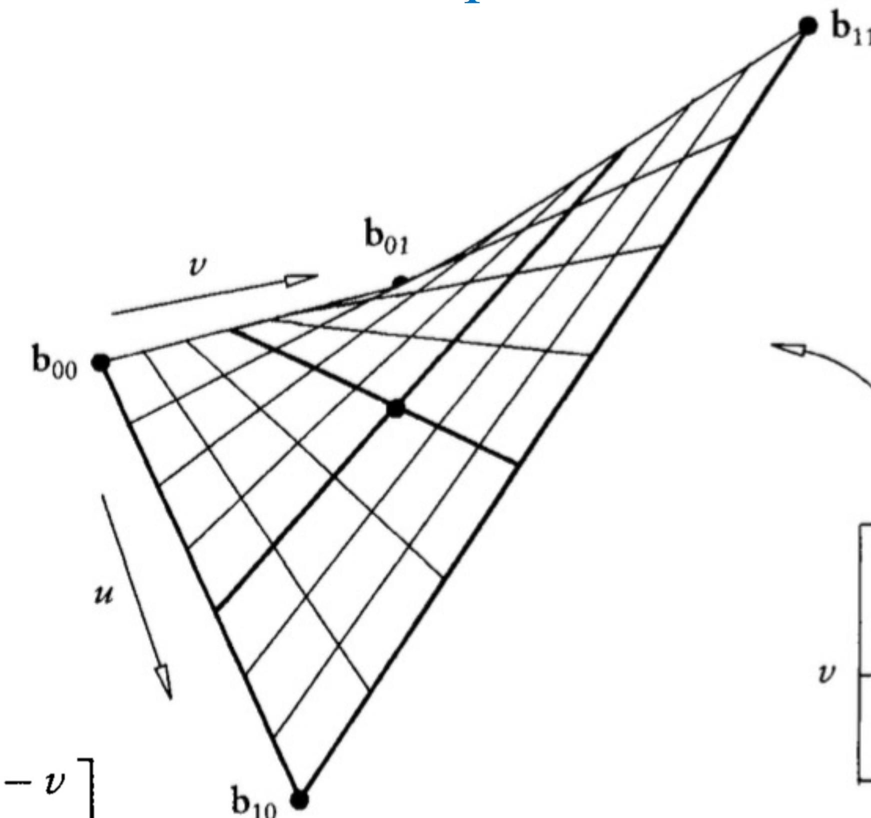
Bézier Surfaces and NURBS Surfaces

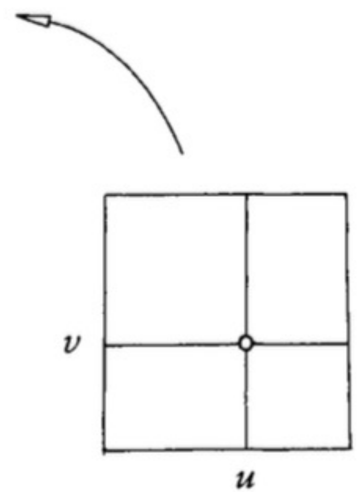
Laurent Busé

Université Côté d'Azur, Inria
Email : laurent.buse@inria.fr

Bilinear interpolation

As linear interpolation fits the « simplest » curve between two points, bilinear interpolation fits the « simplest » surface between four points:



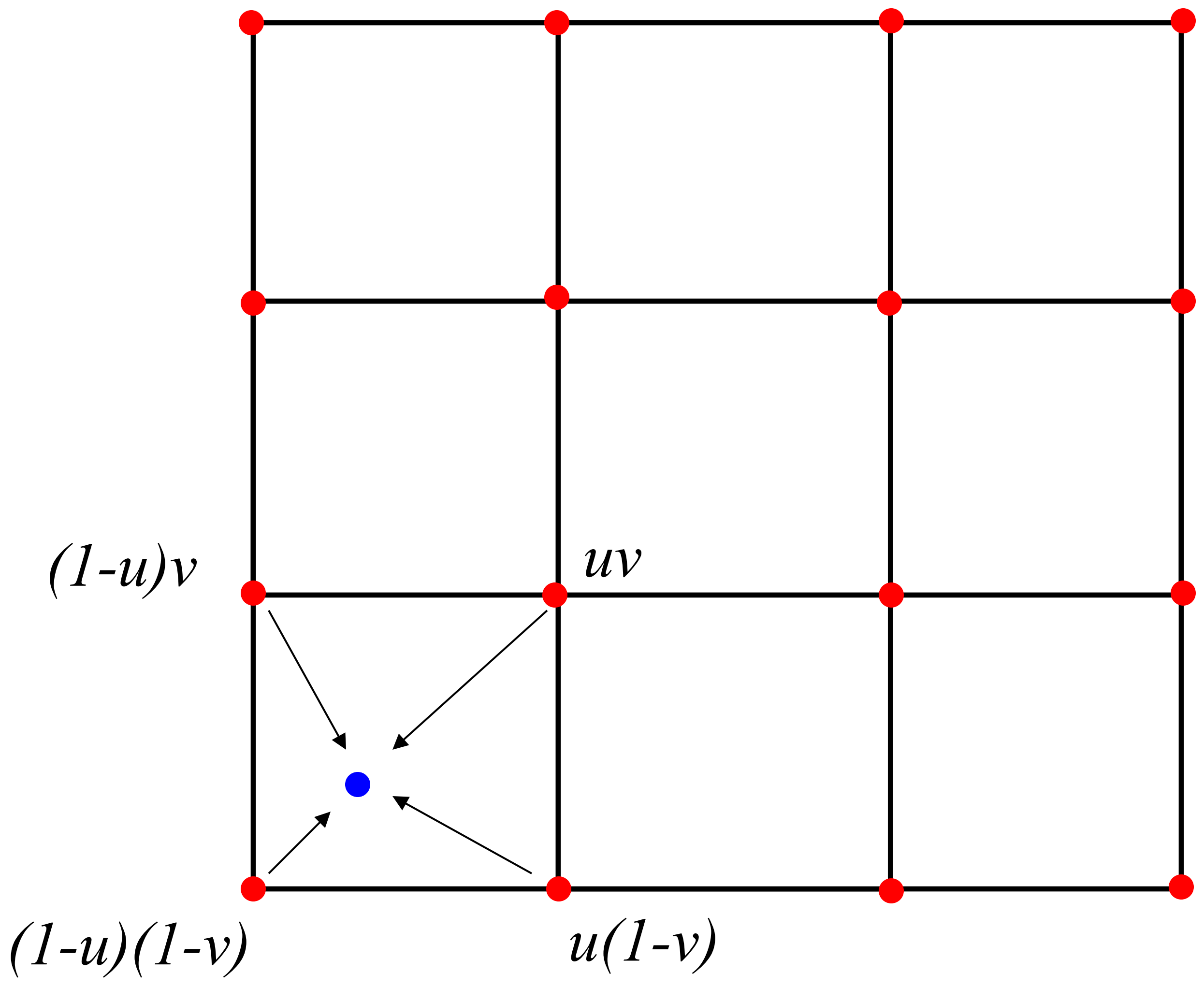


$$\mathbf{x}(u,v) = \sum_{i=0}^{1}\sum_{j=0}^{1} \mathbf{b}_{i,j} B_i^1(u) B_j^1(v)$$

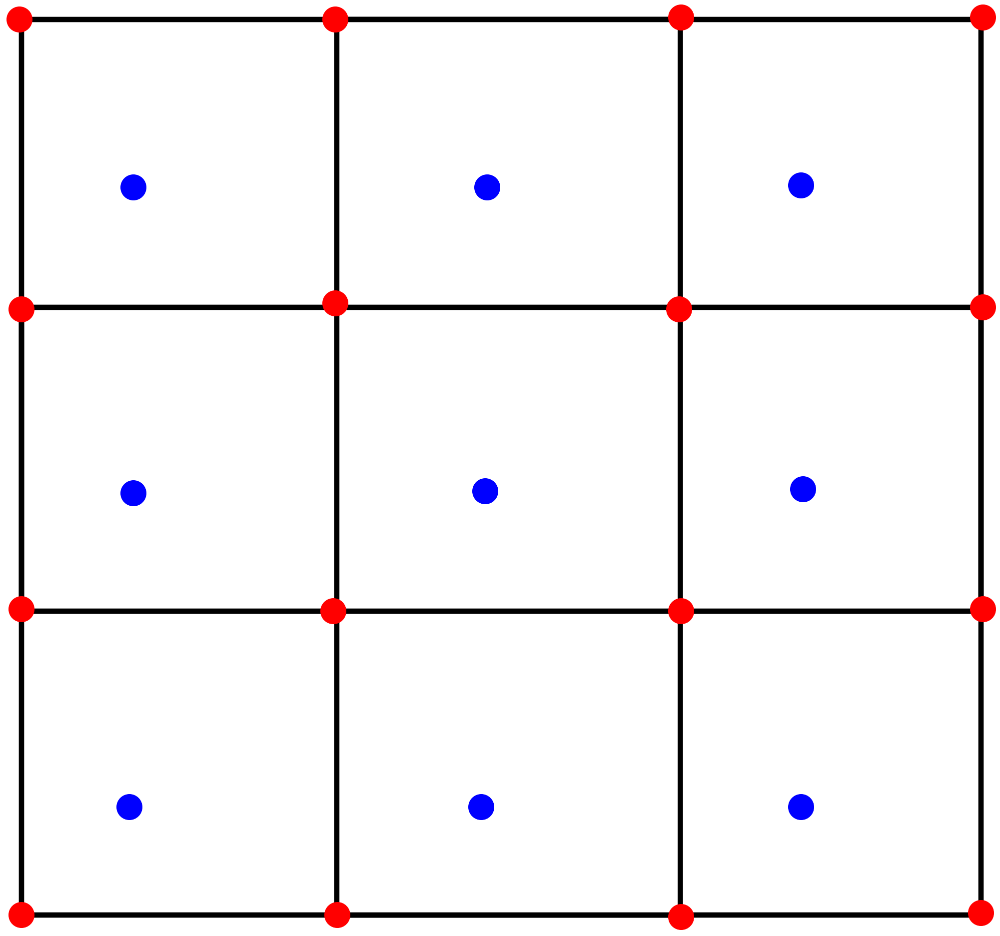
Matrix formulation:

$$\mathbf{x}(u,v) = \begin{bmatrix} 1-u & u \end{bmatrix} \begin{bmatrix} \mathbf{b}_{00} & \mathbf{b}_{01} \\ \mathbf{b}_{10} & \mathbf{b}_{11} \end{bmatrix} \begin{bmatrix} 1-v \\ v \end{bmatrix}$$

