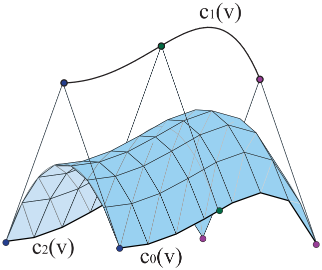
Superficies de Bézier

Leonardo Fernández Jambrina

Matemática Aplicada
E.T.S.I. Navales
Universidad Politécnica de Madrid

De curvas a superficies

- Las superficies son mucho más complejas geométricamente que las curvas.
- No obstante, el diseño está basado tradicionalmente en curvas.
- “Buenas curvas hacen buenas superficies” (refrán popular)



Generación de superficies

- ¿Cómo podemos generar superficies a partir de curvas?
- Si desplazamos los vértices del polígono de control a lo largo de curvas, $\{c_0(v), \ldots, c_m(v)\}$, $v \in [0, 1]$,

$$c(u, v) = \sum_{i=0}^{m} c_i(v) B_i^m(u),$$

las curvas de Bézier $c(u, v_0)$, de polígonos $\{c_0(v_0), \ldots, c_m(v_0)\}$, evolucionan en el espacio describiendo una superficie $c(u, v)$.

Generación de superficies

- ¿Cómo podemos generar superficies a partir de curvas?
- Si desplazamos los vértices del polígono de control a lo largo de curvas, $\{c_0(v), \ldots, c_m(v)\}$, $v \in [0,1]$,

$$c(u,v) = \sum_{i=0}^{m} c_i(v) B_i^m(u),$$

  las curvas de Bézier $c(u, v_0)$, de polígonos $\{c_0(v_0), \ldots, c_m(v_0)\}$, evolucionan en el espacio describiendo una superficie $c(u,v)$.
- Por coherencia, parece razonable postular que la evolución de los vértices sea también polinómica,

$$c_i(v) = \sum_{j=0}^{n} (c_i)_j B_j^n(v),$$

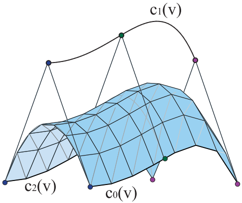
Generación de superficies

- Por coherencia, parece razonable postular que la evolución de los vértices sea también polinómica,

$$c_i(v) = \sum_{j=0}^{n} (c_i)_j B_j^n(v),$$

- La superficie resultante será polinómica de grado *m* en *u* y *n* en *v* (**bigrado** (m,n)),

$$c(u,v) = \sum_{i=0}^{m} \sum_{j=0}^{n} c_{i,j} B_i^m(u) B_j^n(v), \quad u,v \in [0,1].$$



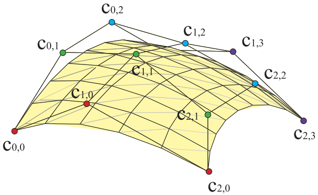
Generación de superficies

- Por coherencia, parece razonable postular que la evolución de los vértices sea también polinómica,

$$c_i(v) = \sum_{j=0}^{n} (c_i)_j B_j^n(v),$$

- La superficie resultante será polinómica de grado *m* en *u* y *n* en *v* (**bigrado** (m,n)),

$$c(u,v) = \sum_{i=0}^{m} \sum_{j=0}^{n} c_{i,j} B_i^m(u) B_j^n(v), \quad u,v \in [0,1].$$



Estructura del curso

- Curvas polinómicas.
- Curvas racionales.
- Curvas *spline*.
- Superficies de Bézier. 

- Generación de superficies.

Esquema

Superficies de Bézier

- Una superficie polinómica de Bézier de bigrado (m, n) se define por una **malla de control**, $\begin{pmatrix} c_{0,0} & \cdots & c_{0,n} \\ \vdots & \ddots & \vdots \\ c_{m,0} & \cdots & c_{m,n} \end{pmatrix}$.

$$c(u, v) = \sum_{i=0}^{m} \sum_{j=0}^{n} c_{i,j} B_i^m(u) B_j^n(v), \quad u, v \in [0, 1].$$

- Nótese que la base de funciones es partición de la unidad,

$$\sum_{i=0}^{m} \sum_{j=0}^{n} B_i^m(u) B_j^n(v) \equiv 1.$$

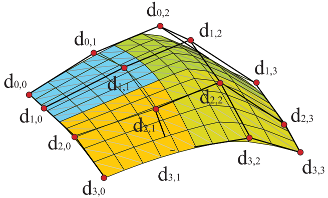
- Esto permite que las superficies hereden las buenas propiedades de las curvas.

