

Interpolation

Vincent Nozick



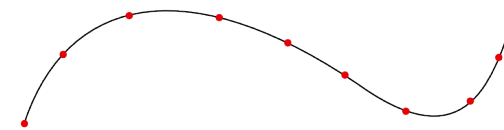
Problématique

L'interpolation est une opération mathématique permettant de construire une courbe à partir de la donnée d'un nombre fini de points.

Problématique



Problématique



Polynômes de Lagrange

Introduction :

La méthode de Lagrange permet de calculer un polynôme de degré $n - 1$ passant exactement par n points.

Polynômes de Lagrange

Méthode :

Le polynôme $P(x)$ passant par les points $\{(x_i, y_i)\}_{i=1..n}$ est :

$$P(x) = y_1 L_{n-1,1}(x) + y_2 L_{n-1,2}(x) + \dots + y_n L_{n-1,n}(x)$$

$L_{n-1,i}(x)$:

- degré du polynôme = $n - 1$

- s'annule en x_j ($j \neq i$) et qui vaut 1 en x_i

Polynômes de Lagrange

Méthode :

Soit un ensemble de points $\{(x_i, y_i)\}_{i=1..n}$

On définit le polynôme $L_{k,i}(x)$

- k : degré du polynôme
- i : qui s'annule en x_j ($j \neq i$) et qui vaut 1 en x_i

$$L_{n,i}(x) = \frac{(x - x_1)(x - x_2) \cdots (x - x_{i-1})(x - x_{i+1}) \cdots (x - x_n)}{(x_i - x_1)(x_i - x_2) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_n)}$$

Polynômes de Lagrange

Exemple :

On cherche le polynôme $P(x)$ passant par les 3 points (x_1, y_1) , (x_2, y_2) et (x_3, y_3)

$$P(x) = y_1 \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)} + y_2 \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)} + y_3 \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)}$$

Polynômes de Lagrange

Exemple :

On cherche le polynôme $P(x)$ passant par les 3 points (x_1, y_1) , (x_2, y_2) et (x_3, y_3)

$$P(x) = y_1 \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)} + y_2 \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)} + y_3 \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)}$$

$$P(x_1) = y_1 \underbrace{\frac{(x_1-x_2)(x_1-x_3)}{(x_1-x_2)(x_1-x_3)}}_1 + y_2 \underbrace{\frac{(x_1-x_1)(x_1-x_3)}{(x_2-x_1)(x_2-x_3)}}_0 + y_3 \underbrace{\frac{(x_1-x_1)(x_1-x_2)}{(x_3-x_1)(x_3-x_2)}}_0$$

$$P(x_1) = y_1$$

Polynômes de Lagrange

Exemple :

On cherche le polynôme $P(x)$ passant par les 3 points (x_1, y_1) , (x_2, y_2) et (x_3, y_3)

$$P(x) = y_1 \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)} + y_2 \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)} + y_3 \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)}$$

$$P(x_2) = y_1 \underbrace{\frac{(x_2-x_2)(x_2-x_3)}{(x_1-x_2)(x_1-x_3)}}_0 + y_2 \underbrace{\frac{(x_2-x_1)(x_2-x_3)}{(x_2-x_1)(x_2-x_3)}}_1 + y_3 \underbrace{\frac{(x_2-x_1)(x_2-x_2)}{(x_3-x_1)(x_3-x_2)}}_0$$

$$P(x_2) = y_2$$

Polynômes de Lagrange

Exemple :

On cherche le polynôme $P(x)$ passant par les 3 points (x_1, y_1) , (x_2, y_2) et (x_3, y_3)

$$P(x) = y_1 \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)} + y_2 \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)} + y_3 \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)}$$

$$P(x_3) = y_1 \underbrace{\frac{(x_3-x_2)(x_3-x_3)}{(x_1-x_2)(x_1-x_3)}}_0 + y_2 \underbrace{\frac{(x_3-x_1)(x_3-x_3)}{(x_2-x_1)(x_2-x_3)}}_0 + y_3 \underbrace{\frac{(x_3-x_1)(x_3-x_2)}{(x_3-x_1)(x_3-x_2)}}_1$$

$$P(x_3) = y_3$$

Polynômes de Lagrange

Exemple :

Le polynôme $P(x)$ de degré $n-1=2$ passant par les 3 points (x_1, y_1) , (x_2, y_2) et (x_3, y_3) est bien :

$$P(x) = y_1 L_{2,1}(x) + y_2 L_{2,2}(x) + y_3 L_{2,3}(x)$$

Interpolation polynômiale

Introduction :

On veut trouver le polynôme P de degré n passant par $n + 1$ points (x_i, y_i) (tous les points sont distincts).