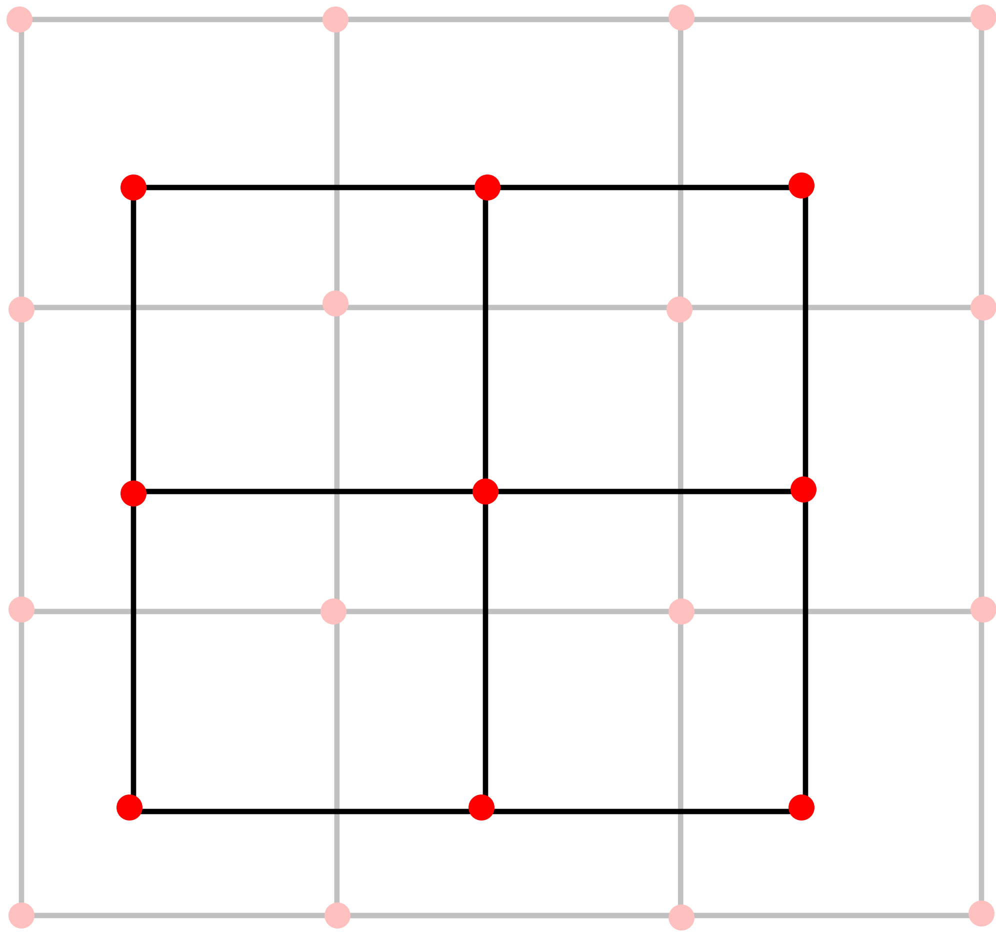
Repeated bilinear interpolation



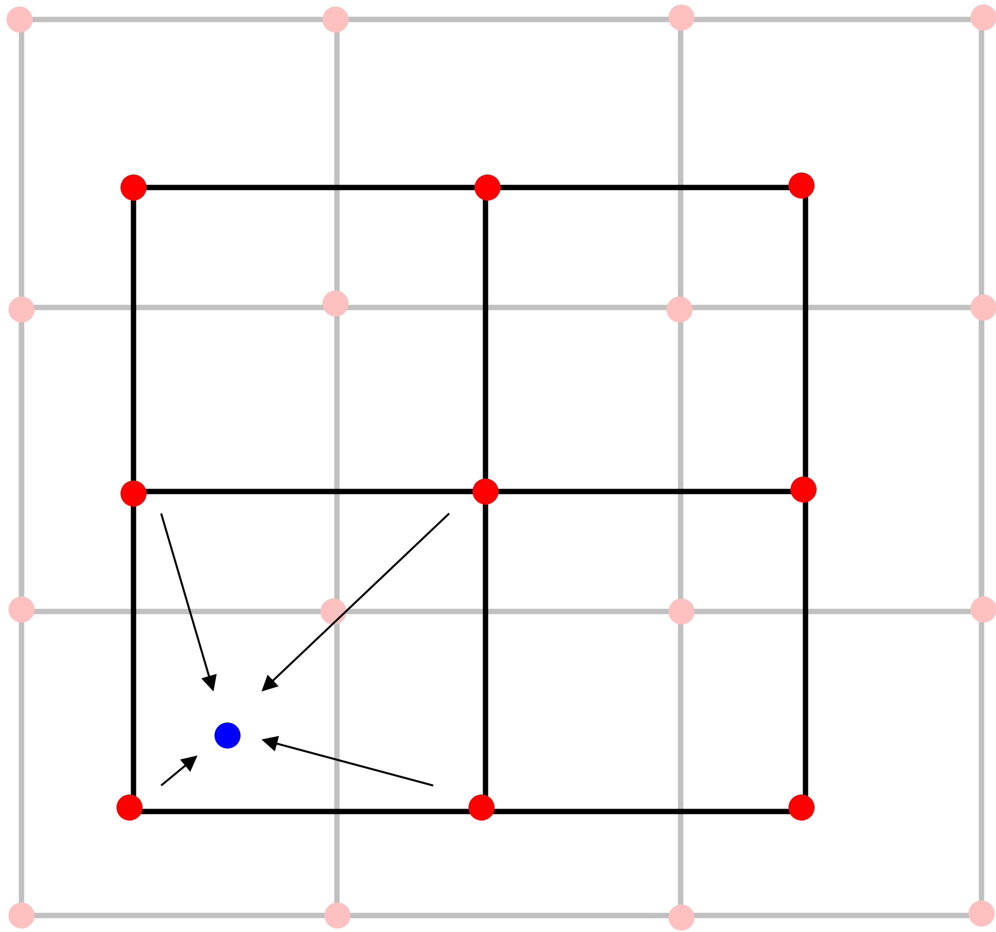
Repeated bilinear interpolation



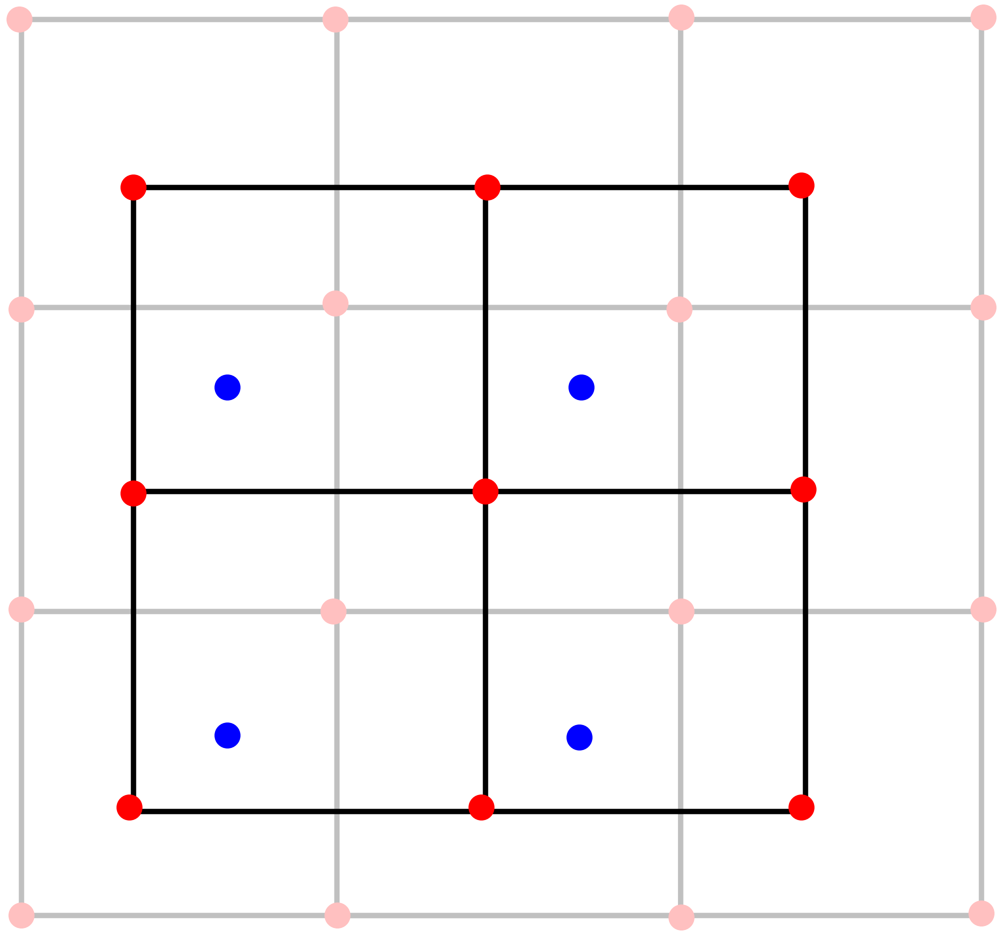
Repeated bilinear interpolation



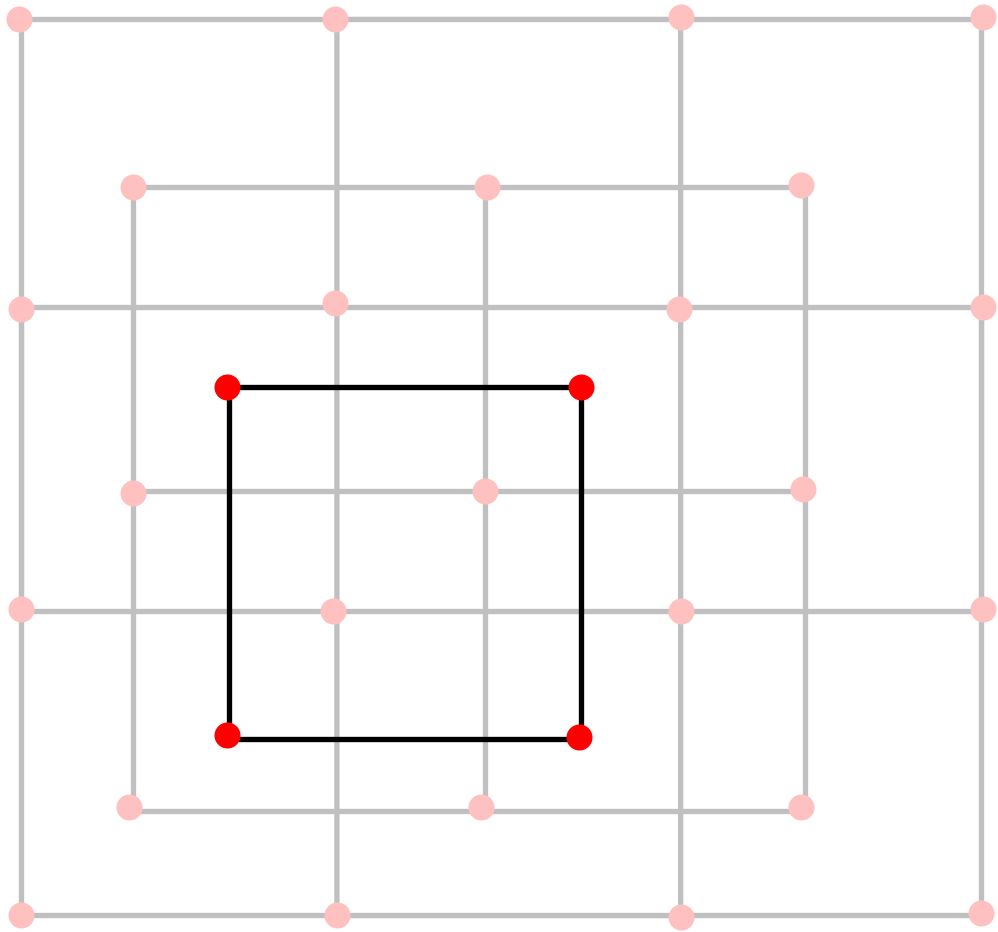
Repeated bilinear interpolation



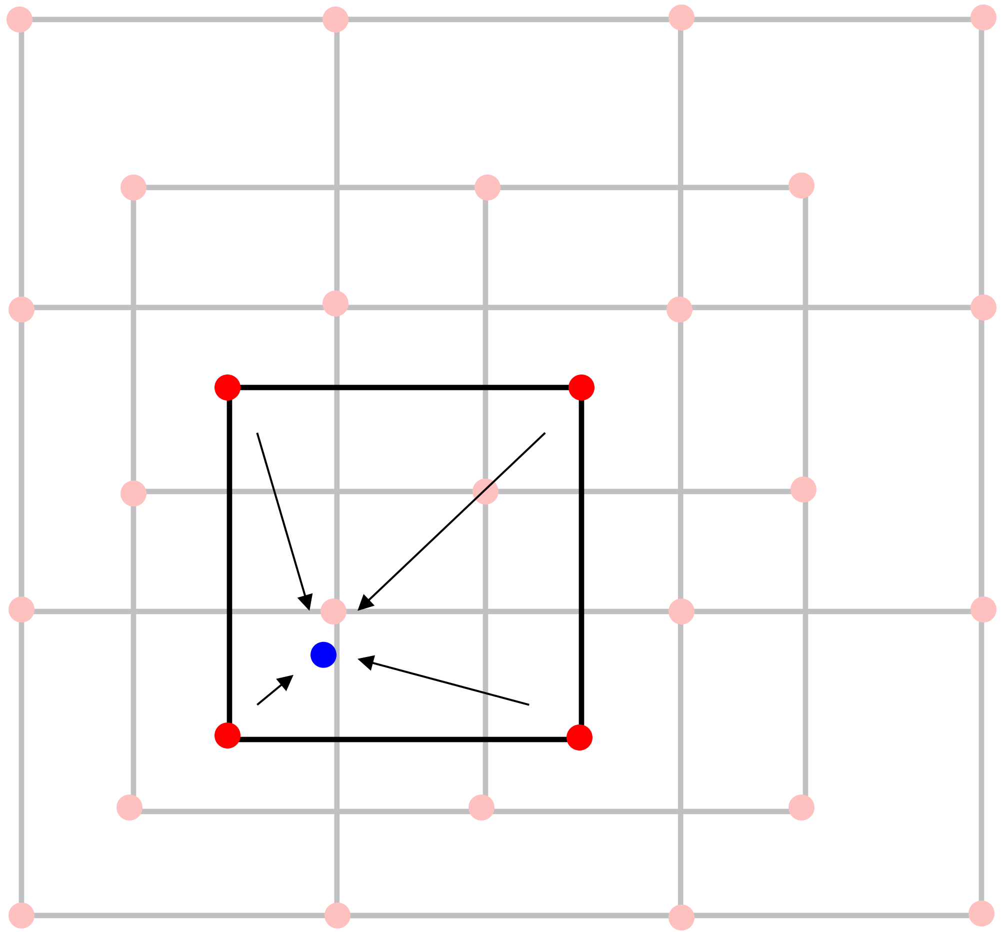
Repeated bilinear interpolation



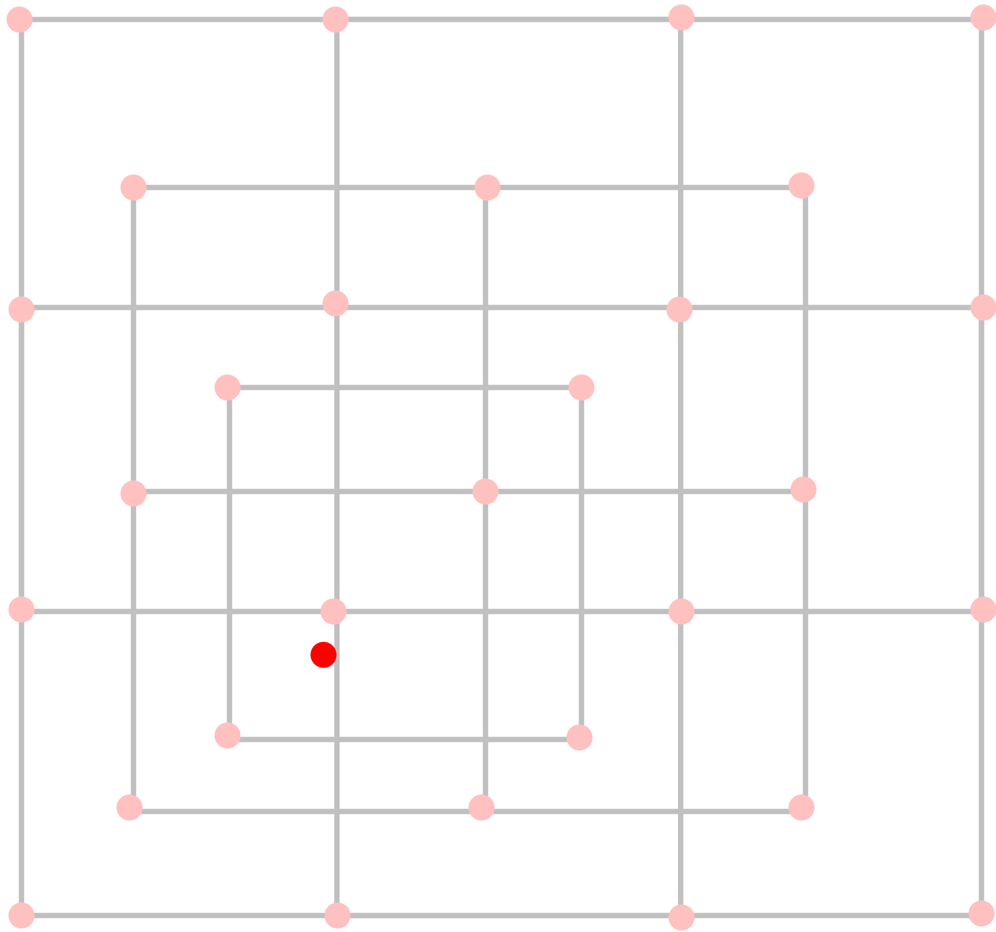
Repeated bilinear interpolation



Repeated bilinear interpolation



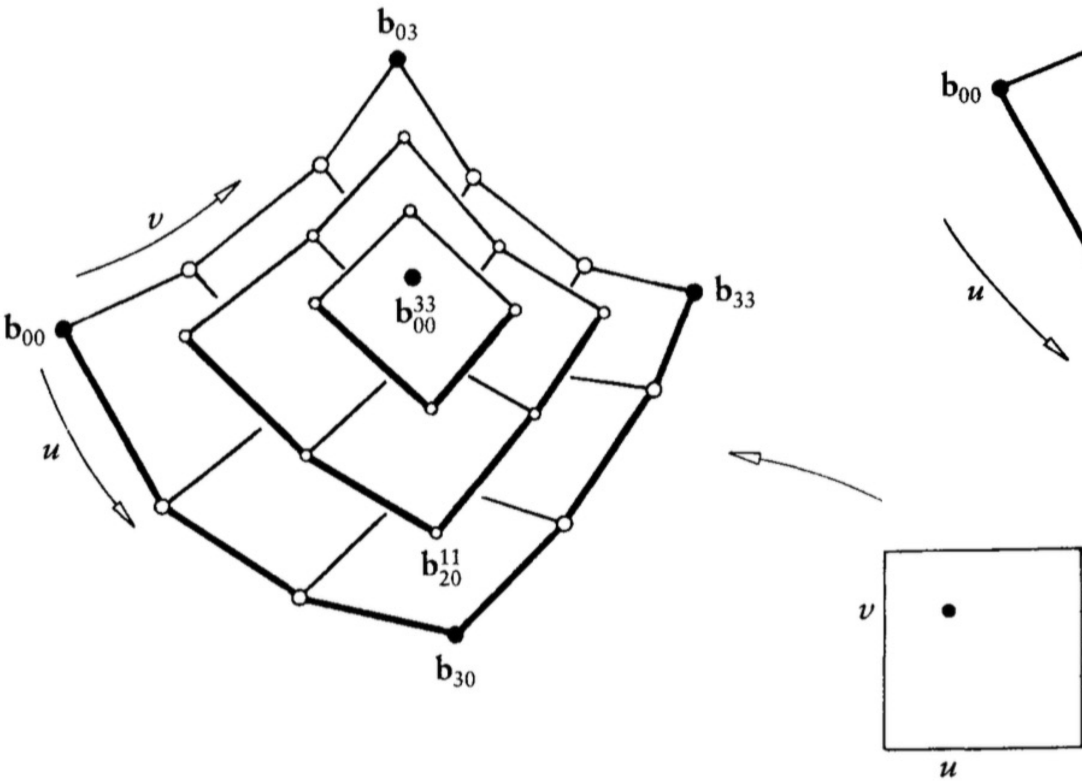
Repeated bilinear interpolation

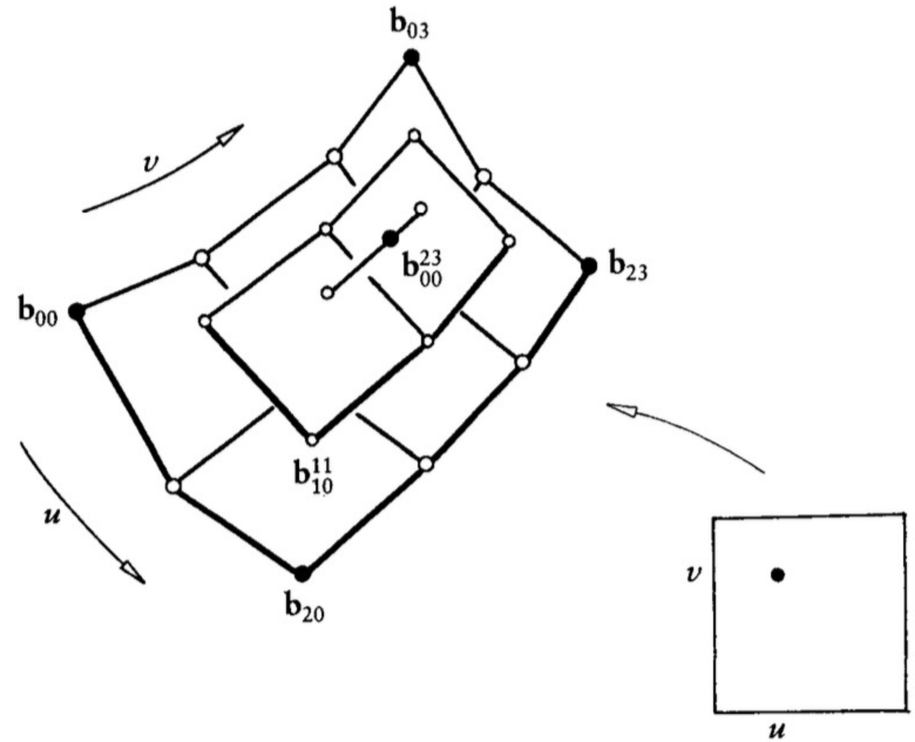


Direct de Casteljau algorithm

- Surfaces are obtained from repeated bilinear interpolation

Bicubic patch





The tensor product approach

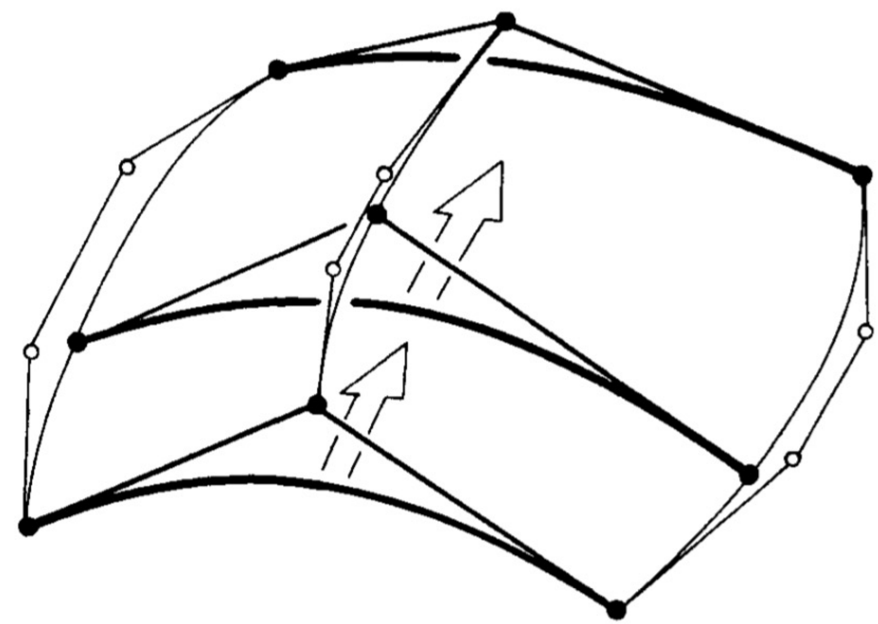
