j^t w_j^t \\ y_j^t w_j^t \\ z_j^t w_j^t \\ w_j^t \end{pmatrix} \quad P_{ij}^w=\begin{pmatrix} (x_i^c+x_j^t)\,w_i^c w_j^t \\ (y_i^c+y_j^t)\,w_i^c w_j^t \\ (z_i^c+z_j^t)\,w_i^c w_j^t \\ w_i^c w_j^t \end{pmatrix}$$

Computer Aided Design

NURBS surfaces

- Case 2 : $M(v)$ is imposed : $S(u,v)=T(v)+M(v)C(u)$

  Purpose : align the section along the trajectory curve



- Determination of $M(v)$
  - Global coordinates : $\{O,X,Y,Z\}$
  - Local coordinates along $T(v)$:

    $\{o(v), x(v), y(v), z(v)\}$

    $o(v)=T(v)$

    $x(v)=\dfrac{T'(v)}{|T'(v)|}$ (tangent vector)

  - Let $B(v)$ a vectorial function satisfying $B(v)\cdot x(v)=0\ \forall v$, that will be computed later. It will serve as a reference axis to set the orientation of the section curve along the trajectory :

$$z(v)=\frac{B(v)}{|B(v)|} \qquad y(v)=z(v)\times x(v)$$

NURBS surfaces

- $M(v)$ is a matrix that allows to transform the coordinates from $\{o(v), x(v), y(v), z(v)\}$ to $\{O, X, Y, Z\}$ (trivial)
- This problem is that $M(v)$ does not lead to a NURBS surface in the general case, because the dependence in v is arbitrary.
- The surface that we want to build is therefore an **approximation**.

$$S(u,v)=T(v)+M(v)C(u)$$

$$\tilde{S}^w(u,v)=\sum_{i=0}^{n}\sum_{j=0}^{m}N_i^p(u)N_j^q(v)P_{ij}^w$$

- How to determine the P_{ij} ?

NURBS surfaces

- Two techniques (among others)

  1) With the algebraic form $S(u,v) = T(v) + M(v)C(u)$, generate a grid of $n \times m$ points exactly on $S(u,v)$. By interpolation, determine positions of CP of a surface passing by these points (not described here)

  - Disadvantage : no isovalues of $\hat{S}$ according to u or v is exactly on S

  2) By interpolating many instances of the section (oriented appropriately by M) along the trajectory, using a technique known as « skinning » (described in the sequel)

  - Allows to interpolate exactly the trajectory and the instances of the profile at nodes v_i – (but the surface remains an approximation everywhere else)

NURBS surfaces

The technique described here :

- We place many instances of the section along the trajectory. These are oriented appropriately by $M(v)$.
  $C_k(u) \quad , \quad k=0\cdots K$
- We then build a surface (skin) interpolating exactly these instances
- The $C_k(u)$ are therefore isoparametrics of the skin $P(u,v)$ for constant values of v ,
  moreover, they are NURBS : $C_k^w(u)=\sum_{i=0}^{n} N_i^p(u)\, C_{i,k}^w$
  $U=\{u_0,\cdots,u_r\}$
- Problems to solve :
  - Computation of the position of points of interpolation along the trajectory curve (especially the vectorial function $B(v)$)
  - Computation of the final surface

NURBS surfaces

- The surface has the following form :

$$\widetilde{S}^{w}(u,v)=\sum_{i=0}^{n}\sum_{k=0}^{K}N_i^p(u)\,N_k^q(v)\,P_{i,k}^{w}$$

- We have to determine :

  - the values of the parameter v for which curves C_k interpolate $\widetilde{S}^{w}(u,v)$. We shall call these values $\bar{V}=\{\bar{v}_0,\cdots,\bar{v}_K\}$
  - the nodal sequence $V=\{v_0,\cdots,v_s\}$
  - the control points $P_{i,k}^{w}$ …

- Computation of values $\bar{v}_i$ for which we interpolate, and deduction of the nodal sequence V
  - The number of nodes of V is $s+1$
  - The number of interpolated positions is $K+1$ (min. given by the user)
  - The degree of the trajectory is q (imposed)

    We want, if possible, to keep the nodal sequence of the trajectory (same domain for v).

  If $s = K + q + 1$ everything is OK.
  If $s \leq K + q$ inserting nodes in the nodal sequence is needed
  $\rightarrow K + q - s + 1$ nodal insertions
  If $s > K + q + 1$, add interpolated positions
  in such a way that $s = K + q + 1$

- Case where we must make nodal insertions
  - We aim at an approximately regular repartition
  - The exact location of these insertions does not matter
  - For instance, subdividing the longest nodal interval in two equal parts (and repeat this $K+q-s+1$ times) is suitable.

$$V=\{0,0,0,1,2,4,8,10,10,10\}$$

$$m=3$$

$$V'=\{0,0,0,1,2,4,6,8,10,10,10\}$$

$$V'=\{0,0,0,1,2,3,4,6,8,10,10,10\}$$

$$V'=\{0,0,0,1,2,3,4,5,6,8,10,10,10\}$$

  - The position of the new control points of the trajectory $T(v)$ is not needed, because its nodal sequence is not modified !!!