Superficies B-*spline*

- Sustituir los polinomios de Bernstein por las funciones B-spline.
- Una superficie B-spline de bigrado (m, n) y $M \times N$ tramos precisa dos listas de nudos, $\{u_0, \ldots, u_{2m+M-2}\}$, $\{v_0, \ldots, v_{2n+N-2}\}$ y una malla B-spline formada por $\{d_{0,0}, \ldots, d_{m+M-1,n+N-1}\}$.
- Está parametrizada en $[u_{m-1}, u_{m+M-1}] \times [v_{n-1}, v_{n+N-1}]$,

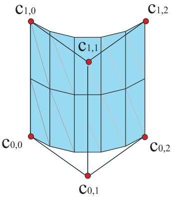
$$c(u,v) = \sum_{i=0}^{m+M-1} \sum_{j=0}^{n+N-1} d_{i,j} N_i^m(u) N_j^n(v),$$

- Las listas suelen comenzar y acabar con m y n nudos repetidos.
- El algoritmo de inserción, de De Boor... hay que aplicarlos a la malla en dos pasadas: una para las filas y otra para las columnas.



- Si la superficie es racional, es preciso dar además la **matriz de pesos** de los vértices, $\begin{pmatrix} w_{0,0} & \cdots & w_{0,n} \\ \vdots & \ddots & \vdots \\ w_{m,0} & \cdots & w_{m,n} \end{pmatrix}$,

$$c(u,v) = \frac{\sum_{i=0}^{m}\sum_{j=0}^{n} w_{i,j} c_{i,j} B_i^m(u) B_j^n(v)}{\sum_{i=0}^{m}\sum_{j=0}^{n} w_{i,j} B_i^m(u) B_j^n(v)}, \quad u,v \in [0,1].$$

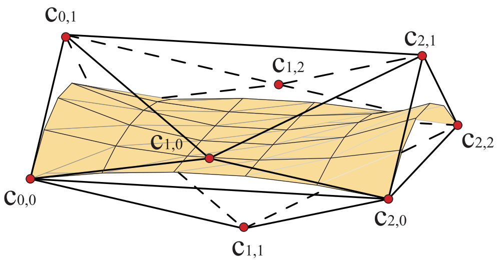


Superficies racionales B-*spline*

- Una superficie NURBS de bigrado (m, n) y $M \times N$ tramos precisa dos listas de nudos, $\{u_0, \ldots, u_{2m+M-2}\}$, $\{v_0, \ldots, v_{2n+N-2}\}$ y una malla B-spline formada por $\{d_{0,0}, \ldots, d_{m+M-1,n+N-1}\}$ y la **matriz de pesos** de los vértices, $\begin{pmatrix} w_{0,0} & \cdots & w_{0,n+N-1} \\ \vdots & \ddots & \vdots \\ w_{m+M-1,0} & \cdots & w_{m+M-1,n+N-1} \end{pmatrix}$.
- Está parametrizada en $[u_{m-1}, u_{m+M-1}] \times [v_{n-1}, v_{n+N-1}]$,

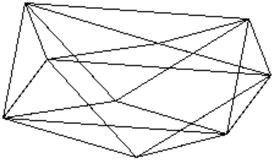
$$c(u,v) = \frac{\displaystyle\sum_{i=0}^{m+M-1} \sum_{j=0}^{n+N-1} w_{i,j} d_{i,j} N_i^m(u) N_j^n(v),}{\displaystyle\sum_{i=0}^{m+M-1} \sum_{j=0}^{n+N-1} w_{i,j} N_i^m(u) N_j^n(v).}$$

- **Invariancia afín** (proyectiva para superficies racionales)**:** La imagen de una superficie de malla $\{c_{0,0}, \ldots, c_{m,n}\}$ bajo una aplicación afín f es una superficie de malla $\{f(c_{0,0}), \ldots, f(c_{m,n})\}$.
- **Envolvente convexa:** La superficie sigue estando comprendida en el menor poliedro convexo que contenga a todos los vértices de la malla.



Propiedades de las superficies de Bézier

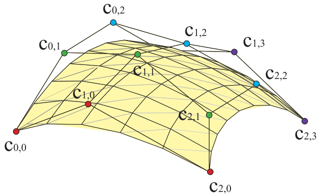
- **Invariancia afín** (proyectiva para superficies racionales)**:** La imagen de una superficie de malla $\{c_{0,0}, \ldots, c_{m,n}\}$ bajo una aplicación afín f es una superficie de malla $\{f(c_{0,0}), \ldots, f(c_{m,n})\}$.
- **Envolvente convexa:** La superficie sigue estando comprendida en el menor poliedro convexo que contenga a todos los vértices de la malla.