A surface is the locus of a curve that is moving through space

- Initial Bézier curve of degree m :

$$\mathbf{b}^m(u) = \sum_{i=0}^{m} \mathbf{b}_i B_i^m(u).$$

- Each control point draws a Bézier curve of degree n

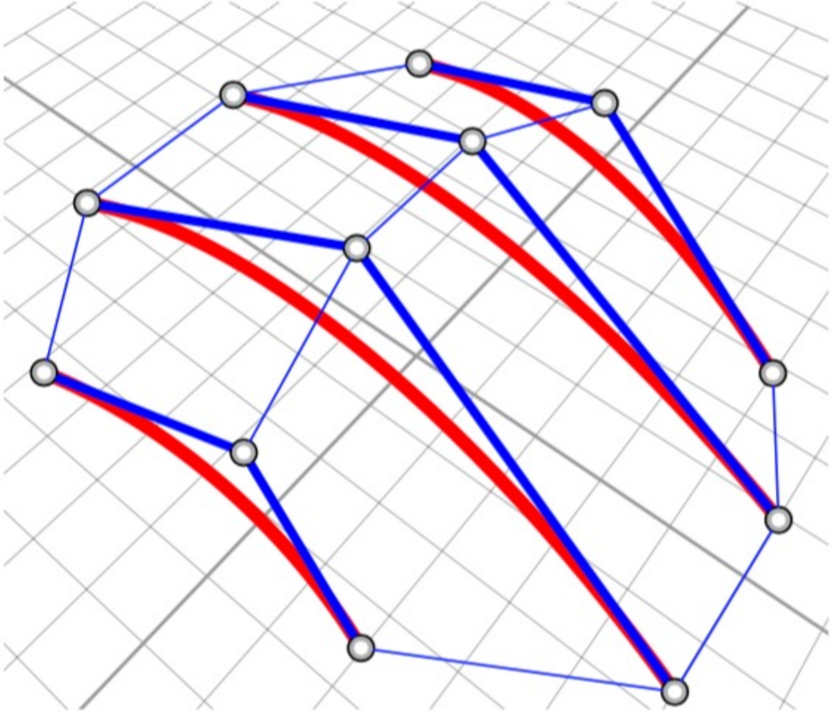
$$\mathbf{b}_i = \mathbf{b}_i(v) = \sum_{j=0}^{n} \mathbf{b}_{i,j} B_j^n(v).$$



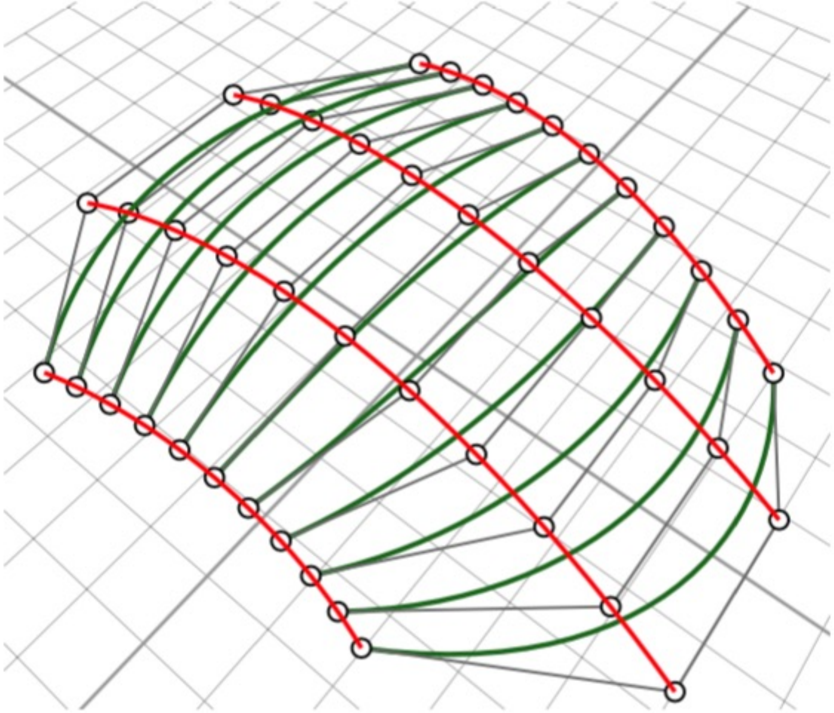
$$\boxed{\mathbf{b}^{m,n}(u,v) = \sum_{i=0}^{m} \sum_{j=0}^{n} \mathbf{b}_{i,j} B_i^m(u) B_j^n(v).}$$

Some special curves

- Four boundary Bézier curves.
- The control curves are the Bézier curves given by a « row » or a « column » of control points.



- Iso-parameter curves are Bézier curves *on the surface* that are obtained by fixing the value of one of the two parameters.