- Computation of the values of the parameter v for the interpolation, $\bar{V}=\{\bar{v}_k\}\ ,\ k=0,\cdots,K$
  - The repartition depends on the nodal sequence v_k
  - For a node have a multiplicity of q, the curve interpolates one of the CPs, therefore this value must be part of the $\bar{v}_k$

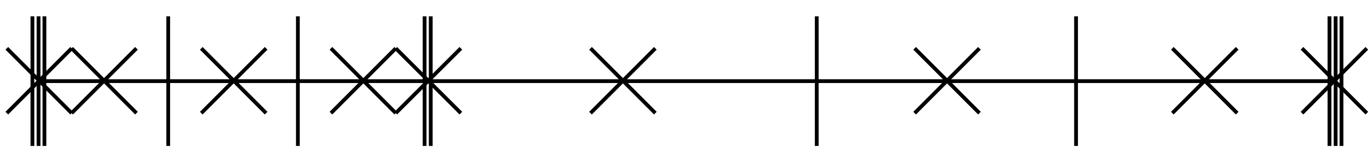
  A sliding average on q nodes (where q is the degree) is a good solution :

$$\bar{v}_k=\frac{1}{q}\sum_{i=1}^{q} v_{k+i}\quad ,\quad k=1,\cdots,K-1\quad ,\quad \bar{v}_0=v_0\quad \bar{v}_K=v_{s'}$$

  Example with $q=2$: 9 control points and as many interpolation points

$$V'=\{0,0,0,1,2,3,3,6,8,10,10,10\}$$

$$\bar{V}=\{0,\frac{1}{2},\frac{3}{2},\frac{5}{2},3,\frac{9}{2},7,9,10\}$$

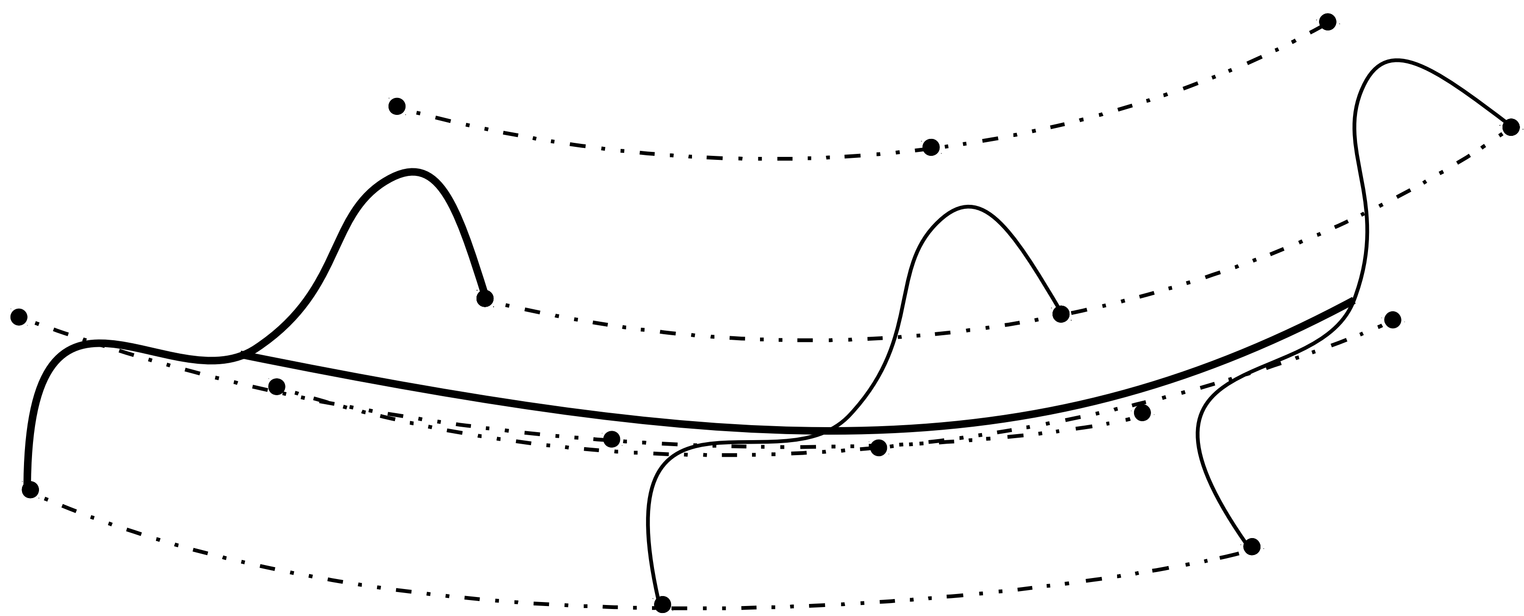


NURBS surfaces

- Two things remain to be done

  1 : computation of positions of the instances of the profile curve, i.e. computations of positions of CPs of each instance C_k

  2 : computation of the position of control points of curves passing through the control points of the instances



NURBS surfaces

- Computation of the instances of the section (profile)