Le polynôme est de la forme :

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_2 x^2 + a_1 x + a_0$$

et vérifie :

$$P(x_i) = y_i \quad \forall i \in 1 \dots n + 1$$

Interpolation polynômiale

Remarque :

L'évaluation d'un polynôme $P(x)$ en x_0 peut s'exprimer comme un produit scalaire :

$$P(x_0) = \begin{pmatrix} x_0^n & \dots & x_0^3 & x_0^2 & x_0 & 1 \end{pmatrix} \begin{pmatrix} a_n \\ \vdots \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{pmatrix}$$

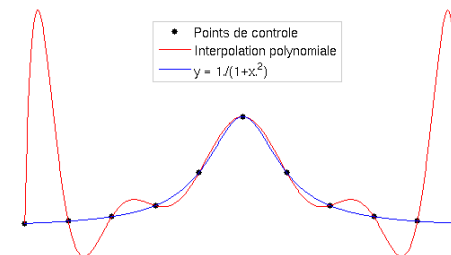
Interpolation polynômiale

Pour $n + 1$ points, on obtient donc le système suivant :

$$\begin{bmatrix} x_0^n & \dots & x_0^3 & x_0^2 & x_0 & 1 \\ x_1^n & \dots & x_1^3 & x_1^2 & x_1 & 1 \\ x_2^n & \dots & x_2^3 & x_2^2 & x_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^n & \dots & x_n^3 & x_n^2 & x_n & 1 \end{bmatrix} \begin{pmatrix} a_n \\ \vdots \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

Interpolation polynômiale

Remarques :



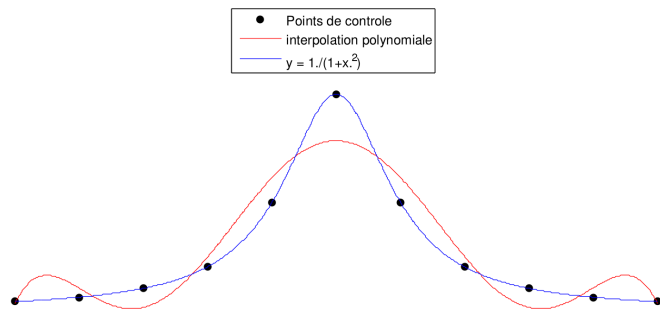
En général, les méthodes polynômiales ne sont pas très stables entre les points de contrôle (phénomène de Runge).

Interpolation polynômiale

Stabilité :

- On diminue le degré du polynôme \Rightarrow moindres carrés
le polynôme ne passera pas nécessairement par tous les points de contrôle.
- on découpe l'intervalle d'interpolation en petits intervalles \Rightarrow pleins de polynômes de petit degré.

Interpolation polynômiale



\rightarrow 11 points de contrôle
 \rightarrow polynôme de degré 7

Interpolation polynômiale

Moindres carrés :

On choisit délibérément le degré m du polynôme inférieur au degré n prévu.

\rightarrow système surdéterminé (moindres carrés).

$$\begin{bmatrix} x_0^m & x_0^{m-1} & x_0^{m-2} & \cdots & x_0 & 1 \\ x_1^m & x_1^{m-1} & x_1^{m-2} & \cdots & x_1 & 1 \\ x_2^m & x_2^{m-1} & x_2^{m-2} & \cdots & x_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{n-1}^m & x_{n-1}^{m-1} & x_{n-1}^{m-2} & \cdots & x_{n-1} & 1 \\ x_n^m & x_n^{m-1} & x_n^{m-2} & \cdots & x_n & 1 \end{bmatrix} \begin{pmatrix} a_m \\ a_{m-1} \\ \vdots \\ a_0 \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{n-1} \\ y_n \end{pmatrix}$$