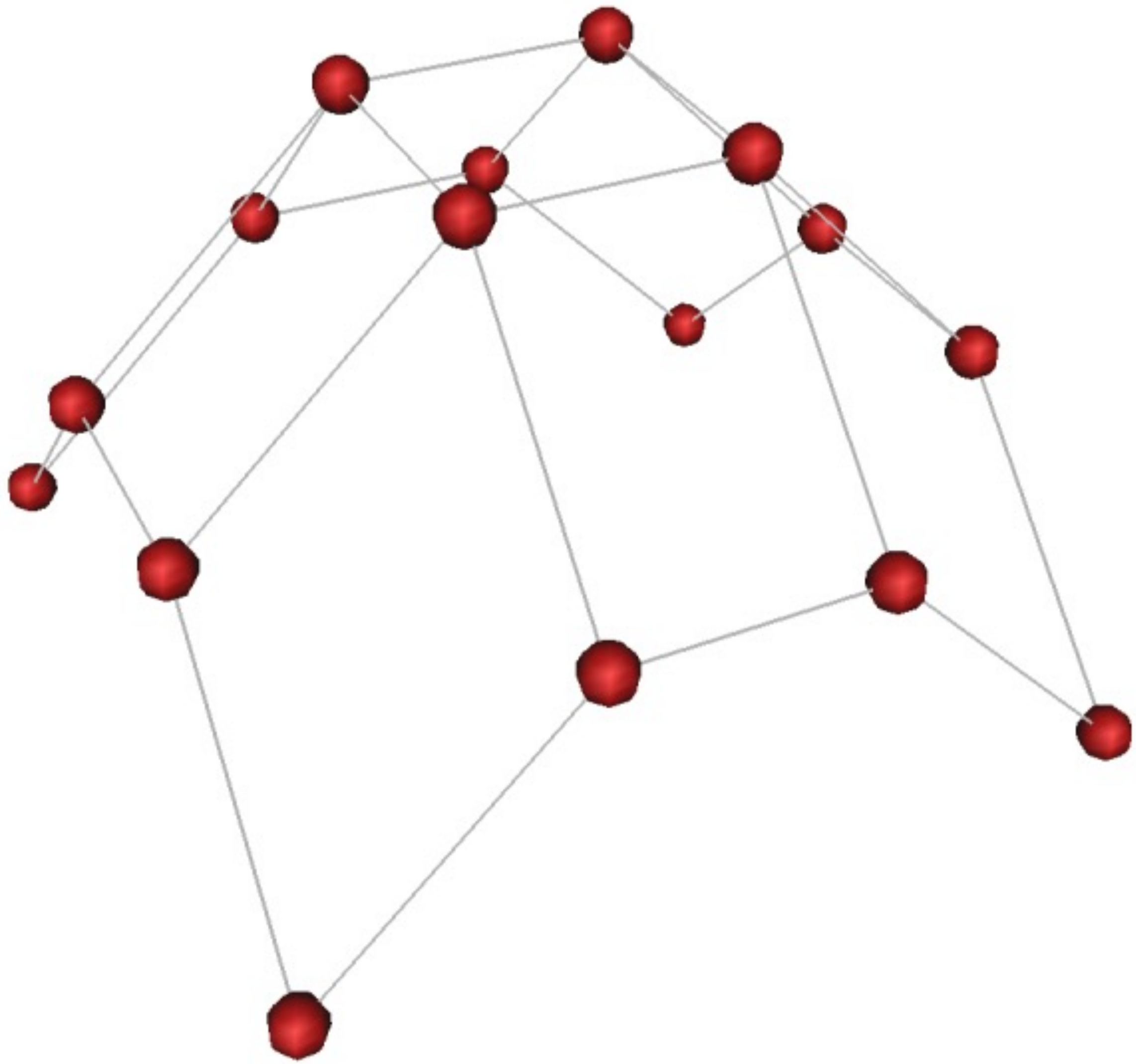


The de Casteljau algorithm

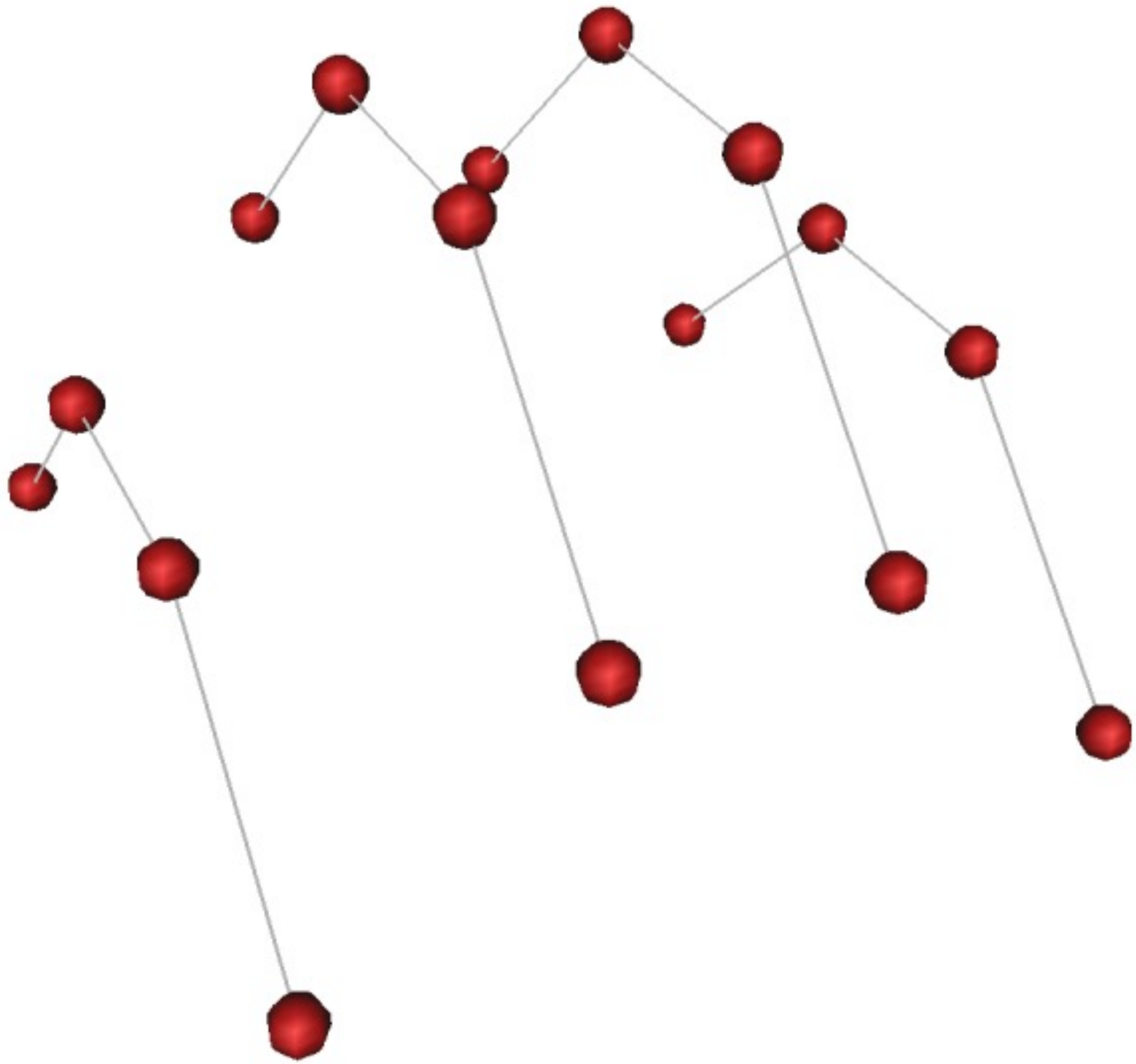
- To evaluate a point on a tensor product Bézier patch corresponding to a given parameter *(s,t),* one can:
  - Evaluate the *v*-control curves at $v=t$,
  - Then evaluate this iso-parameter curve at $u=s$.
  - Alternatively, one could proceed with the u-control curves first.

- This algorithm is also used to cut a tensor product Bézier patch into two pieces, either in the *u* or *v* parameter