$$S(u,v)=T(v)+M(v)C(u)$$

$$C^w(u)=\sum_{i=0}^{n} N_i^p(u)\,C_i^w$$

$$\downarrow$$

$$C_k^w(u)=\sum_{i=0}^{n} N_i^p(u)\,C_{i,k}^w$$

$$C_{i,k}^w=M^w(\bar{v}_k)\cdot C_i^w$$

$$\{O,X,Y,Z\}$$

$$\{o(\bar{v}_k),x(\bar{v}_k),y(\bar{v}_k),z(\bar{v}_k)\}$$

$o(\bar{v}_k)=T(\bar{v}_k)$ $\quad$ $B(\bar{v}_k)$ given

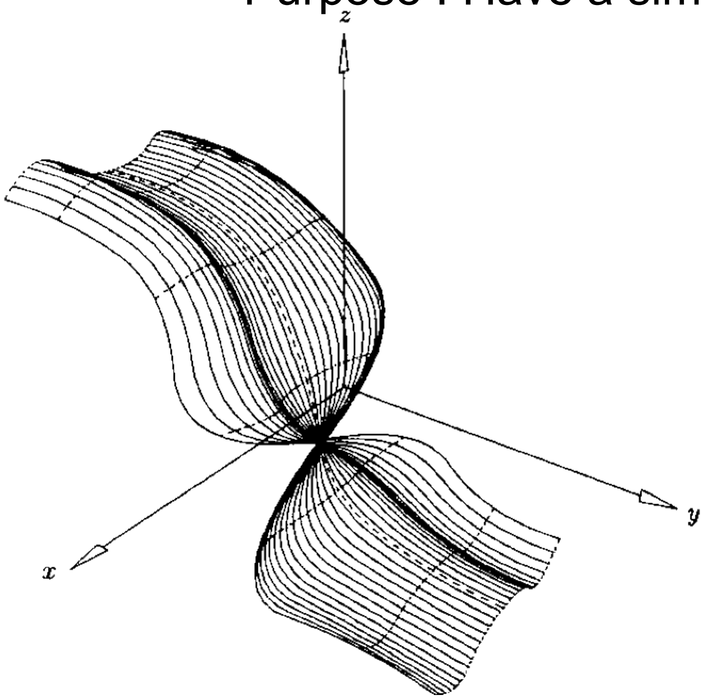
$$x(\bar{v}_k)=\frac{T'(\bar{v}_k)}{|T'(\bar{v}_k)|} \qquad z(\bar{v}_k)=\frac{B(\bar{v}_k)}{|B(\bar{v}_k)|}$$

$$y(\bar{v}_k)=z(\bar{v}_k)\times x(\bar{v}_k)$$

$$C_i^w=\begin{pmatrix} x_i w_i \\ y_i w_i \\ z_i w_i \\ w_i \end{pmatrix} \qquad C_{i,k}^w=\begin{pmatrix} x_{i,k} w_{i,k} \\ y_{i,k} w_{i,k} \\ z_{i,k} w_{i,k} \\ w_{i,k} \end{pmatrix} \qquad M^w(\bar{v}_k)=\begin{pmatrix} | & | & | & | \\ x(\bar{v}_k) & y(\bar{v}_k) & z(\bar{v}_k) & o(\bar{v}_k) \\ | & | & | & | \\ 0 & 0 & 0 & 1 \end{pmatrix}\cdot w(\bar{v}_k)$$

NURBS surfaces

- Computation of $B(\bar{v}_k)$
  - Purpose : Have a similar orientation as the Frenet frame…



Three problems if $B(\bar{v}_k)$ is related to Frenet frame:

1- $B(\bar{v}_k)$ is not defined at places where $T(v)$ is a straight line (locally) or at inflexion points

2- $B(\bar{v}_k)$ abruptly changes its orientation before and after an inflexion point

3- For three-dimensional trajectories, vectors obtained by the use of $B(\bar{v}_k)$ can turn arbitrarily fast around the curve

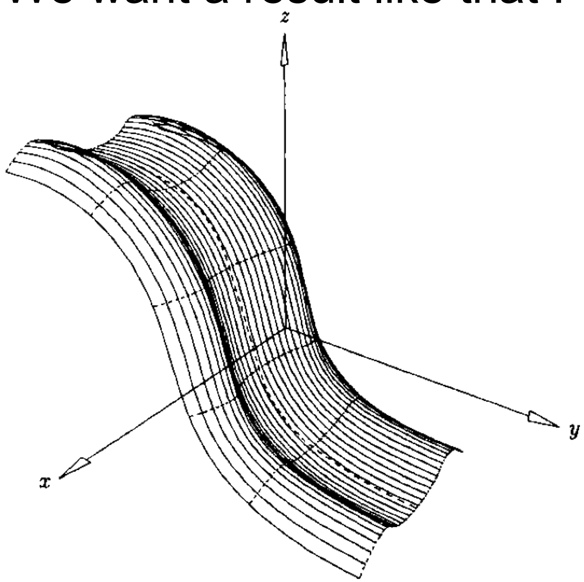
… by avoiding problems raised by the following definition (Frenet)

$$B(\bar{v}_k)=\frac{T'(\bar{v}_k)\times T''(\bar{v}_k)}{\left|T'(\bar{v}_k)\times T''(\bar{v}_k)\right|}$$

Computer Aided Design

NURBS surfaces

- Computation of $B(\bar{v}_k)$
  - We want a result like that :



Attention: avoid having $T_k \,//\, B_{k-1}$
Therefore K must such that the curve "turns" less than 90° between $\bar{v}_{k-1}$ and $\bar{v}_k$)

- Method of the normal projection*
  - We are going to compute explicitly the values of $B(\bar{v}_k)$ for each parameter $\bar{v}_k$
  - Let $\bar{v}_0 \cdots \bar{v}_k$ the increasing sequence of the parameter v . We compute $B_k = B(\bar{v}_k)$ by the following way :

$$T_k = \frac{T'(\bar{v}_k)}{\left|T'(\bar{v}_k)\right|}$$

$$b_k = B_{k-1} - (B_{k-1} \cdot T_k)\, T_k$$

$$B_k = \frac{b_k}{\left|b_k\right|}$$

B_0 is imposed by the user

* P. Stiltanen and C. Woodward, Normal orientation methods for 3D offset curves, sweep surfaces and skinning, *Proc. Eurographics'92*, **11**(3), pp. C 449 – C 457, 1992.