Borde de la superficie

- **Extremos:** La superficie sólo pasa por las esquinas de la malla,

$$c_{0,0} = c(0,0), \quad c_{m,0} = c(1,0), \quad c_{0,n} = c(0,1), \quad c_{m,n} = c(1,1).$$



- **Bordes:** Las filas y columnas del borde la malla describen el borde de la superficie.
- La curva $u = 0$,

$$c(0,v) = \sum_{i=0}^{m} \sum_{j=0}^{n} c_{i,j} B_i^m(0) B_j^n(v) = \sum_{j=0}^{n} c_{0,j} B_j^n(v), \quad v \in [0,1],$$

tiene por polígono $\{c_{0,0}, \ldots, c_{0,n}\}$, primera fila de la malla.

Borde de la superficie

- **Extremos:** La superficie sólo pasa por las esquinas de la malla,

$$c_{0,0} = c(0,0), \quad c_{m,0} = c(1,0), \quad c_{0,n} = c(0,1), \quad c_{m,n} = c(1,1).$$

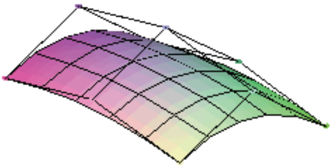
- **Bordes:** Las filas y columnas del borde la malla describen el borde de la superficie.
- La curva $u = 0$, tiene por polígono $\{c_{0,0}, \dots, c_{0,n}\}$, primera fila de la malla.
- La última fila, $\{c_{m,0}, \dots, c_{m,n}\}$, es el polígono de la curva $c(1, v)$.
- El polígono de $c(u, 0)$ es la primera columna, $\{c_{0,0}, \dots, c_{m,0}\}$.
- El polígono de $c(u, 1)$ es la última columna, $\{c_{0,n}, \dots, c_{m,n}\}$.
- Estas propiedades se mantienen en las superficies *spline* en el caso de nudos repetidos.

Borde de la superficie

- **Extremos:** La superficie sólo pasa por las esquinas de la malla,

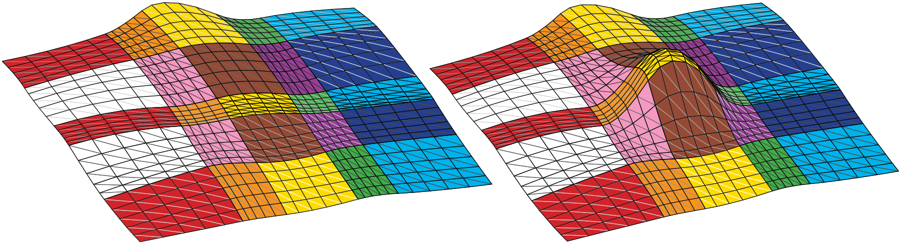
$$c_{0,0} = c(0,0), \quad c_{m,0} = c(1,0), \quad c_{0,n} = c(0,1), \quad c_{m,n} = c(1,1).$$

- **Bordes:** Las filas y columnas del borde la malla describen el borde de la superficie.



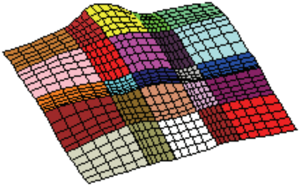
Control local

- **Control local:** Las superficies *spline* siguen teniendo esta propiedad.
- Un vértice de la malla de control afecta a lo sumo a $(m+1)\cdot(n+1)$ tramos de la superficie.
- Obviamente, un vértices próximo al borde afecta a menos tramos.



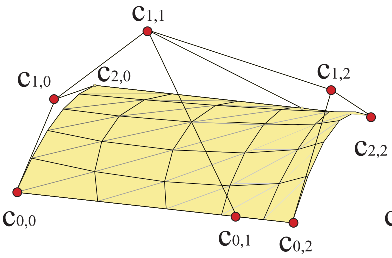
Control local

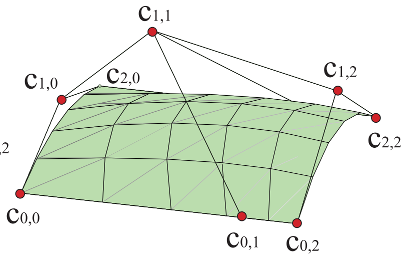
- **Control local:** Las superficies *spline* siguen teniendo esta propiedad.
- Un vértice de la malla de control afecta a lo sumo a $(m+1)\cdot(n+1)$ tramos de la superficie.
- Obviamente, un vértices próximo al borde afecta a menos tramos.



Pesos

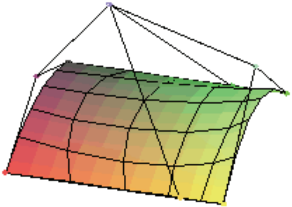
- Al aumentar un peso, la superficie se acerca al vértice correspondiente de la malla.
- Un peso afecta a lo sumo a $(m+1)\cdot(n+1)$ tramos de la superficie.