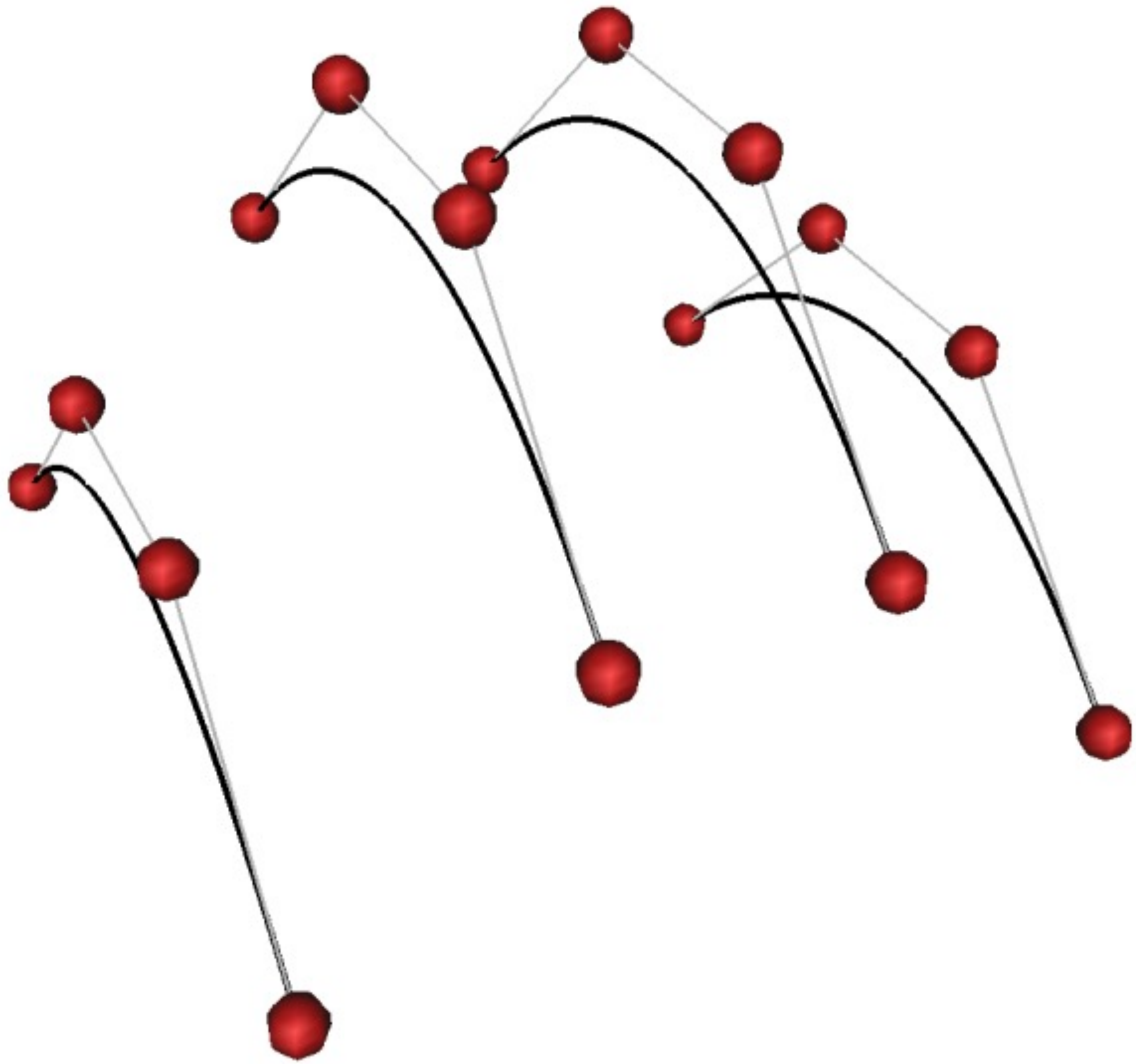
A « bicubic » grid of control points



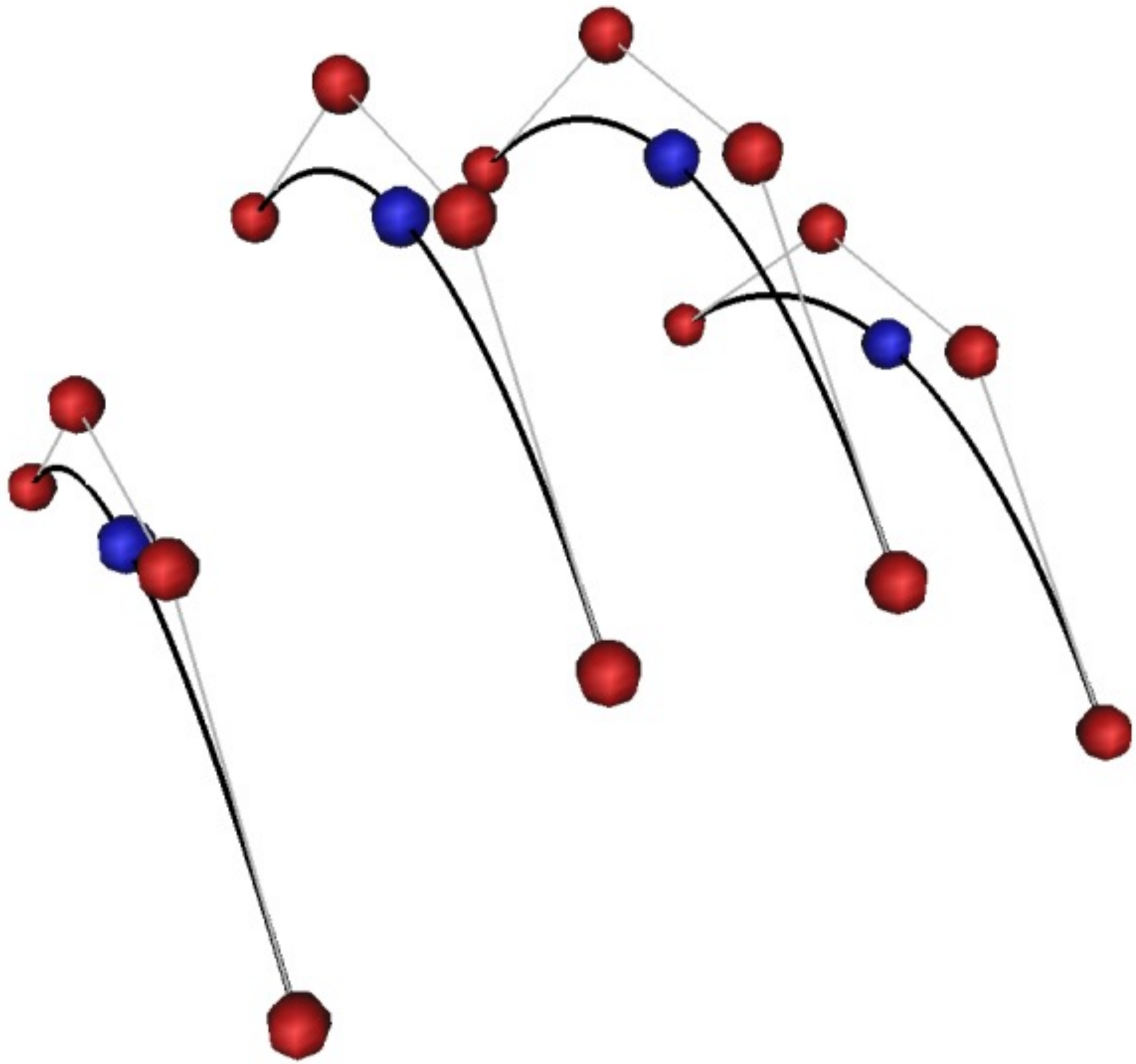
The de Casteljau algorithm



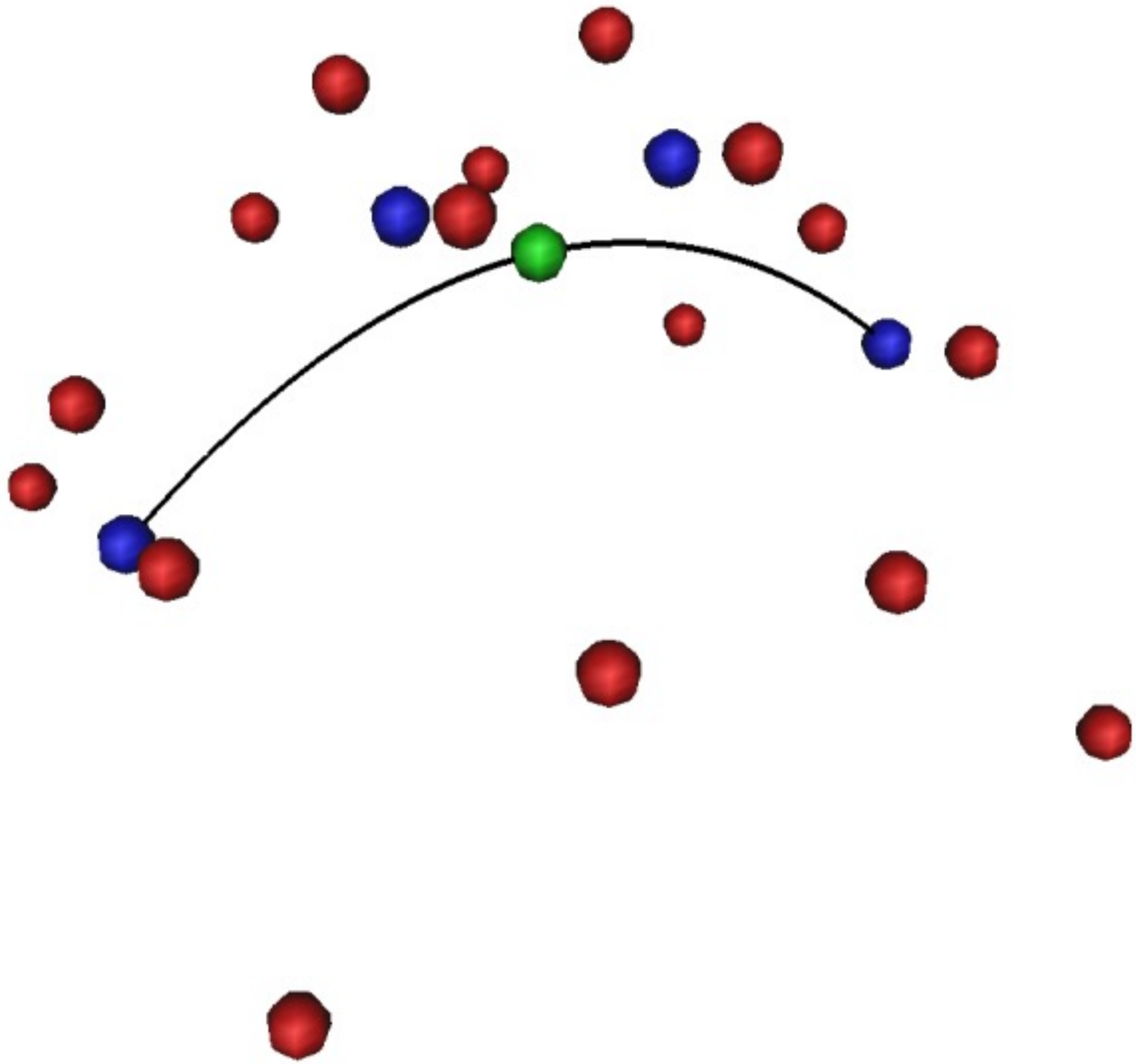
The de Casteljau algorithm



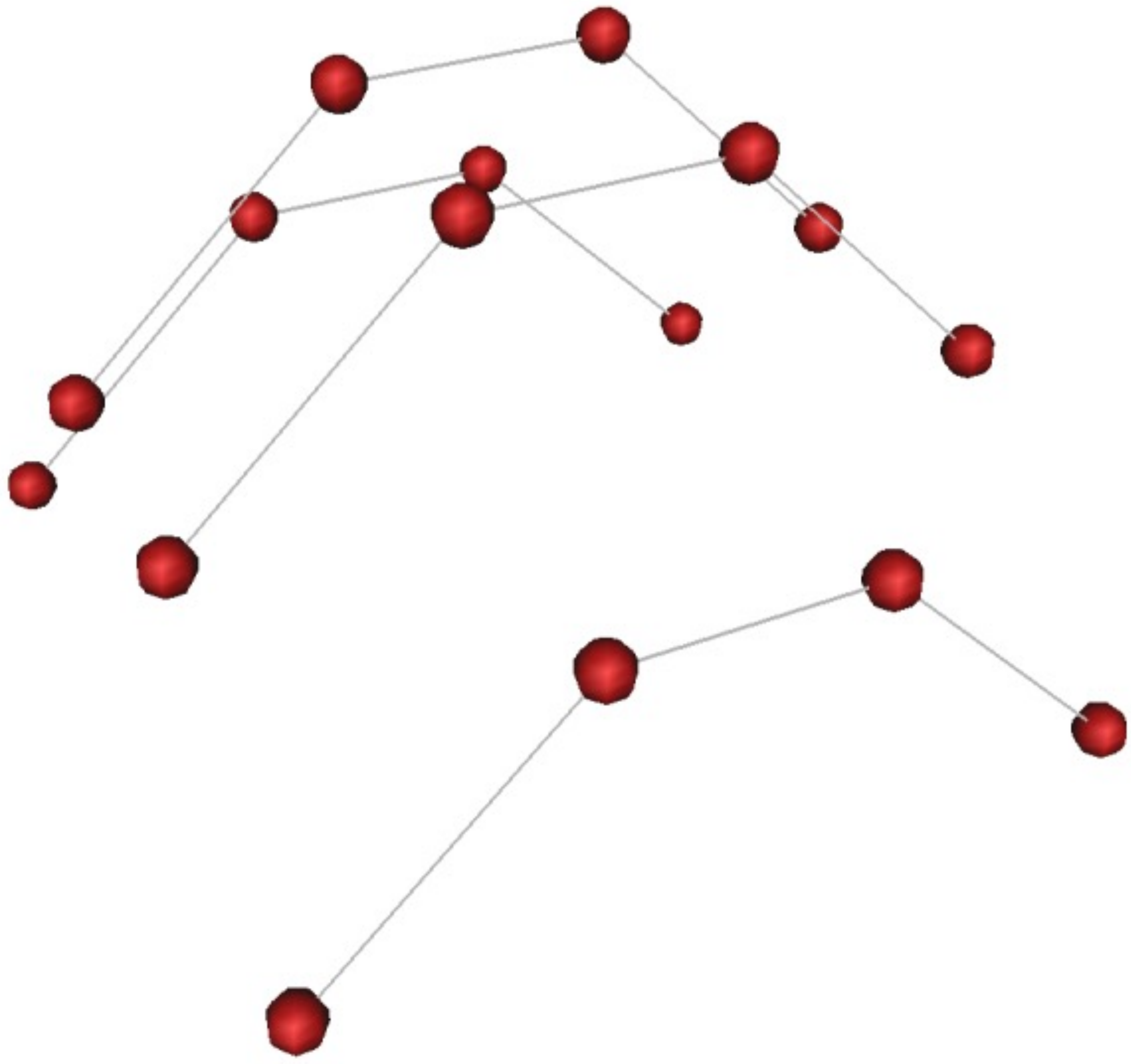
The de Casteljau algorithm



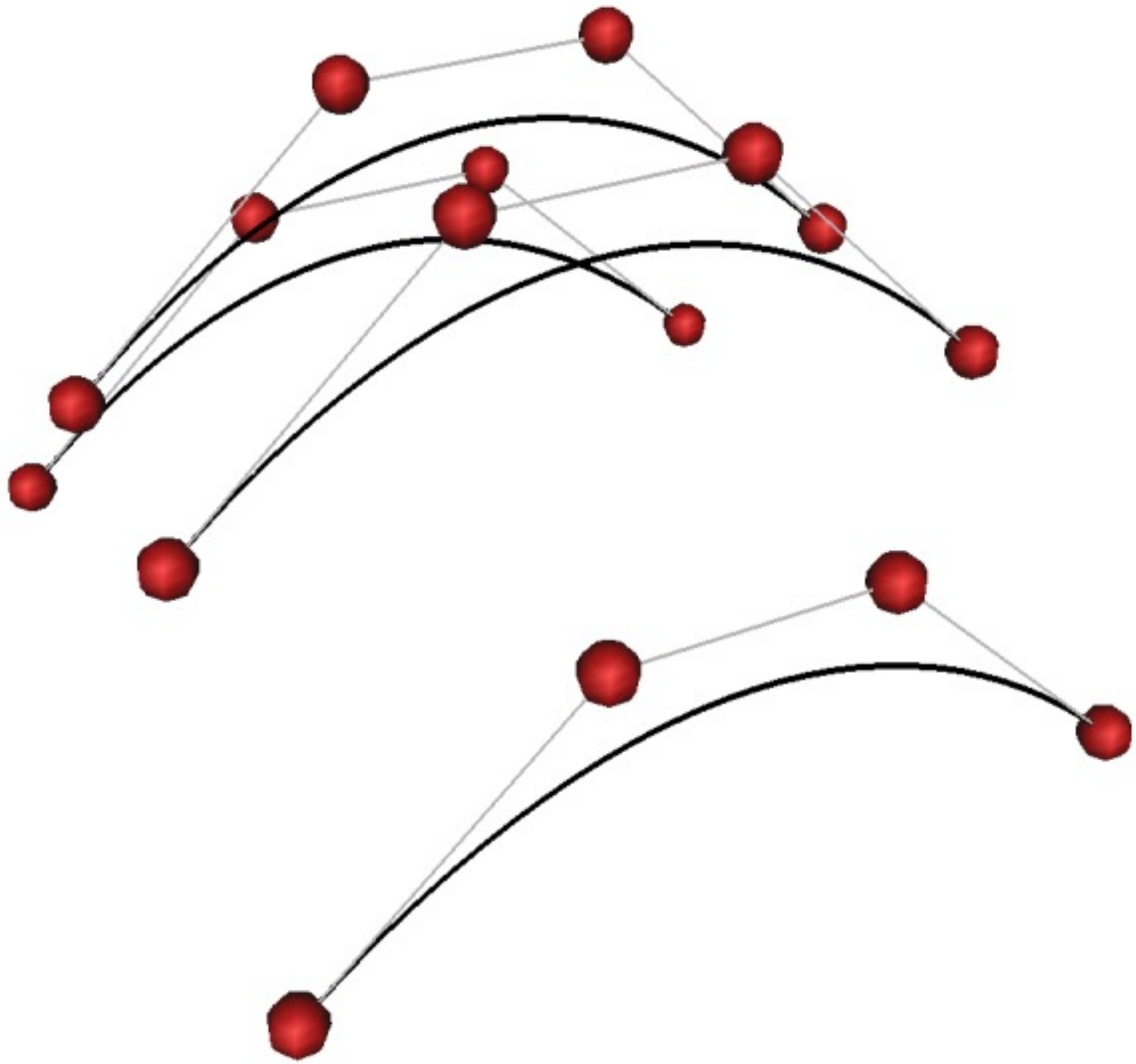
The de Casteljau algorithm



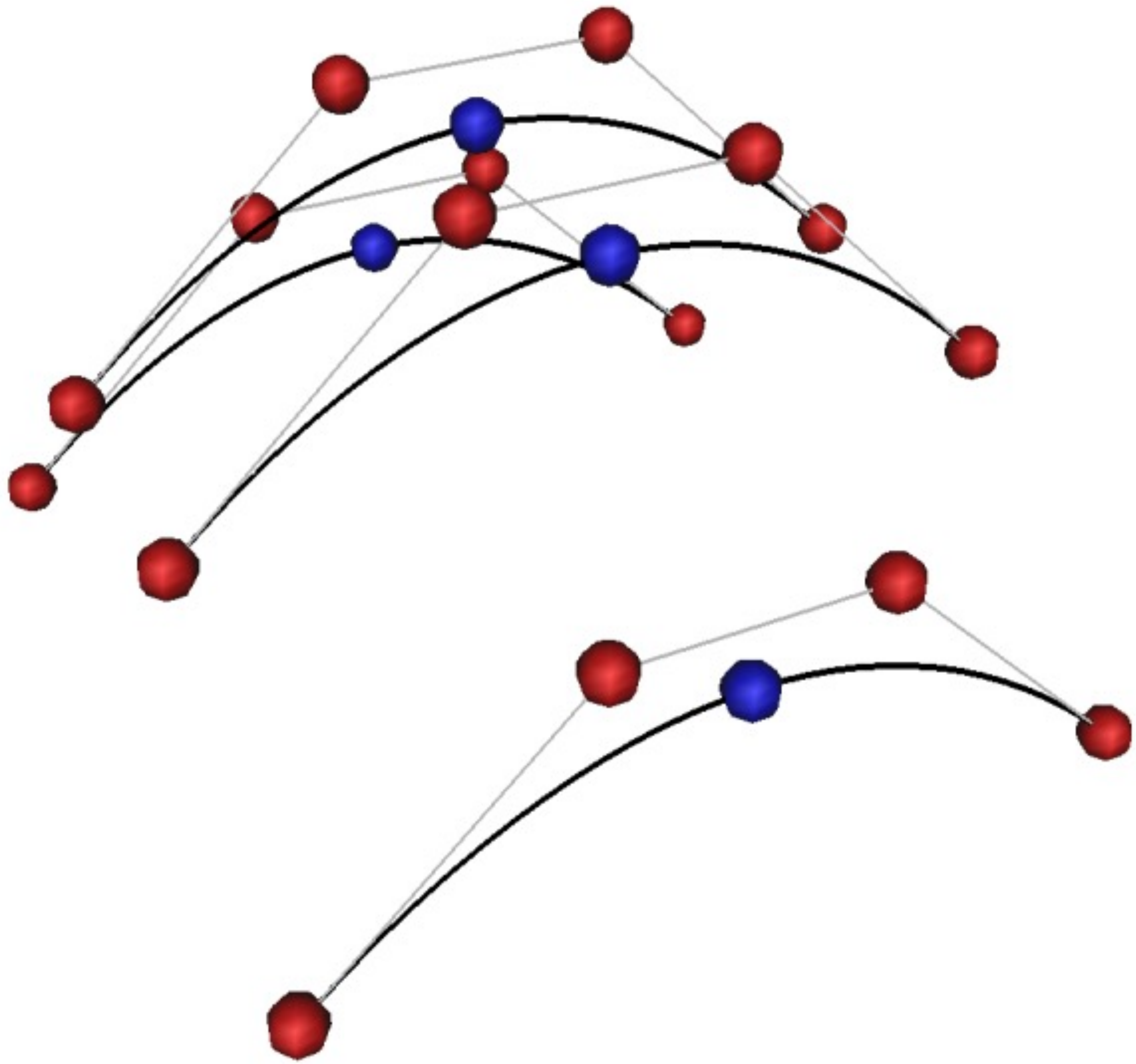
The de Casteljau algorithm



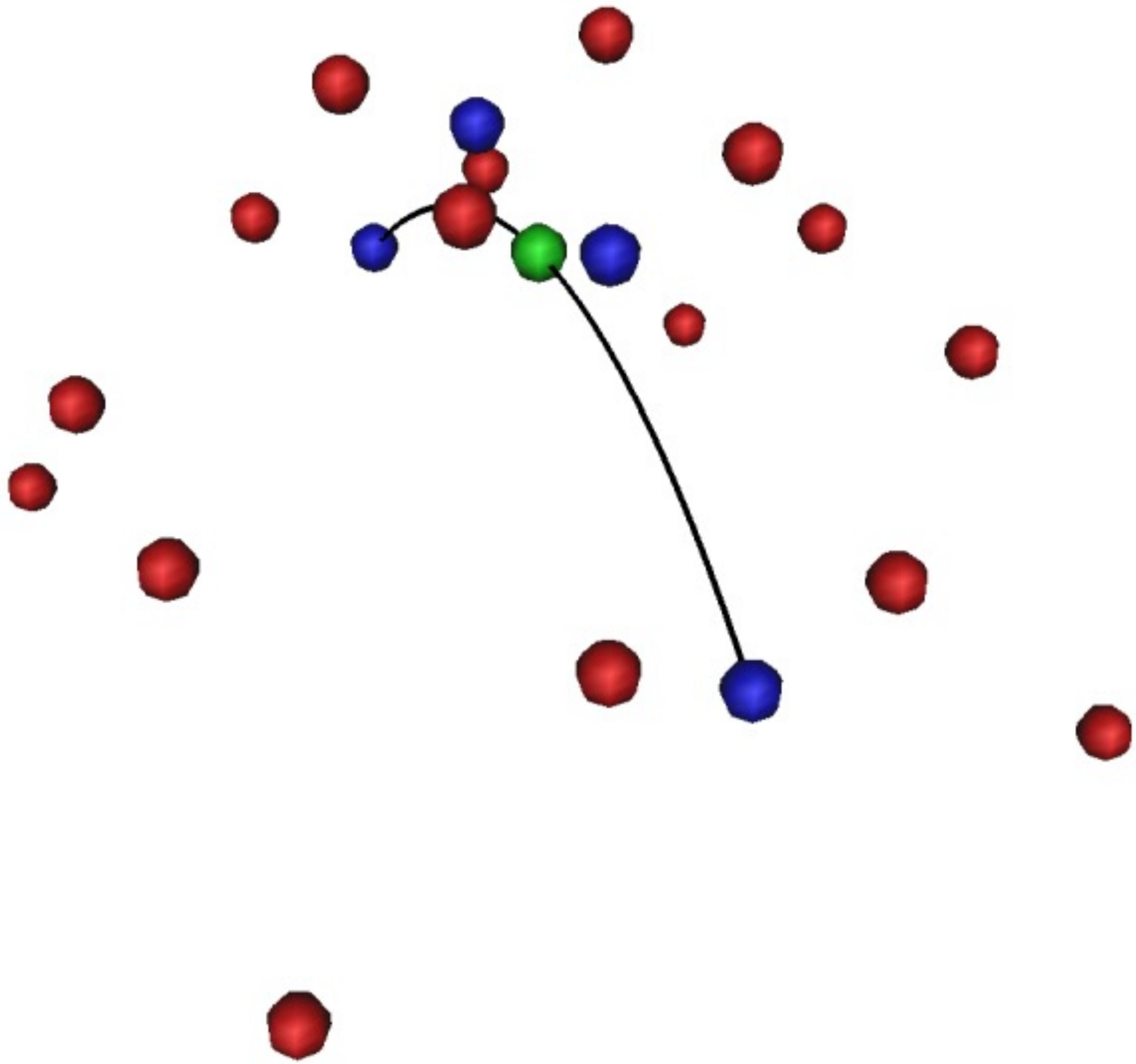
The de Casteljau algorithm



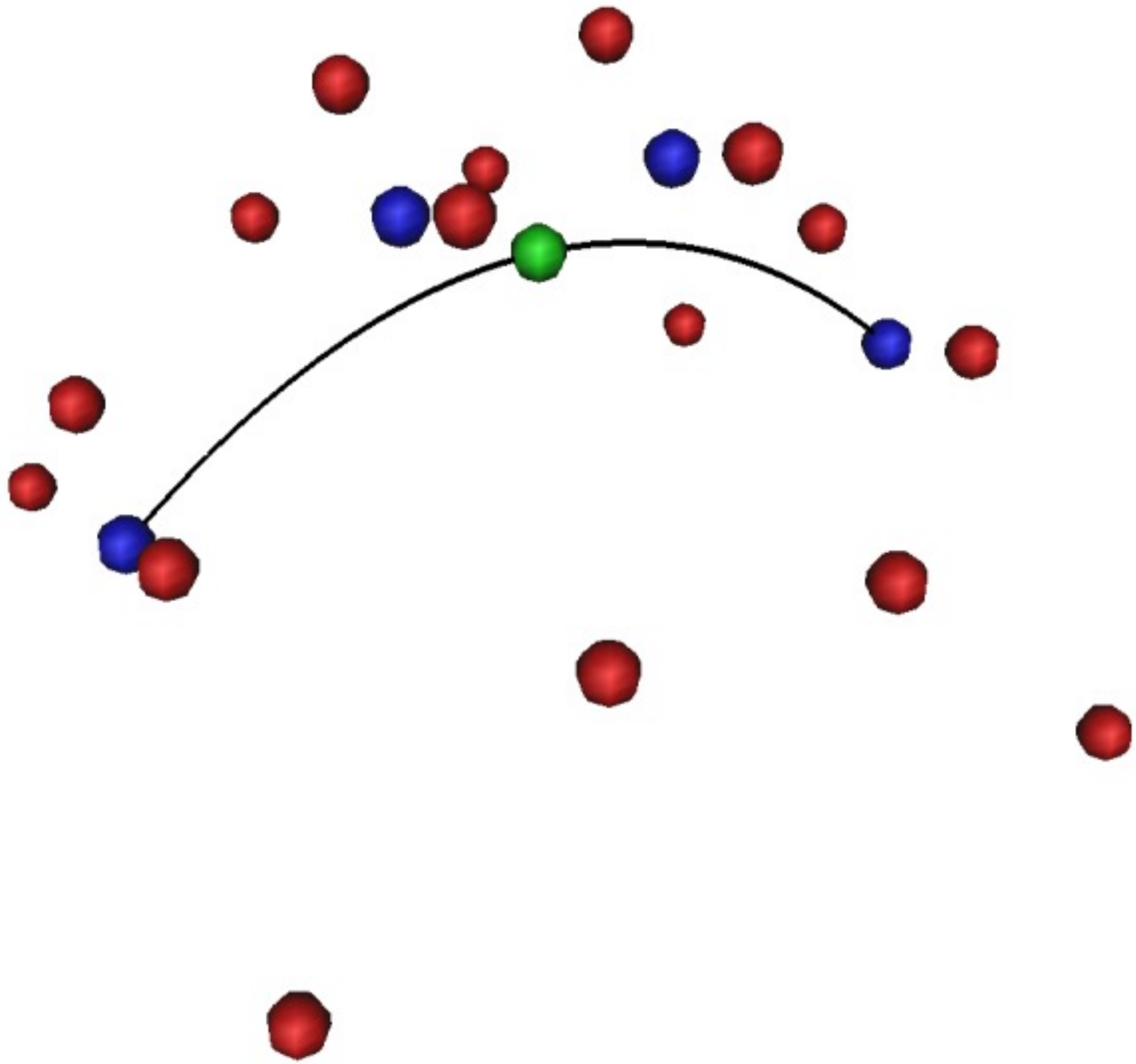
The de Casteljau algorithm



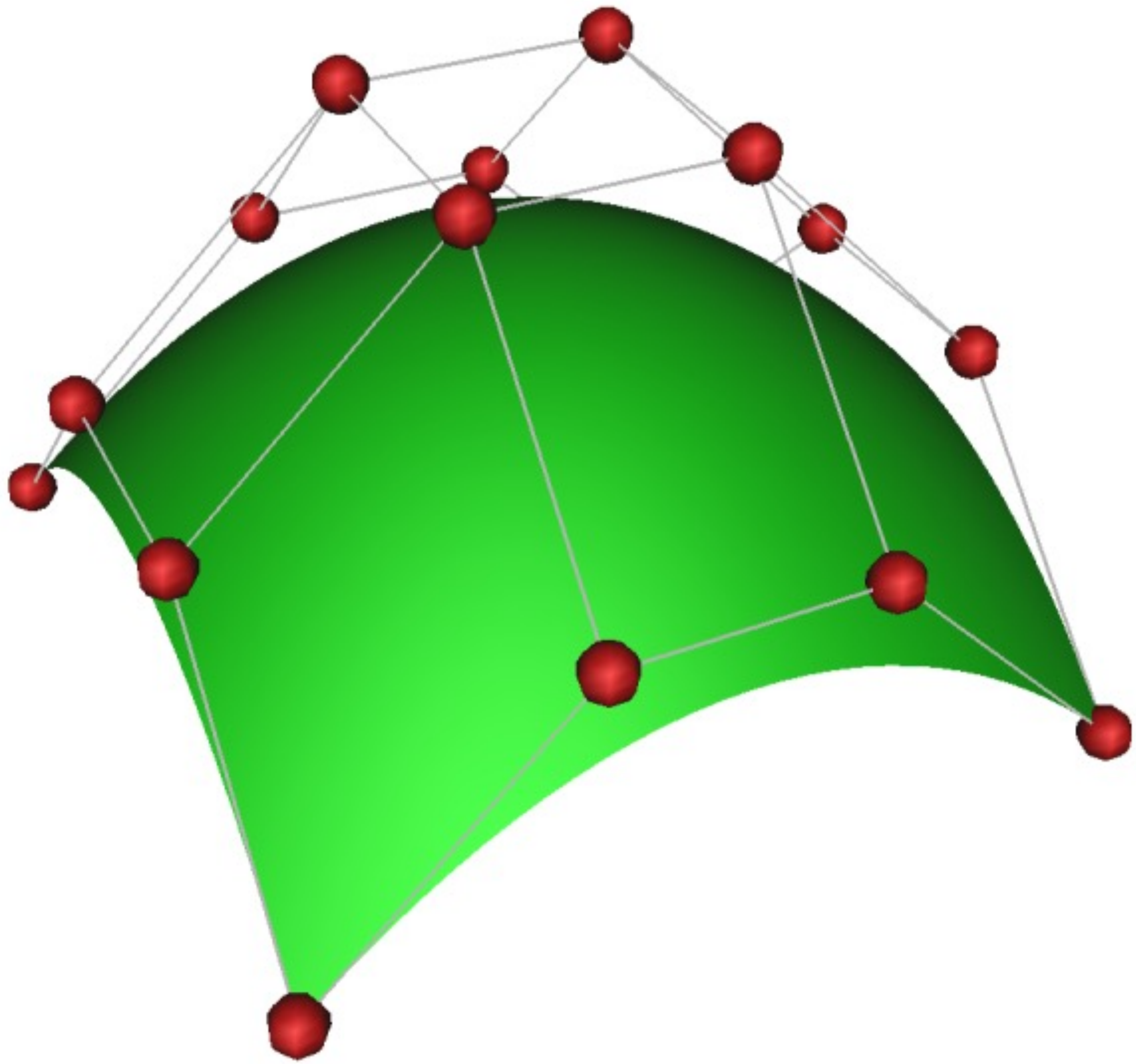
The de Casteljau algorithm



The de Casteljau algorithm

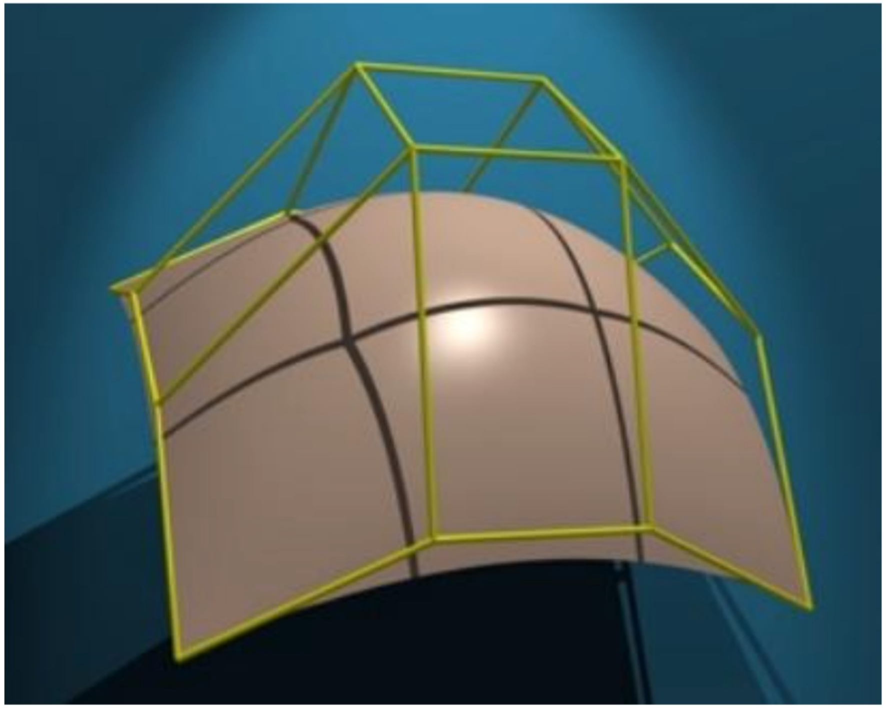


A « bicubic » Bézier patch



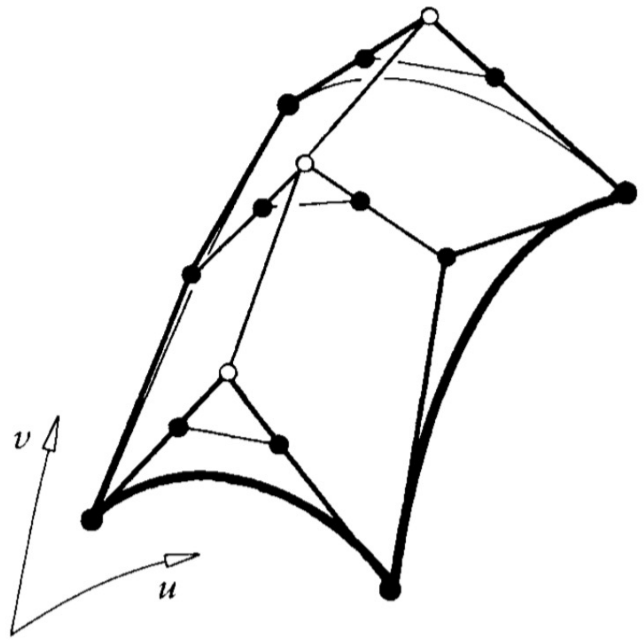
Properties of tensor product Bézier patch

- Affine invariance (follows from the definition)
- Convex hull property : the patch lies in convex hull of its control points
- The patch interpolates the four corner control points
- *Variation diminishing property is NOT inherited from the univariate case*



Degree elevation

Goal : rewrite a Bézier patch of degree *(m,n)* as one of degree *(m+1,n)*