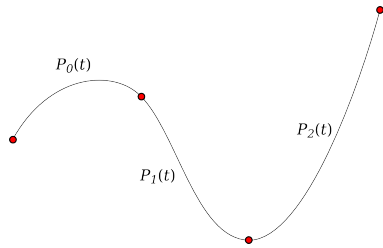
Cubic Spline

Introduction :

Pour éviter le phénomène de Runge, on peut choisir de découper l'intervalle d'interpolation en intervalles plus petits sur lesquels on utilise un polynôme de petit degré.

\rightarrow Les splines cubiques sont un ensemble de polynômes de degré 3 passant par un ensemble de n points de contrôle.

Cubic Spline



En 1D :

Pour une spline 1D passant par $n+1$ points de contrôle (x_0, x_1, \dots, x_n) , la i ème partie de la spline est représenté par le polynôme $P_i(t)$:

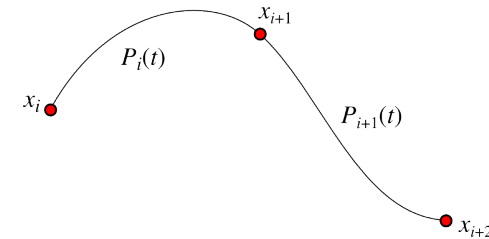
$$P_i(t) = a_i + b_it + c_it^2 + d_it^3$$

avec $t \in [0, 1]$ et $i = 0, \dots, n + 1$.

Cubic Spline

Chaque bout de spline doit être cohérente avec la suivante :

- $P_i(1) = P_{i+1}(0) = x_{i+1}$ la fin de $P_i(t)$ = début de $P_{i+1}(t)$
- $P'_i(1) = P'_{i+1}(0)$ idem pour la dérivée
- $P''_i(1) = P''_{i+1}(0)$ idem pour la dérivée seconde



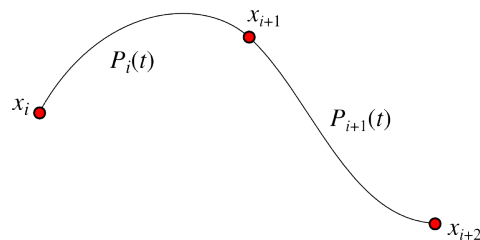
Cubic Spline

$$P_i(t) = a_i + b_it + c_it^2 + d_it^3$$

Points de contrôle :

$$P_i(0) = x_i = a_i$$

$$P_i(1) = x_{i+1} = a_i + b_i + c_i + d_i$$



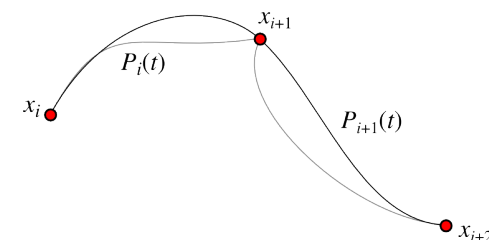
Cubic Spline

$$P_i(t) = a_i + b_it + c_it^2 + d_it^3$$

Dérivée première :

$$P'_i(0) \equiv D_i = b_i$$

$$P'_i(1) \equiv D_{i+1} = b_i + 2c_i + 3d_i$$



Radial Basis Function Interpolation

Introduction :

Cette méthode permet d'interpoler une fonction :

$$u = f(x_1, x_2, \dots, x_n)$$

Applications :

- interpoler une couleur dans une image 2d
- interpoler une température dans un espace 3d
- interpoler une vitesse dans un espace 3d + t
- ...

Radial Basis Function Interpolation

Méthode :

entrée :

- un ensemble de k points nd $\{\mathbf{x}_i = (x_1^i, x_2^i, \dots, x_n^i)\}_{i=1\dots k}$
- leur valeur $u_i = f(\mathbf{x}_i)$

sortie : On cherche une fonction $g(\mathbf{x})$ telle que $g(\mathbf{x}_i) = u_i, \forall i$

Radial Basis Function Interpolation

On définit $g(\mathbf{x})$ telle que :

$$g(\mathbf{x}) = \sum_{i=1}^k \omega_i \cdot \phi(|\mathbf{x} - \mathbf{x}_i|)$$

avec :

- ω_i un ensemble de coefficients à déterminer.
- $\phi(|\mathbf{a} - \mathbf{b}|)$ une fonction "radiale" dont la valeur dépend de la distance entre \mathbf{a} et \mathbf{b} .