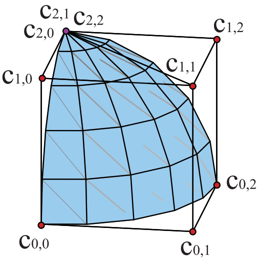
Pesos

- Al aumentar un peso, la superficie se acerca al vértice correspondiente de la malla.
- Un peso afecta a lo sumo a $(m+1)\cdot(n+1)$ tramos de la superficie.



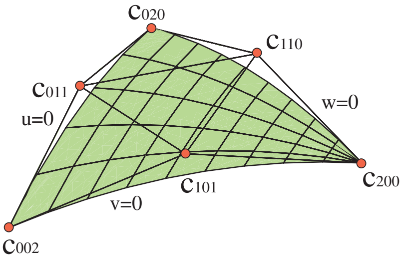
Problemas de las superficies de Bézier

- Esta representación de las superficies se conoce como representación **producto tensorial**, porque la base de funciones es el producto de las bases de polinomios en las variables u, v.
- Es cómoda para superficies abiertas.
- Presenta problemas en otras topologías (cilindros, esferas), ya que suelen requerir mallas degeneradas (vértices repetidos).



Problemas de las superficies de Bézier

- Esta representación de las superficies se conoce como representación **producto tensorial**, porque la base de funciones es el producto de las bases de polinomios en las variables u, v.
- Es cómoda para superficies abiertas.
- Presenta problemas en otras topologías (cilindros, esferas), ya que suelen requerir mallas degeneradas (vértices repetidos).
- Existen otras representaciones más versátiles: **triángulos de Bézier**.



- No presenta problemas aplicarlo a una superficie de malla $\{c_{0,0}, \ldots, c_{m,n}\}$.
- Se aplica dos veces:
- La primera a las $n+1$ columnas de la malla,

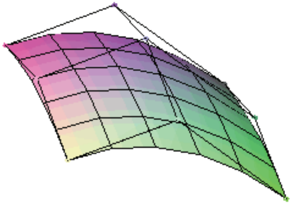
$$\begin{pmatrix} c_{0,0} & \cdots & c_{0,n} \\ \vdots & \ddots & \vdots \\ c_{m,0} & \cdots & c_{m,n} \end{pmatrix} \longrightarrow \{c_0(u), \ldots, c_n(u)\} \longrightarrow c(u,v),$$

y la segunda al polígono resultante. O a la inversa.

Polarización

- Del mismo modo, se define la polarización $c[u_1, \ldots, u_m; v_1, .., v_m]$ resultado de interpolar con un valor distinto en cada paso del algoritmo.
- Recupera los vértices, $c_{i,j} = c[0^{<m-i>}, 1^{<i>}; 0^{<n-j>}, 1^{<j>}]$.
- **Restricción:** si queremos restringir la superficie a los intervalos $u \in [a, b]$, $v \in [c, d]$, los vértices de la nueva malla son

$$\tilde{c}_{i,j} = c[a^{<m-i>}, b^{<i>}; c^{<n-j>}, d^{<j>}].$$



Elevación del grado

- Si tenemos una malla de bigrado (m, n), $\{c_{0,0}, \ldots, c_{m,n}\}$ la expresamos como superficie de bigrado $(m+1, n)$ aplicando el algoritmo de elevación a las $n+1$ columnas de la malla.

$$\begin{aligned} c(u,v) &= \sum_{i=0}^{m}\sum_{j=0}^{n} c_{i,j}B_i^m(u)B_j^n(v) = \sum_{i=0}^{m+1}\sum_{j=0}^{n} c_{i,j}^{1,0}B_i^{m+1}(u)B_j^n(v), \\ c_{i,j}^{1,0} &= \left(1-\frac{i}{m+1}\right)c_{i,j} + \frac{i}{m+1}c_{i-1,j}, \end{aligned}$$

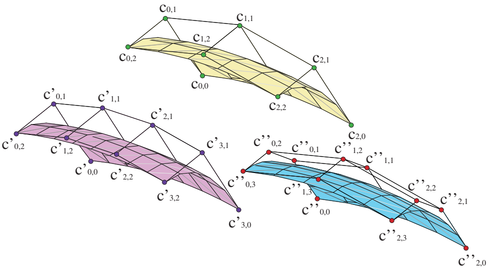
- Si queremos elevar el bigrado a $(m, n+1)$, aplicamos el algoritmo a las $m+1$ filas de la malla de control,

$$\begin{aligned} c(u,v) &= \sum_{i=0}^{m}\sum_{j=0}^{n} c_{i,j}B_i^m(u)B_j^n(v) = \sum_{i=0}^{m}\sum_{j=0}^{n+1} c_{i,j}^{0,1}B_i^{m}(u)B_j^{n+1}(v), \\ c_{i,j}^{0,1} &= \left(1-\frac{j}{n+1}\right)c_{i,j} + \frac{j}{n+1}c_{i,j-1}. \end{aligned}$$

Elevación del grado

- Si tenemos una malla de bigrado (m,n), $\{c_{0,0},\dots,c_{m,n}\}$ la expresamos como superficie de bigrado $(m+1,n)$ aplicando el algoritmo de elevación a las $n+1$ columnas de la malla.