- Find the new coefficients such that

$$\mathbf{b}^{m,n}(u,v) = \sum_{j=0}^{n} \left[\sum_{i=0}^{m+1} \mathbf{b}_{i,j}^{(1,0)} B_i^{m+1}(u) \right] B_j^n(v)$$

- It can be reduced to a series of univariate problems.

$$\mathbf{b}_{i,j}^{(1,0)} = \frac{i}{m+1}\mathbf{b}_{i-1,j} + \left(1 - \frac{i}{m+1}\right)\mathbf{b}_{i,j}; \begin{cases} i = 0, \ldots, m+1 \\ j = 0, \ldots, n. \end{cases}$$

Derivatives

As in the curve case, taking derivatives is accomplished by differencing the control points

- Partial derivative in u-direction:

$$\frac{\partial}{\partial u}\mathbf{b}^{m,n}(u,v) = \sum_{j=0}^{n}\left[\frac{\partial}{\partial u}\sum_{i=0}^{m}\mathbf{b}_{i,j}B_i^m(u)\right]B_j^n(v)$$

$$\frac{\partial}{\partial u}\mathbf{b}^{m,n}(u,v) = m\sum_{j=0}^{n}\sum_{i=0}^{m-1}\Delta^{1,0}\mathbf{b}_{i,j}B_i^{m-1}(u)B_j^n(v) \qquad \Delta^{1,0}\mathbf{b}_{i,j} = \mathbf{b}_{i+1,j} - \mathbf{b}_{i,j}$$

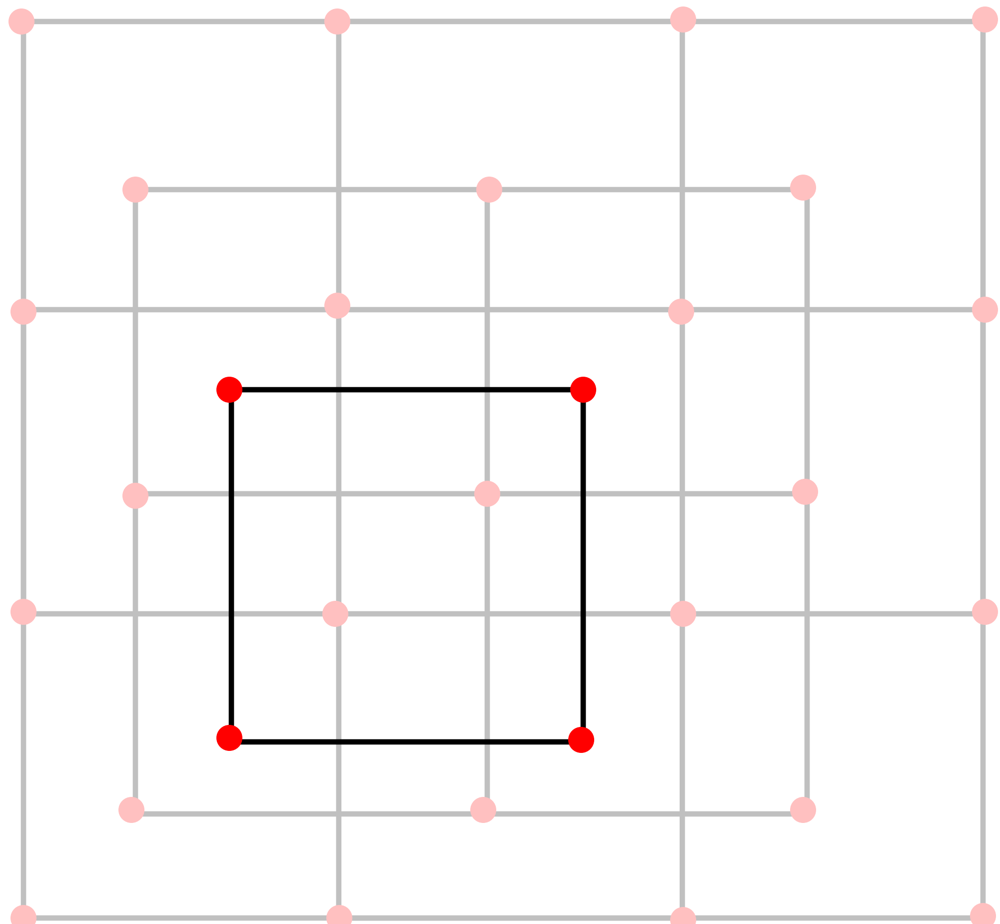
$$\frac{\partial^r}{\partial u^r}\mathbf{b}^{m,n}(u,v) = \frac{m!}{(m-r)!}\sum_{j=0}^{n}\sum_{i=0}^{m-r}\Delta^{r,0}\mathbf{b}_{i,j}B_i^{m-r}(u)B_j^n(v) \quad \Delta^{r,0}\mathbf{b}_{i,j} = \Delta^{r-1,0}\mathbf{b}_{i+1,j} - \Delta^{r-1,0}\mathbf{b}_{i,j}$$