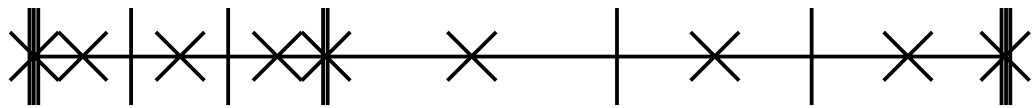
- Case of periodic curves
  - In general, $B_K \neq B_0$
  - We can do the computation in two opposite directions:

$$\hat{B}_0 \rightarrow \hat{B}_K$$
$$\overline{B}_K \rightarrow \overline{B}_0$$

  - Then we set

$$B_k = \frac{\overline{B}_k + \hat{B}_k}{2} \quad , \quad k = 1, \cdots, K-1$$

- Global interpolation on a curve
  - We have interpolation points $C^w_{i,k}$
  - We have a nodal sequence : $V'=\{v'_i\}\ ,\ i=0,\cdots,s'$
  - We have the values of v for the interpolation : $\bar{V}=\{\bar{v}_j\}\ ,\ j=0,\cdots,K$



  - Now, we need to compute the expression of curves passing through the CP of the instances of the profile:

$$T^w_i(v)=\sum_{k=0}^{K} N^p_k(v)\,P^w_{i,k}\ \text{ such that }\ T^w_i(\bar{v}_k)=C^w_{i,k}\quad \forall k=0,\cdots,K$$

  - The control points of these curves are the control points of the surface that is sought. Why ?

    - because the expression of the surface is separable in u and v. See how we were able to compute the control points of an isoparametric on a B-Spline surface – see e.g. slide 36 of chapter 5

NURBS surfaces

$$T_i^w(v)=\sum_{k=0}^{K} N_k^p(v)\,P_{i,k}^w \quad \text{such that} \quad T_i^w(\bar{v}_k)=C_{i,k}^w \quad \forall\, k=0,\cdots,K$$

- We obtain a linear system

$$\begin{pmatrix} N_0^p(\bar{v}_0) & \cdots & N_K^p(\bar{v}_0) \\ \vdots & \ddots & \vdots \\ N_0^p(\bar{v}_K) & \cdots & N_K^p(\bar{v}_K) \end{pmatrix} \begin{pmatrix} P_{i,0}^w \\ \vdots \\ P_{i,K}^w \end{pmatrix} = \begin{pmatrix} C_{i,0}^w \\ \vdots \\ C_{i,K}^w \end{pmatrix}$$

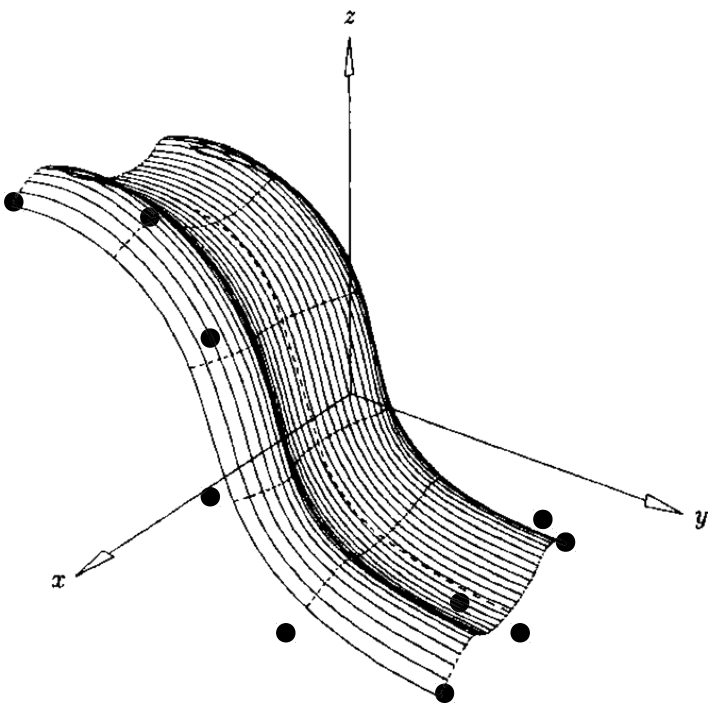
$$(A)\cdot(P_i^w)=(C_i^w)$$

- The matrix A only depends on the nodal sequence $V'=\{v_i'\}$ and the values $\bar{V}=\{\bar{v}_j\}$
- For each series of CP, this system is to be solved 4 times (once for each coordinate x,y,z and w) , $4(n+1)$ times in total.
  - Best choice : LU decomposition (once) + back substitution ($4(n+1)$ times with each different right hand side)