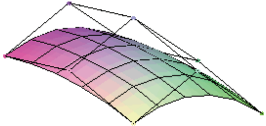
$$
\begin{aligned}
c(u,v) &= \sum_{i=0}^{m}\sum_{j=0}^{n} c_{i,j}B_i^m(u)B_j^n(v) = \sum_{i=0}^{m+1}\sum_{j=0}^{n} c_{i,j}^{1,0}B_i^{m+1}(u)B_j^n(v),\\
c_{i,j}^{1,0} &= \left(1-\frac{i}{m+1}\right)c_{i,j} + \frac{i}{m+1}c_{i-1,j},
\end{aligned}
$$



Elevación del grado

- Si tenemos una malla de bigrado (m, n), $\{c_{0,0}, \ldots, c_{m,n}\}$ la expresamos como superficie de bigrado $(m+1, n)$ aplicando el algoritmo de elevación a las $n+1$ columnas de la malla.

$$\begin{aligned} c(u,v) &= \sum_{i=0}^{m}\sum_{j=0}^{n} c_{i,j} B_i^m(u) B_j^n(v) = \sum_{i=0}^{m+1}\sum_{j=0}^{n} c_{i,j}^{1,0} B_i^{m+1}(u) B_j^n(v), \\ c_{i,j}^{1,0} &= \left(1 - \frac{i}{m+1}\right) c_{i,j} + \frac{i}{m+1} c_{i-1,j}, \end{aligned}$$



Derivadas parciales

- La derivada con respecto a u,

$$\frac{\partial c(u,v)}{\partial u} = m \sum_{j=0}^{n} \sum_{i=0}^{m-1} \Delta^{1,0} c_{i,j} B_i^{m-1}(u) B_j^n(v),$$

$$\Delta^{1,0} c_{i,j} = c_{i+1,j} - c_{i,j}, \qquad \Delta^{0,1} c_{i,j} = c_{i,j+1} - c_{i,j}.$$

- La derivada parcial respecto a v es

$$\frac{\partial c(u,v)}{\partial v} = n \sum_{i=0}^{m} \sum_{j=0}^{n-1} \Delta^{0,1} c_{i,j} B_i^{m}(u) B_j^{n-1}(v),$$

- La generalización a derivadas superiores es inmediata,

$$\frac{\partial^{r+s} c(u,v)}{\partial u^r \partial v^s} = \frac{m!\,n!}{(m-r)!(n-s)!} \sum_{i=0}^{m-r} \sum_{j=0}^{n-s} \Delta^{r,s} c_{i,j} B_i^{m-r}(u) B_j^{n-s}(v).$$

Derivadas en los bordes

- Consideraremos el borde en $u = 0$.
- La novedad es la aparición de las derivadas transversales al borde, las parciales con respecto a u,

$$\left.\frac{\partial^r c(u,v)}{\partial u^r}\right|_{u=0} = \frac{m!}{(m-r)!}\sum_{j=0}^{n} \Delta^{r,0} c_{0,j} B_j^n(v).$$

- Y en $u = 1$,

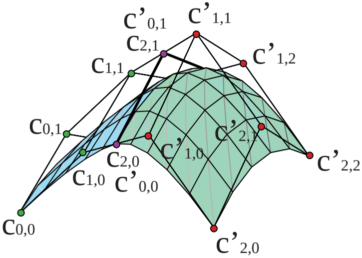
$$\left.\frac{\partial^r c(u,v)}{\partial u^r}\right|_{u=1} = \frac{m!}{(m-r)!}\sum_{j=0}^{n} \Delta^{r,0} c_{m-r,j} B_j^n(v).$$

- Permite interpretar las hileras interiores de vértices de la malla: la primera define el borde, la segunda la tangente y así sucesivamente.

Unión de superficies

- Dos superficies $c(u,v)$, $\tilde{c}(u,v)$ en $[u_0,u_1]\times[v_0,v_1]$, $[u_1,u_2]\times[v_0,v_1]$, con mallas $\{c_{0,0},\dots,c_{m,n}\}$ y $\{\tilde{c}_{0,0},\dots,\tilde{c}_{m,n}\}$.
- Queremos unirlas a lo largo del borde $u=u_1$.
- La continuidad exige que $c(u_1,v)\equiv\tilde{c}(u_1,v)$.
- Es decir, sus polígonos de control son los mismos: la última fila de la malla de la primera superficie y la primera fila de la segunda,

$$c(u_1,v)=\tilde{c}(u_1,v)\Rightarrow c_{m,j}=\tilde{c}_{0,j},\quad j=0,\dots,n.$$