- General formula:

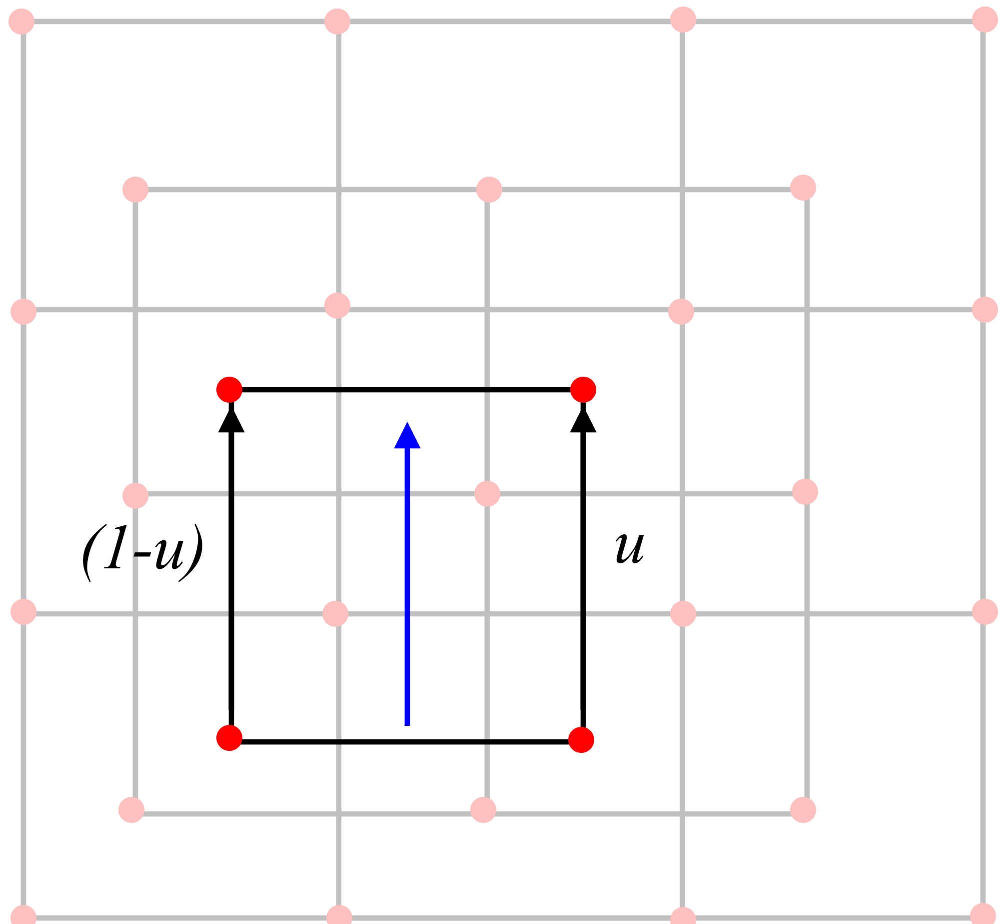
$$\frac{\partial^{r+s}}{\partial u^r \partial v^s}\mathbf{b}^{m,n}(u,v)$$

$$= \frac{m!n!}{(m-r)!(n-s)!}\sum_{i=0}^{m-r}\sum_{j=0}^{n-s}\Delta^{r,s}\mathbf{b}_{i,j}B_i^{m-r}(u)B_j^{n-s}(v)$$

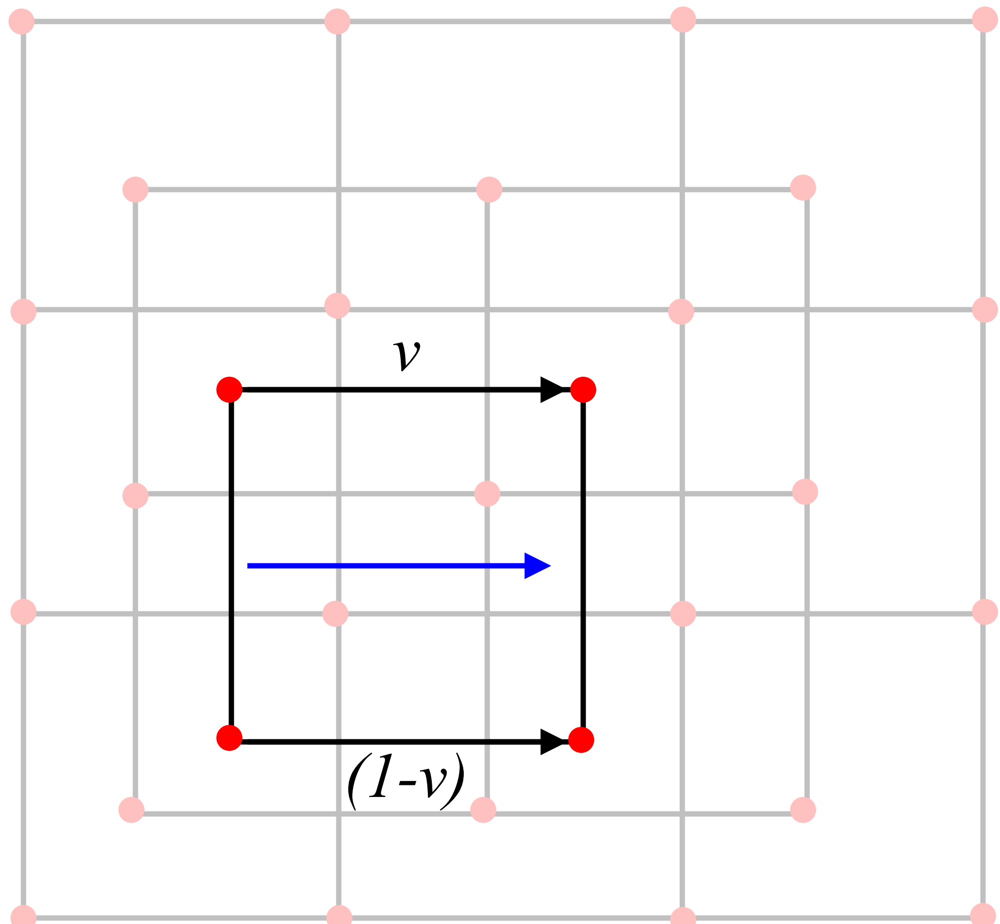
Derivatives using de Casteljau's algorithm



Derivatives using de Casteljau's algorithm



Derivatives using de Casteljau's algorithm



Blossoming for tensor-product patches

$$b(u_1, \ldots, u_m | v_1, \ldots, v_n)$$

- **Symmetry**:

For any permutation q of $(1,\ldots,m)$ and r of $(1,\ldots,n)$:

$$b(u_{q(1)}, \ldots, u_{q(m)} | v_{r(1)}, \ldots, v_{r(n)}) = b(u_1, \ldots, u_m | v_1, \ldots, v_n)$$

- **Multi-affine**:

$$\begin{aligned} b(u_1, \ldots, (1-d)u_k + ds_k, \ldots, u_m | v_1, \ldots, (1-e)v_k + ew_k, \ldots, v_n) \\ = (1-d)(1-e).b(u_1, \ldots, u_k, \ldots, u_m | v_1, \ldots, v_k, \ldots, v_n) \\ +(1-d)e.b(u_1, \ldots, u_k, \ldots, u_m | v_1, \ldots, w_k, \ldots, v_n) \\ +de.b(u_1, \ldots, s_k, \ldots, u_m | v_1, \ldots, w_k, \ldots, v_n) \\ +d(1-e).b(u_1, \ldots, s_k, \ldots, u_m | v_1, \ldots, v_k, \ldots, v_n) \end{aligned}$$

- **Diagonal**:

$$b(u, \ldots, u | v, \ldots, v) = b(u, v)$$

Curves on Bézier surfaces

- Given two points in the domain of a Bézier patch of degree *(n,n)*, they define a straight line

$$\mathbf{u}(t) = (1-t)\mathbf{p} + t\mathbf{q} \qquad \mathbf{p} = (\mathbf{p}_u, \mathbf{p}_v) \text{ and } \mathbf{q} = (\mathbf{q}_u, \mathbf{q}_v)$$

- This line is mapped to a curve on the surface. What are its Bézier control points ?

- Using the blossom of the surface, a point on this curve is given by

$$\mathbf{b}[((1-t)\mathbf{p}_u + t\mathbf{q}_u)^{<n>} \mid ((1-t)\mathbf{p}_v + t\mathbf{q}_v)^{<n>}].$$

Curves on Bézier surfaces

$$\sum_{i+j=n} \binom{n}{i,j} (1-t)^i t^j \mathbf{b}[\mathbf{p}_u^{<i>}, \mathbf{q}_u^{<j>} \mid ((1-t)\mathbf{p}_v + t\mathbf{q}_v)^{<n>}].$$

$$\binom{n}{i,j} = \frac{n!}{i!j!}$$

$$\sum_{i+j=n} \sum_{r+s=n} \binom{n}{i,j} \binom{n}{r,s} (1-t)^i t^j (1-t)^r t^s \mathbf{b}[\mathbf{p}_u^{<i>}, \mathbf{q}_u^{<j>} \mid \mathbf{p}_v^{<r>}, \mathbf{q}_v^{<s>}]$$

$$\sum_{i=0}^{n} \sum_{j=0}^{n} \binom{n}{i} \binom{n}{j} (1-t)^{2n-i-j} t^{i+j} \mathbf{b}[\mathbf{p}_u^{<i>}, \mathbf{q}_u^{<n-i>} \mid \mathbf{p}_v^{<j>}, \mathbf{q}_v^{<n-j>}]$$

$$\sum_{k=0}^{2n} \sum_{i+j=k} \frac{\binom{n}{i}\binom{n}{j}}{\binom{2n}{k}} B_k^{2n} \mathbf{b}[\mathbf{p}_u^{<i>}, \mathbf{q}_u^{<n-i>} \mid \mathbf{p}_v^{<j>}, \mathbf{q}_v^{<n-j>}]$$

curve of degree *2n*

Finally, the control points are given by :

$$\mathbf{c}_k = \frac{\binom{n}{i}\binom{n}{j}}{\binom{2n}{k}} \mathbf{b}[\mathbf{p}_u^{<i>}, \mathbf{q}_u^{<n-i>} \mid \mathbf{p}_v^{<j>}, \mathbf{q}_v^{<n-j>}]$$

Normal vectors

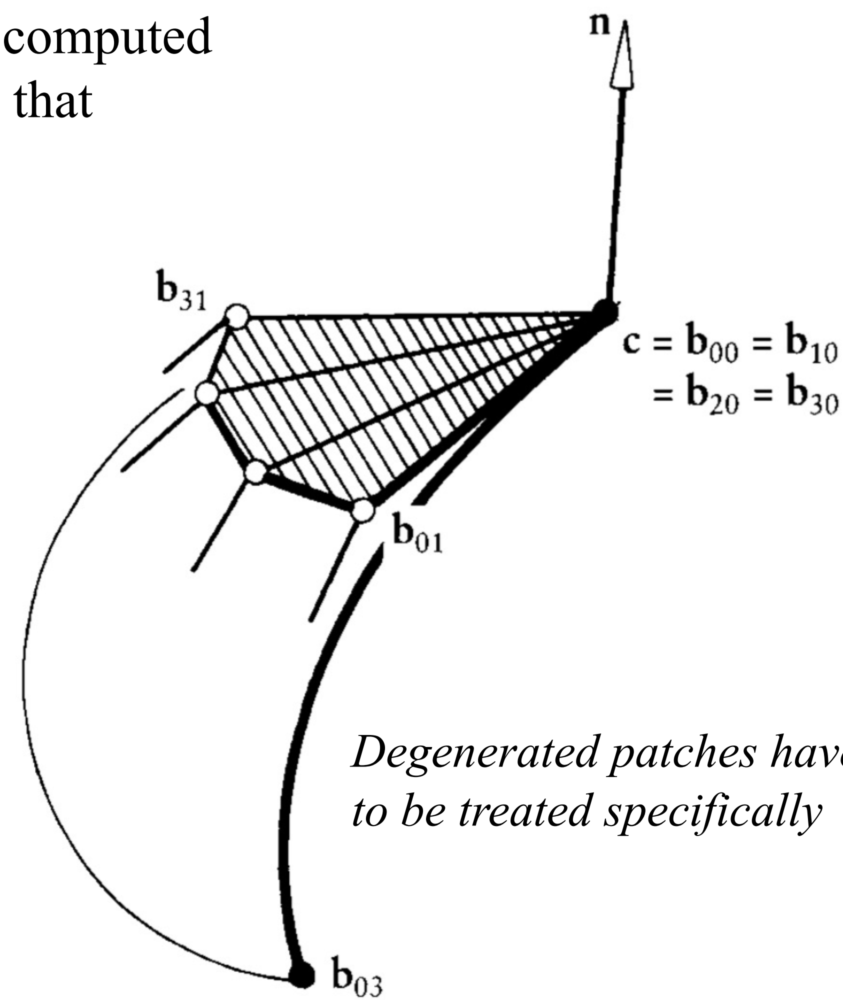
- The normal vector of a surface can be computed from the cross product of any two vectors that are tangent to the surface.