NURBS surfaces

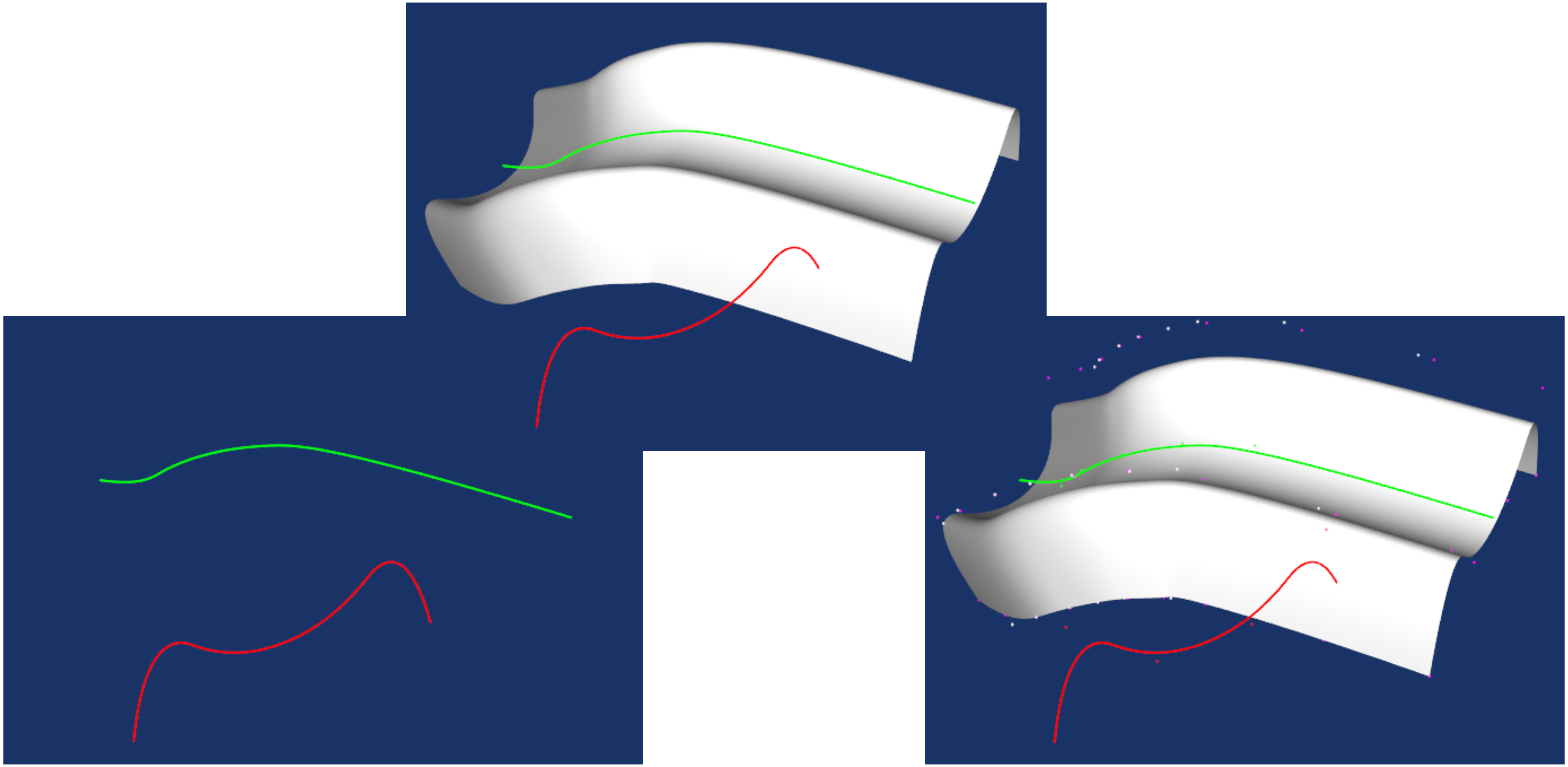
- Final surface

$$\tilde{S}^w(u,v)=\sum_{i=0}^{n}\sum_{k=0}^{K} N_i^p(u)\,N_k^q(v)\,P_{i,k}^w \qquad V'=\{v_i'\}\;,\quad i=0,\cdots,s'$$



NURBS surfaces

- Extrusion of the red curve along the green one



- Skinning
  - Consists in generation of a « skin » supported by a series of curves $C_k(u) \quad , \quad k=0 \cdots K$
    - The curves C_k are interpolated
    - The $C_k(u)$ so are isoparametrics of the skin $P(u,v)$ and are NURBS curves :

$$C_k^w(u) = \sum_{i=0}^{n} N_i^p(u) C_{i,k}^w \qquad U = \{u_0, \cdots, u_r\}$$

    - We assume they are compatible (same nodal sequence, same degree, same number of CPs)
    - If it is not the case, use algorithms seen before to make them compatible (nodal insertion and degree elevation)

- Skinning
  - The technique seen for building the profiled surface may be used
  - However, the trajectory curve is not known
    - We need to build a nodal sequence V, choose an order q and the values $\bar{V}=\{\bar{v}_j\}$ for which we interpolate the curves C_k.
    - The number of curves C_k is imposed : it is $K+1$.

$$V=\{v_i\} \quad , \quad i=0,\cdots,s$$

$$\bar{V}=\{\bar{v}_j\} \quad , \quad j=0,\cdots,K$$

    - The explicit expression of the trajectory curve is, in fact, not needed !

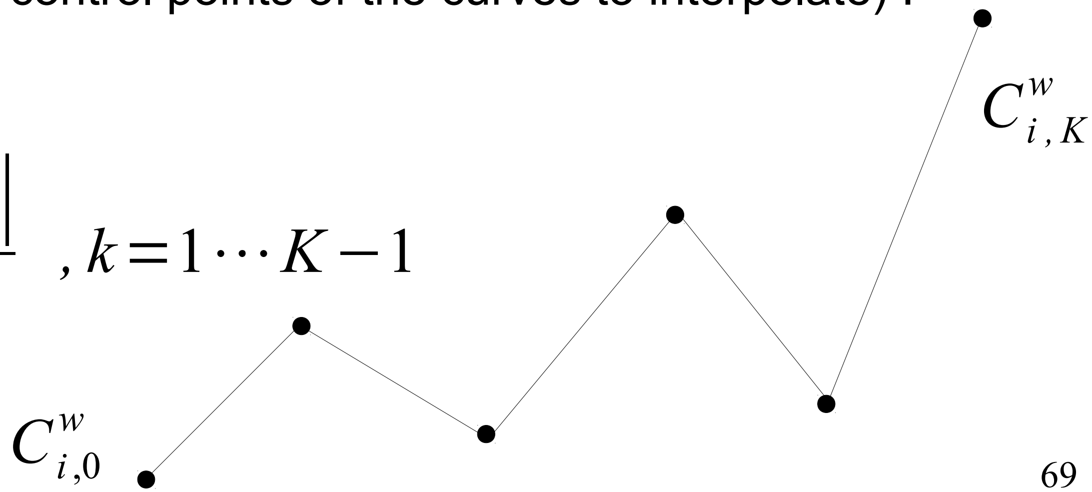
NURBS surfaces

- Skinning
  - Determination of the degree q
    - Arbitrary (user choice) but must be below $K+1$
  - Determination of the values $\bar{V} = \{\bar{v}_j\}$
    - $K+1$ (nb of curves to interpolate) is fixed.
    - It is done by computing an approximation of the average arc length (averaged over the n control points of the curves to interpolate) :