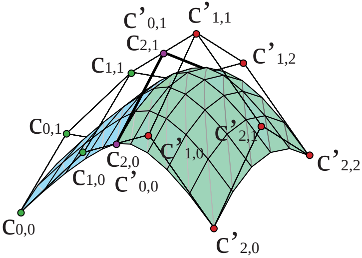


Unión de superficies

- Dos superficies $c(u, v)$, $\tilde{c}(u, v)$ en $[u_0, u_1] \times [v_0, v_1]$, $[u_1, u_2] \times [v_0, v_1]$, con mallas $\{c_{0,0}, \ldots, c_{m,n}\}$ y $\{\tilde{c}_{0,0}, \ldots, \tilde{c}_{m,n}\}$.
- Queremos unirlas a lo largo del borde $u = u_1$.
- La superficie compuesta es C^1 si

$$\left.\frac{\partial c(u,v)}{\partial u}\right|_{u=u_1} = \left.\frac{\partial \tilde{c}(u,v)}{\partial u}\right|_{u=u_1} \Rightarrow \frac{\Delta^{1,0} c_{m-1,j}}{\Delta u_0} = \frac{\Delta^{1,0}\tilde{c}_{0,j}}{\Delta u_1}, \quad j = 0, \ldots, n.$$

- Ser C^1 afecta a la franja de vértices de las dos últimas filas de la primera superficie y las dos primeras filas de la segunda.

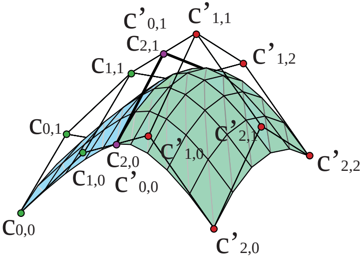


Unión de superficies

- Dos superficies $c(u,v)$, $\tilde{c}(u,v)$ en $[u_0,u_1]\times[v_0,v_1]$, $[u_1,u_2]\times[v_0,v_1]$, con mallas $\{c_{0,0},\ldots,c_{m,n}\}$ y $\{\tilde{c}_{0,0},\ldots,\tilde{c}_{m,n}\}$.
- Queremos unirlas a lo largo del borde $u=u_1$.
- La superficie compuesta es C^r si

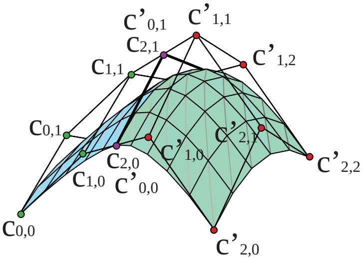
$$\frac{\Delta^{r,0}c_{m-r,j}}{(\Delta u_0)^r}=\frac{\Delta^{r,0}\tilde{c}_{0,j}}{(\Delta u_1)^r},\quad j=0,\ldots,n.$$

- Ser C^r afecta a la franja de vértices de las $r+1$ últimas filas de la primera superficie y las $r+1$ primeras filas de la segunda.



Unión de superficies

- Dos superficies $c(u,v)$, $\tilde{c}(u,v)$ en $[u_0,u_1]\times[v_0,v_1]$, $[u_1,u_2]\times[v_0,v_1]$, con mallas $\{c_{0,0},\ldots,c_{m,n}\}$ y $\{\tilde{c}_{0,0},\ldots,\tilde{c}_{m,n}\}$.
- Queremos unirlas a lo largo del borde $u=u_1$.
- El problema de construir superficies compuestas se complica si queremos añadir superficies de Bézier definidas en otros bordes, ya que un vértice está determinado por varias condiciones.
- Las superficies B-spline solucionan estos problemas salvo que repitamos nudos.



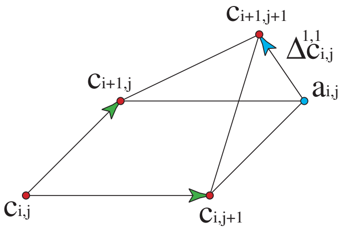
Twists

- Las derivadas segundas cruzadas se suelen llamar *twists*,

$$\frac{\partial^2 c(u,v)}{\partial u \partial v} = mn \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \Delta^{1,1} c_{i,j} B_i^{m-1}(u) B_j^{n-1}(v),$$

$$\Delta^{1,1} c_{i,j} = c_{i+1,j+1} - a_{i,j}, \quad a_{i,j} = c_{i,j} + \overrightarrow{c_{i,j} c_{i+1,j}} + \overrightarrow{c_{i,j} c_{i,j+1}}.$$

- El vector $\Delta^{1,1} c_{i,j}$ representa la separación del vértice $c_{i+1,j+1}$ respecto del paralelogramo que determinan $c_{i,j}, c_{i+1,j}, c_{i,j+1}, a_{i,j}$.