- Using the partial derivatives:

$$\mathbf{n}(u,v) = \frac{\frac{\partial}{\partial u}\mathbf{b}^{m,n}(u,v) \wedge \frac{\partial}{\partial v}\mathbf{b}^{m,n}(u,v)}{\|\frac{\partial}{\partial u}\mathbf{b}^{m,n}(u,v) \wedge \frac{\partial}{\partial v}\mathbf{b}^{m,n}(u,v)\|}$$

- At the four corner points, we have for instance:

$$\mathbf{n}(0,0) = \frac{\Delta^{1,0}\mathbf{b}_{0,0} \wedge \Delta^{0,1}\mathbf{b}_{0,0}}{\|\Delta^{1,0}\mathbf{b}_{0,0} \wedge \Delta^{0,1}\mathbf{b}_{0,0}\|}$$



Degenerated patches have to be treated specifically

The matrix form of a Bézier patch

- The « geometry matrix » of the patch:

$$\mathbf{b}^{m,n}(u,v) = \begin{bmatrix} B_0^m(u) & \dots & B_m^m(u) \end{bmatrix} \begin{bmatrix} \mathbf{b}_{00} & \dots & \mathbf{b}_{0n} \\ \vdots & & \vdots \\ \mathbf{b}_{m0} & \dots & \mathbf{b}_{mn} \end{bmatrix} \begin{bmatrix} B_0^n(v) \\ \vdots \\ B_n^n(v) \end{bmatrix}$$

- Basis transformation to the monomial basis:

$$\mathbf{b}^{m,n}(u,v) = \begin{bmatrix} u^0 & \dots & u^m \end{bmatrix} M^{\mathrm{T}} \begin{bmatrix} \mathbf{b}_{00} & \dots & \mathbf{b}_{0n} \\ \vdots & & \vdots \\ \mathbf{b}_{m0} & \dots & \mathbf{b}_{mn} \end{bmatrix} N \begin{bmatrix} v^0 \\ \vdots \\ v^n \end{bmatrix}$$

where the matrices M and N are given by :

$$m_{ij} = (-1)^{j-i} \binom{m}{j} \binom{j}{i} \qquad n_{ij} = (-1)^{j-i} \binom{n}{j} \binom{j}{i}$$

Bézier patch of a graph

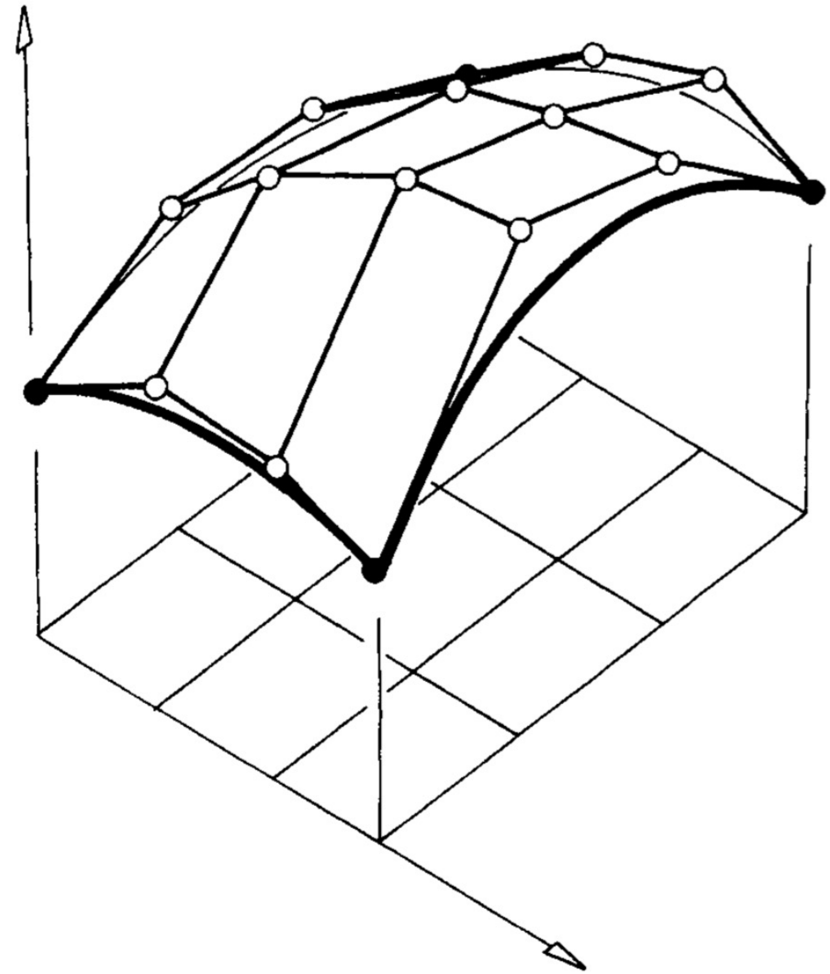
- Parameterization of the form

$$\mathbf{x}(u,v) = \begin{bmatrix} u \\ v \\ f(u,v) \end{bmatrix}$$

- If $f(x,y) = \sum_{i}^{m} \sum_{j}^{n} b_{ij} B_i^m(x) B_j^n(y)$

  then the control points of the patch are given by:

$$\mathbf{b}_{ij} = \begin{bmatrix} i/m \\ j/n \\ b_{ij} \end{bmatrix}$$



The control points are located over a regular partition of the domain rectangle

Composite Bézier surfaces

- Suppose given two Bézier patches of degree *(m,n)*:

left patch :

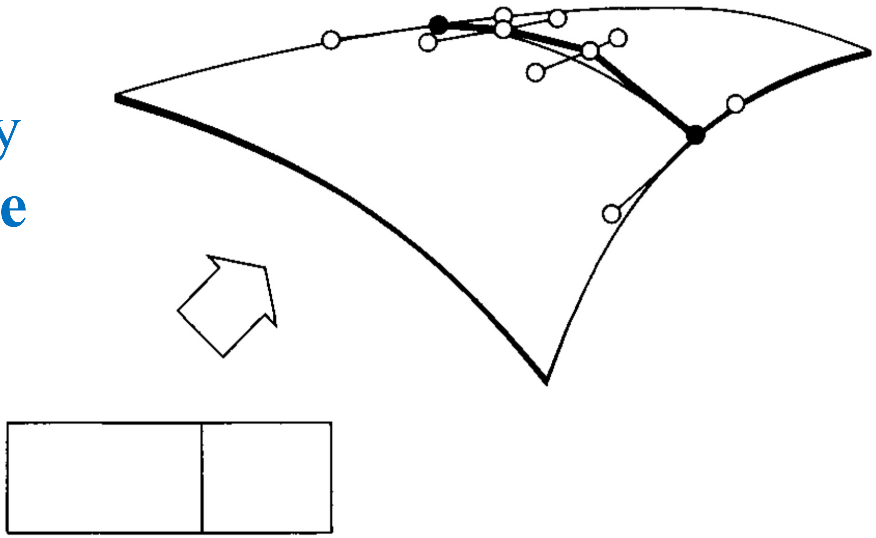
$\{\mathbf{b}_{ij}\};\ 0 \le i \le m,\ 0 \le j \le n$

right patch :

$\{\mathbf{b}_{ij}\};\ m \le i \le 2m,\ 0 \le j \le n$

- To get r times differentiability accross their common boundary, one must evaluate the *u*-partial derivatives.

It is equivalent to the fact that every pair of **adjacent control curves are *r-differentiable***

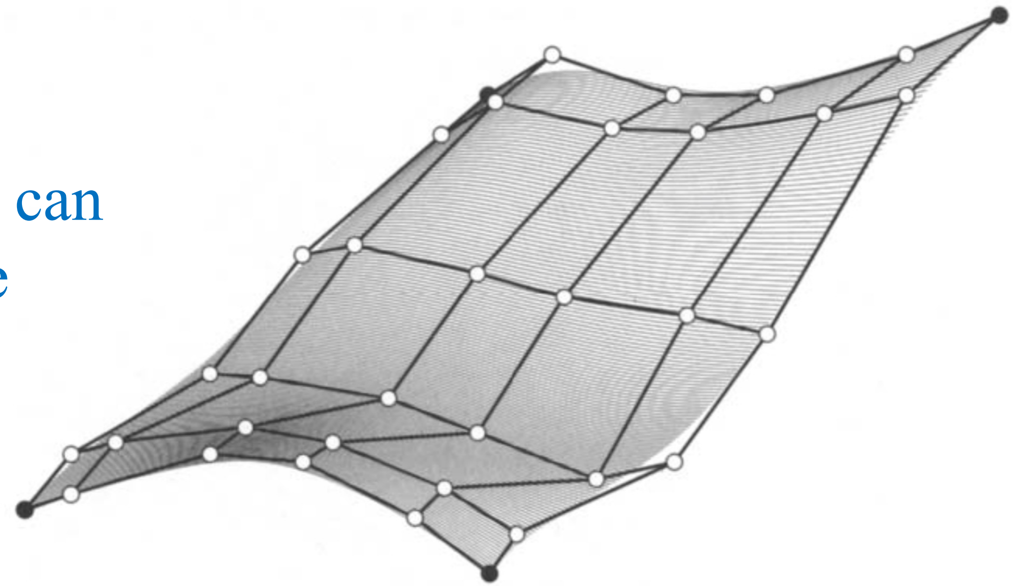


Tensor product B-spline surfaces

$$b(u,v) = \sum_i \sum_j b_{i,j} N_i^m(u) N_j^n(v)$$

- Need two knot sequences : one in the *u*-direction and one in the *v*-direction

All methods and algorithms can be reduced to the curve case



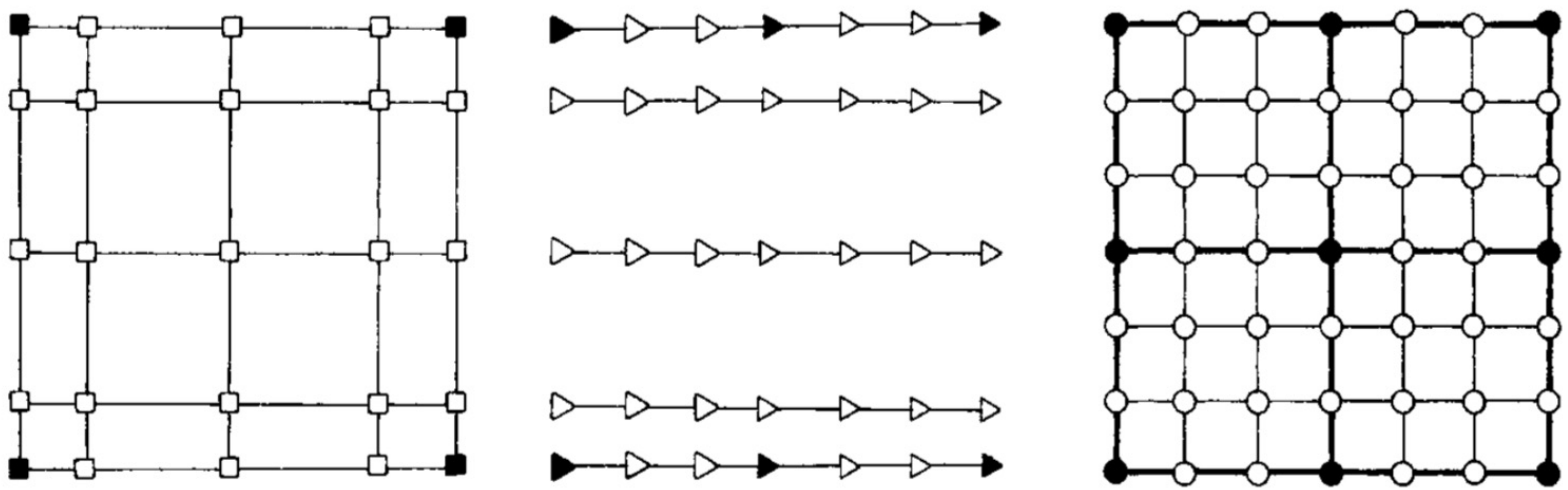
A bicubic B-spline surface consisting of 3x3 Bézier patches

Conversion to Bézier patches

Conversion to Bézier patches is done by using the univariate method :

- Interpret the B-spline control net row by row as univariate B-spline polygons
- convert them to piecewise Bezier form.

- The Bezier points thus obtained may be interpreted, column by column, as B-spline polygons, which we may again transform to Bezier form one by one.



Bringing a bicubic B-spline surface into piecewise bicubic Bézier form: we first perform B-spline–Bézier curve conversion row by row, then column by column.

Rational Bézier and B-spline surfaces

- Bézier and B-spline surfaces can be generalized to their rational counterpart as for curves

- Rational Bézier and B-spline surfaces are projections of a 4D tensor product of B-spline surface

Rational Bézier surface

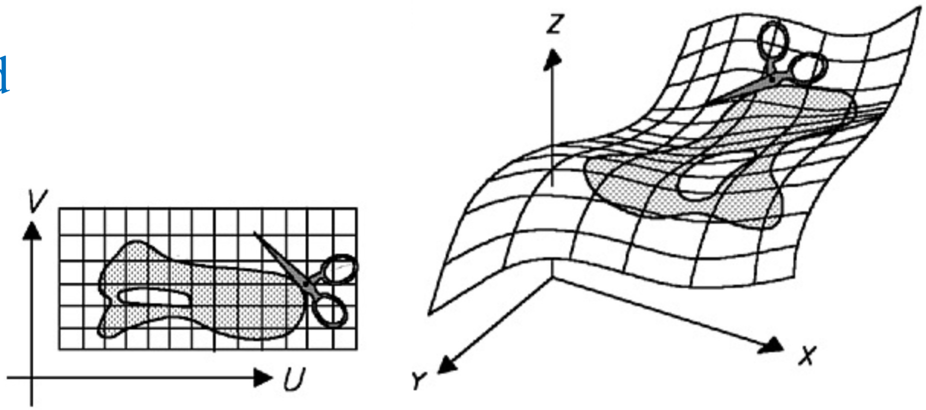
$$\frac{\sum_i \sum_j w_{i,j} b_{i,j} B_i^m(u) B_j^n(v)}{\sum_i \sum_j w_{i,j} B_i^m(u) B_j^n(v)}$$

Rational B-spline surface

$$\frac{\sum_i \sum_j w_{i,j} b_{i,j} N_i^m(u) N_j^n(v)}{\sum_i \sum_j w_{i,j} N_i^m(u) N_j^n(v)}$$

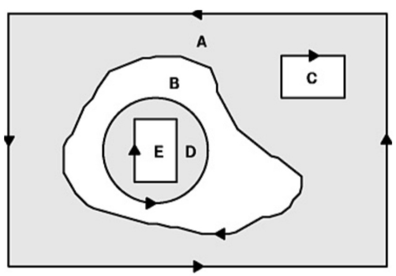
Trimmed surfaces

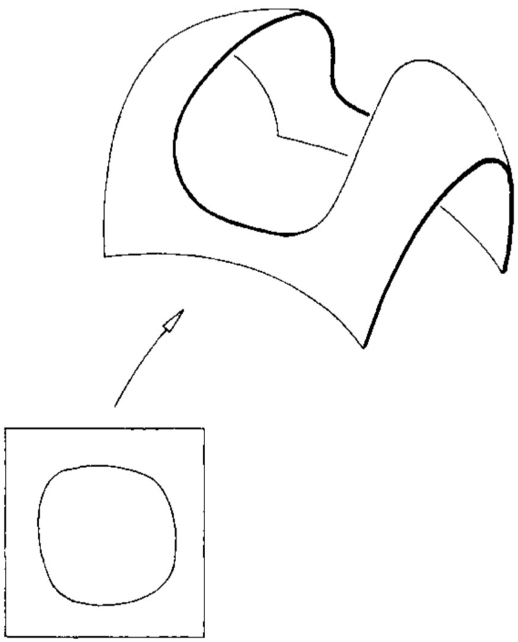
- A parametric curve in the domain is used to trim the surface



- A curve of degree *d* yields a curve of degree *(m+n)d* on the surface

- Orientation of the curve defines the inside/outside points. The test is done by launching a « random » ray.





B-Rep of CAD models

A CAD model : on the left, the trimmed surfaces, on the right the surfaces are printed without trimming