$$\bar{v}_0 = 0\,;\, \bar{v}_K = 1\,;$$

$$\bar{v}_k = \bar{v}_{k-1} + \sum_{i=0}^{n} \frac{\left| C^w_{i,k} - C^w_{i,k-1} \right|}{d_i} \quad , k = 1 \cdots K-1$$

$$d_i = \sum_{k=1}^{K} \left| C^w_{i,k} - C^w_{i,k-1} \right|$$

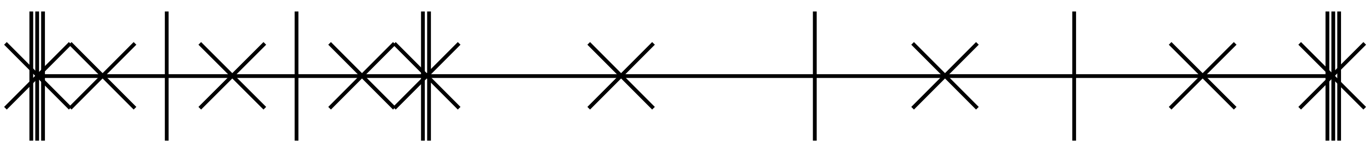


Computer Aided Design

NURBS surfaces

- Skinning
  - Determination of the nodal sequence
    - The same technique of sliding average previously used...

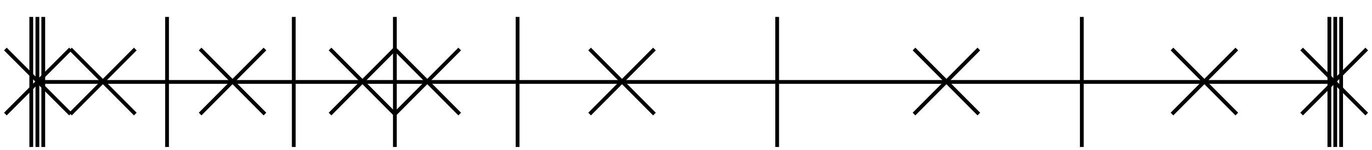
$$\bar{v}_k = \frac{1}{q}\sum_{i=1}^{q} v_{k+i} \quad , \quad k = 1, \cdots, K-1 \quad , \quad \bar{v}_0 = v_0 \quad \bar{v}_K = v_s$$



, but it is “reversed” to get v_k in terms of $\bar{v}_k$

$$v_{k+q} = \frac{1}{q}\sum_{i=k}^{k+q-1} \bar{v}_i \quad ; \quad k = 1, \cdots, K-q \quad ; \quad v_0 = \cdots = v_q = \bar{v}_0 \quad ; \quad v_{K+1} = \cdots = v_{K+q+1} = \bar{v}_K$$

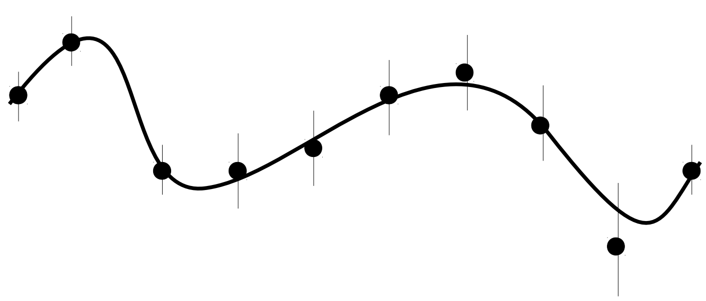
- There can't be multiple nodes except at boundaries ...

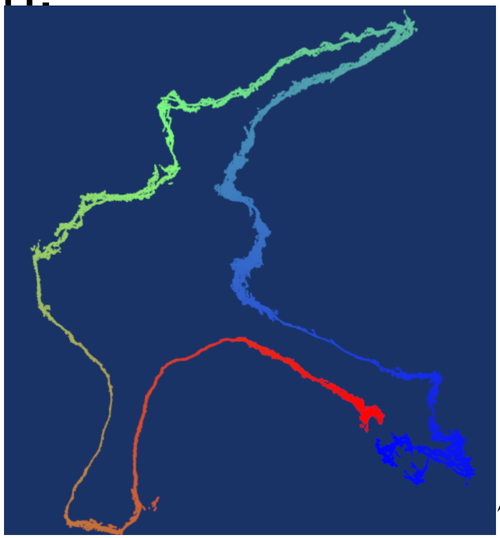


- Skinning
  - We now have a nodal sequence, values of v for which the curves C_k are interpolated, and their control points.
  - The remaining (determination of the coordinates of the CPs of the surface) is identical to the previous case of an extrusion along a defined curve.

Least Squares

- Least squares
  - Suppose we have a huge number of 3D samples (from laser sampler), for an object.
    We want to reconstruct a shape, for which the description shall be both light and accurate.
    However, there are sampling errors, let's suppose those errors follow a normal distribution.