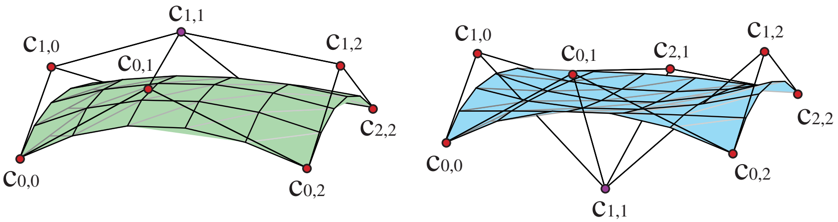
Twists

- Las derivadas segundas cruzadas se suelen llamar *twists*,

$$\frac{\partial^2 c(u,v)}{\partial u \partial v} = mn \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \Delta^{1,1} c_{i,j} B_i^{m-1}(u) B_j^{n-1}(v),$$

$$\Delta^{1,1} c_{i,j} = c_{i+1,j+1} - a_{i,j}, \quad a_{i,j} = c_{i,j} + \overrightarrow{c_{i,j} c_{i+1,j}} + \overrightarrow{c_{i,j} c_{i,j+1}}.$$

- En la esquina $c_{0,0}$, el *twist* mide la separación del vértice $c_{1,1}$ del plano tangente en $c_{0,0}$.



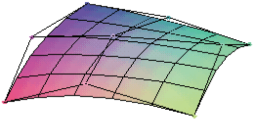
Twists

- Las derivadas segundas cruzadas se suelen llamar *twists*,

$$\frac{\partial^2 c(u,v)}{\partial u \partial v} = mn \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \Delta^{1,1} c_{i,j} B_i^{m-1}(u) B_j^{n-1}(v),$$

$$\Delta^{1,1} c_{i,j} = c_{i+1,j+1} - a_{i,j}, \quad a_{i,j} = c_{i,j} + \overrightarrow{c_{i,j}c_{i+1,j}} + \overrightarrow{c_{i,j}c_{i,j+1}}.$$

- En una superficie bicúbica la malla posee 16 vértices (12 en el borde). Los 4 vértices interiores los fijan los *twists*.



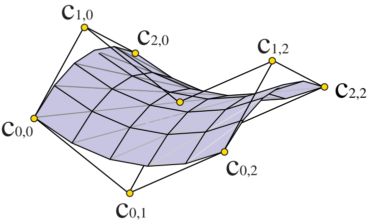
Twists

- Las derivadas segundas cruzadas se suelen llamar *twists*,

$$\frac{\partial^2 c(u,v)}{\partial u \partial v} = mn \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \Delta^{1,1} c_{i,j} B_i^{m-1}(u) B_j^{n-1}(v),$$

$$\Delta^{1,1} c_{i,j} = c_{i+1,j+1} - a_{i,j}, \quad a_{i,j} = c_{i,j} + \overrightarrow{c_{i,j}c_{i+1,j}} + \overrightarrow{c_{i,j}c_{i,j+1}}.$$

- Las superficies de *twists* nulos se llaman **traslacionales**.



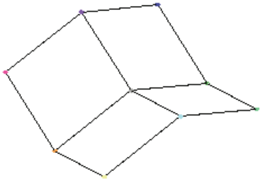
Twists

- Las derivadas segundas cruzadas se suelen llamar *twists*,

$$\frac{\partial^2 c(u,v)}{\partial u \partial v} = mn \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \Delta^{1,1} c_{i,j} B_i^{m-1}(u) B_j^{n-1}(v),$$

$$\Delta^{1,1} c_{i,j} = c_{i+1,j+1} - a_{i,j}, \quad a_{i,j} = c_{i,j} + \overrightarrow{c_{i,j} c_{i+1,j}} + \overrightarrow{c_{i,j} c_{i,j+1}}.$$

- Las superficies de *twists* nulos se llaman **traslacionales**.



Interpolación polinómica

- Tenemos una nube de $(m+1)\cdot(n+1)$ datos $\{a_{0,0},\dots,a_{m,n}\}$ por los cuales queremos interpolar una superficie $c(u,v)$ tal que

$$a_{i,j} = c(u_i, v_j), \qquad i = 0,\dots,m \qquad j = 0,\dots,n.$$

- En vez de atacar el problema como un sistema de $(m+1)\cdot(n+1)$ ecuaciones, lo escribimos como $B_U C B_V^t = A$.

$$A = \begin{pmatrix} a_{0,0} & \cdots & a_{0,n} \\ \vdots & \ddots & \vdots \\ a_{m,0} & \cdots & a_{m,n} \end{pmatrix}, \quad B_U = \begin{pmatrix} B_0^m(u_0) & \cdots & B_m^m(u_0) \\ \vdots & \ddots & \vdots \\ B_0^m(u_m) & \cdots & B_m^m(u_m) \end{pmatrix}$$