Least Squares

- 1D case (curves)
  - Suppose we have N samples :

$$e_k = \begin{pmatrix} x_k \\ y_k \\ z_k \end{pmatrix}, k = 0 \ldots N-1, \text{ with a standard deviation } \sigma_k$$

One wants to approximate these with a curve that has n parameters , with $n \ll N$:

$$C(u) = \sum_{i=0}^{n-1} P_i \cdot \varphi_i(u)$$

Least Squares

- The discrepancy $\left\|C(u_k)-e_k\right\|$ between the curve and the samples is weighted by the inverse of the normal deviation
  - if the latter is small, then the curve shall be closer to the sample
- We get : $err_k=\left(\frac{1}{\sigma_k}\left\|C(u_k)-e_k\right\|\right)^2=\left(\frac{1}{\sigma_k}\left\|\sum_{i=0}^{n-1} P_i\cdot\varphi_i(u_k)-e_k\right\|\right)^2$
- We do not have the u_k's yet. Those must be computed, for instance considering that the samples are equidistant in the parametric space, or this way :

$u_k-u_{k-1}=\left\|e_k-e_{k-1}\right\|$, $k=1\ldots N-1$ and $u_0=0.$

  - Anyway; this sequence should be built **before** minimizing the error so that the problem remains linear.

Least Squares

- One wishes to minimize the total error over all samples :

$$\chi^2 = \sum_{k=0}^{N-1} \frac{1}{\sigma_k^2} \left\| \sum_{i=0}^{n-1} P_i \cdot \varphi_i(u_k) - e_k \right\|^2$$

with respect to the control points $P_i = \begin{pmatrix} px_i \\ py_i \\ pz_i \end{pmatrix}$, $i = 0 \cdots n-1$

- One can express the total error along each axis :

$$\chi^2 = \sum_{k=0}^{N-1} \frac{1}{\sigma_k^2} \left(\sum_{i=0}^{n} px_i \cdot \varphi_i(u_k) - x_k \right)^2 + \text{ terms in } y \text{ and } z$$

Least Squares

- One can put it in a matrix form :

$$\chi^2 = \left(\mathbf{J}\,\mathbf{P}_x - \mathbf{E}_x\right)^{\mathrm{T}} \mathbf{W} \left(\mathbf{J}\,\mathbf{P}_x - \mathbf{E}_x\right) + \left(\mathbf{J}\,\mathbf{P}_y - \mathbf{E}_y\right)^{\mathrm{T}} \mathbf{W} \left(\mathbf{J}\,\mathbf{P}_y - \mathbf{E}_y\right) + \left(\mathbf{J}\,\mathbf{P}_z - \mathbf{E}_z\right)^{\mathrm{T}} \mathbf{W} \left(\mathbf{J}\,\mathbf{P}_z - \mathbf{E}_z\right)$$

with

$$\mathbf{J} = \begin{pmatrix} \varphi_0(u_0) & \cdots & \varphi_{n-1}(u_0) \\ \vdots & & \vdots \\ \varphi_0(u_{N-1}) & \cdots & \varphi_{n-1}(u_{N-1}) \end{pmatrix} \qquad \mathbf{P}_x = \begin{pmatrix} px_0 \\ \vdots \\ px_{n-1} \end{pmatrix}$$

$$\mathbf{W} = \begin{pmatrix} 1/\sigma_0^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1/\sigma_{N-1}^2 \end{pmatrix} = \begin{pmatrix} w_0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & w_{N-1} \end{pmatrix} \qquad \mathbf{E}_x = \begin{pmatrix} x_0 \\ \vdots \\ x_{N-1} \end{pmatrix}$$

- Now one wants to minimize the error
  - thus the differential of the error with respect to each P_i should vanish

e.g. $$\frac{\partial \chi^2}{\partial xp_i} = \frac{\partial (\mathbf{J}\,\mathbf{P}_x - \mathbf{E}_x)^{\mathrm{T}} \mathbf{W} (\mathbf{J}\,\mathbf{P}_x - \mathbf{E}_x)}{\partial xp_i}$$

$$= \frac{\partial (\mathbf{J}\,\mathbf{P}_x - \mathbf{E}_x)^{\mathrm{T}}}{\partial xp_i} \mathbf{W} (\mathbf{J}\,\mathbf{P}_x - \mathbf{E}_x) + (\mathbf{J}\,\mathbf{P}_x - \mathbf{E}_x)^{\mathrm{T}} \mathbf{W} \frac{\partial (\mathbf{J}\,\mathbf{P}_x - \mathbf{E}_x)}{\partial xp_i}$$

$$= \frac{\partial \mathbf{P}_x^{\mathrm{T}}}{\partial xp_i} \mathbf{J}^{\mathrm{T}} \mathbf{W} (\mathbf{J}\,\mathbf{P}_x - \mathbf{E}_x) + (\mathbf{J}\,\mathbf{P}_x - \mathbf{E}_x)^{\mathrm{T}} \mathbf{W}\,\mathbf{J} \frac{\partial \mathbf{P}_x}{\partial xp_i}$$

$$(0, \cdots, 1, \cdots, 0)$$

$$\begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix}$$

$$= 2\left[\mathbf{J}^{\mathrm{T}} \mathbf{W}\,\mathbf{J}\,\mathbf{P}_x - \mathbf{J}^{\mathrm{T}} \mathbf{W}\,\mathbf{E}_x\right]_{i^{th}\ \text{line}} = 0$$

Least Squares

- Overall, this should be written for each variable, thus :