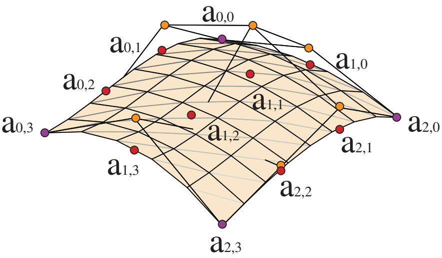
$$C = \begin{pmatrix} c_{0,0} & \cdots & c_{0,n} \\ \vdots & \ddots & \vdots \\ c_{m,0} & \cdots & c_{m,n} \end{pmatrix}, \quad B_V = \begin{pmatrix} B_0^n(v_0) & \cdots & B_n^n(v_0) \\ \vdots & \ddots & \vdots \\ B_0^n(v_n) & \cdots & B_n^n(v_n) \end{pmatrix}.$$

Interpolación polinómica

- Tenemos una nube de $(m+1)\cdot(n+1)$ datos $\{a_{0,0},\ldots,a_{m,n}\}$ por los cuales queremos interpolar una superficie $c(u,v)$ tal que

$$a_{i,j} = c(u_i, v_j), \qquad i = 0,\ldots,m \qquad j = 0,\ldots,n.$$

- En vez de atacar el problema como un sistema de $(m+1)\cdot(n+1)$ ecuaciones, lo escribimos como $B_U C B_V^t = A$.
- Resolvemos los sistemas $\tilde{C}B_V^t = A$ y con la solución obtenida, $\tilde{C}$, resolvemos $B_U C = \tilde{C}$.
- Hemos reducido el problema a dos sistemas de $(m+1)$ y $(n+1)$ ecuaciones.



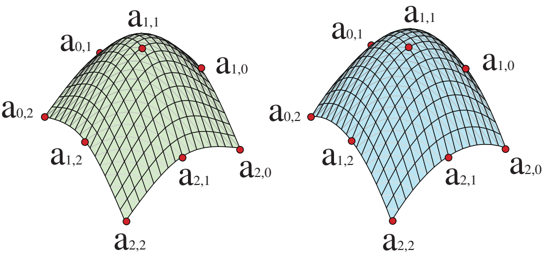
Interpolación polinómica

- Tenemos una nube de $(m+1)\cdot(n+1)$ datos $\{a_{0,0},\ldots,a_{m,n}\}$ por los cuales queremos interpolar una superficie $c(u,v)$ tal que

$$a_{i,j} = c(u_i, v_j), \qquad i = 0,\ldots,m \qquad j = 0,\ldots,n.$$

- En vez de atacar el problema como un sistema de $(m+1)\cdot(n+1)$ ecuaciones, lo escribimos como $B_U C B_V^t = A$.

- Más común es interpolar mediante superficies B-*spline* bicúbicas.
- El planteamiento es el mismo, sustituyendo los polinomios de Bernstein por funciones B-*spline*.
- Las listas de nudos son $\{u_0, \dots, u_M\}$, $\{v_0, \dots, v_N\}$, con los nudos inicial y final repetidos tres veces.
- Hace falta imponer condiciones de tangencia en los extremos.



Interpolación bicúbica *spline*

- Más común es interpolar mediante superficies B-*spline* bicúbicas.
- El planteamiento es el mismo, sustituyendo los polinomios de Bernstein por funciones B-*spline*.
- Las listas de nudos son $\{u_0, \ldots, u_M\}$, $\{v_0, \ldots, v_N\}$, con los nudos inicial y final repetidos tres veces.
- Hace falta imponer condiciones de tangencia en los extremos.

Datos sin estructura

- En general los datos no están organizados en malla rectangular.
- Una superficie de bigrado (m, n) que aproxime un conjunto de datos, $\{a_0, \ldots, a_M\}$, tales que $a_i = c(u_i, v_i)$.
- Sistema $BC = A$, $(m+1)\cdot(n+1)$ incógnitas, $M+1$ ecuaciones,

$$\begin{pmatrix} B_0^m(u_0)B_0^n(v_0) & \cdots & B_m^m(u_0)B_n^n(v_0) \\ \vdots & \ddots & \vdots \\ B_0^m(u_M)B_0^n(v_M) & \cdots & B_m^m(u_M)B_n^n(v_M) \end{pmatrix} \begin{pmatrix} c_{0,0} \\ \vdots \\ c_{m,n} \end{pmatrix} = \begin{pmatrix} a_0 \\ \vdots \\ a_M \end{pmatrix}.$$

- Se resuelve por mínimos cuadrados: $B^t BC = B^t A$.
- Para un número alto de datos el sistema está mal condicionado.