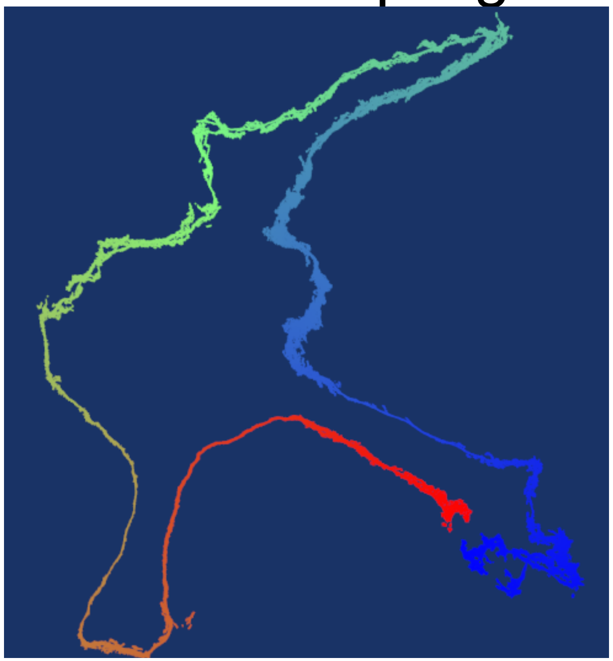
$$\nabla_P \chi^2 = \begin{pmatrix} 2\,\mathbf{J}^{\mathrm{T}}\mathbf{W}\,\mathbf{J}\,\mathbf{P}_x - 2\,\mathbf{J}^{\mathrm{T}}\mathbf{W}\,\mathbf{E}_x \\ 2\,\mathbf{J}^{\mathrm{T}}\mathbf{W}\,\mathbf{J}\,\mathbf{P}_y - 2\,\mathbf{J}^{\mathrm{T}}\mathbf{W}\,\mathbf{E}_y \\ 2\,\mathbf{J}^{\mathrm{T}}\mathbf{W}\,\mathbf{J}\,\mathbf{P}_z - 2\,\mathbf{J}^{\mathrm{T}}\mathbf{W}\,\mathbf{E}_z \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}$$

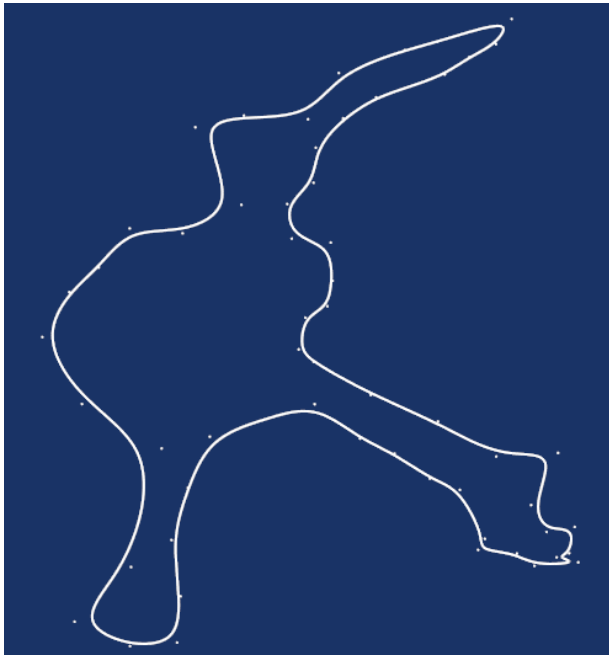
$$\begin{cases} \mathbf{P}_x = \left(\mathbf{J}^{\mathrm{T}}\mathbf{W}\,\mathbf{J}\right)^{-1}\mathbf{J}^{\mathrm{T}}\mathbf{W}\,\mathbf{E}_x \\ \mathbf{P}_y = \left(\mathbf{J}^{\mathrm{T}}\mathbf{W}\,\mathbf{J}\right)^{-1}\mathbf{J}^{\mathrm{T}}\mathbf{W}\,\mathbf{E}_y \\ \mathbf{P}_z = \left(\mathbf{J}^{\mathrm{T}}\mathbf{W}\,\mathbf{J}\right)^{-1}\mathbf{J}^{\mathrm{T}}\mathbf{W}\,\mathbf{E}_z \end{cases}$$

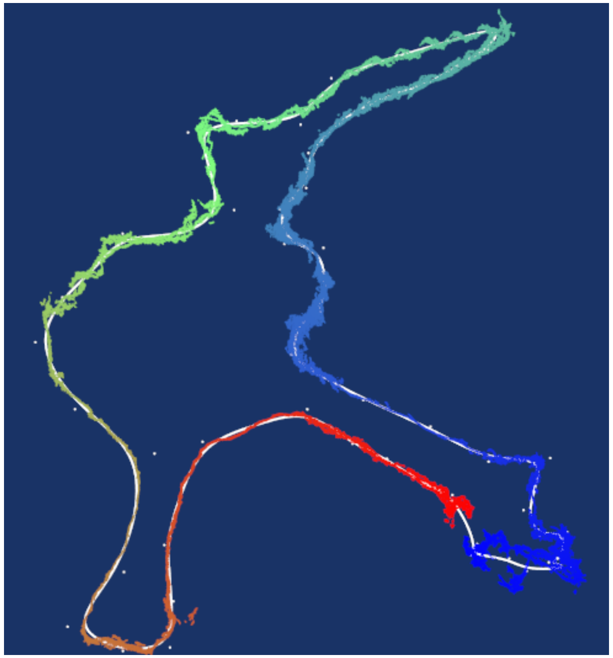
- This system can be solved by an LU decomposition of $\mathbf{J}^{\mathrm{T}}\mathbf{W}\,\mathbf{J}$.

Least Squares

- Sampling of a trunk, slice as a periodic B-Spline







NURBS surfaces

- NURBS = open modelling system
- The following geometries cannot be represented exactly using NURBS :
  - Profiles extruded along any trajectory (except straight lines and circles)
  - Curve at a given distance of another curve
  - Intersection of two NURBS surfaces
  - Projection of a NURBS curve on a surface
  - Many other cases … however, by increasing the number of control points and/or the degree, convergence toward the exact geometry is usually **very fast**.