Radial Basis Function Interpolation

Fonction ϕ :

Soit $d = |\mathbf{a} - \mathbf{b}|$ la norme L_2 du vecteur $\mathbf{a} - \mathbf{b}$, $\phi(d)$ est une fonction "radiale" dont la valeur dépend uniquement de d (et éventuellement de quelques constantes).

→ Il existe une multitude de fonctions radiales telles que :

- $\phi(d) = d$
- une *multiquadric* $\phi(d) = \sqrt{1 + (\epsilon d)^2}$

ou bien (radiales décroissantes)

- une *inverse quadric* $\phi(d) = \frac{1}{1 + (\epsilon d)^2}$
- une *gaussienne* $\phi(d) = e^{-\epsilon d^2}$

Radial Basis Function Interpolation

$$g(\mathbf{x}) = \sum_{i=1}^k \omega_i \cdot \phi(|\mathbf{x} - \mathbf{x}_i|)$$

- $\phi(|\mathbf{x} - \mathbf{x}_i|)$: on peut le calculer $\forall \mathbf{x}_i \rightarrow \text{OK}$.
- ω_i : on les connaît pas, mais on sait que $g(\mathbf{x}_i) = u_i, \forall i$.

On obtient k contraintes sur les ω_i :

$$\sum_{i=1}^k \omega_i \cdot \phi(|\mathbf{x}_m - \mathbf{x}_i|) = u_m \quad \forall m \in 1 \dots k$$

Radial Basis Function Interpolation

$$\sum_{i=1}^k \omega_i \cdot \phi(|\mathbf{x}_m - \mathbf{x}_i|) = u_m \quad \forall m \in 1 \dots k$$

On peut l'écrire matriciellement sous la forme :

$$\begin{bmatrix} \phi(|\mathbf{x}_1 - \mathbf{x}_1|) & \phi(|\mathbf{x}_1 - \mathbf{x}_2|) & \cdots & \phi(|\mathbf{x}_1 - \mathbf{x}_k|) \\ \phi(|\mathbf{x}_2 - \mathbf{x}_1|) & \phi(|\mathbf{x}_2 - \mathbf{x}_2|) & \cdots & \phi(|\mathbf{x}_2 - \mathbf{x}_k|) \\ \vdots & \vdots & \vdots & \vdots \\ \phi(|\mathbf{x}_k - \mathbf{x}_1|) & \phi(|\mathbf{x}_k - \mathbf{x}_2|) & \cdots & \phi(|\mathbf{x}_k - \mathbf{x}_k|) \end{bmatrix} \begin{pmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_k \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_k \end{pmatrix}$$

\rightarrow on peut résoudre et trouver les ω_i .

Radial Basis Function Interpolation

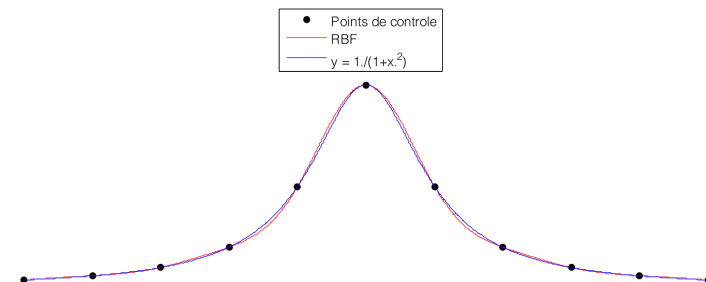
Évaluation d'un point :

Pour un point \mathbf{y} quelconque que l'on souhaite interpoler, on obtient :

$$u_{\mathbf{y}} = \sum_{i=1}^k \omega_i \cdot \phi(|\mathbf{y} - \mathbf{x}_i|)$$

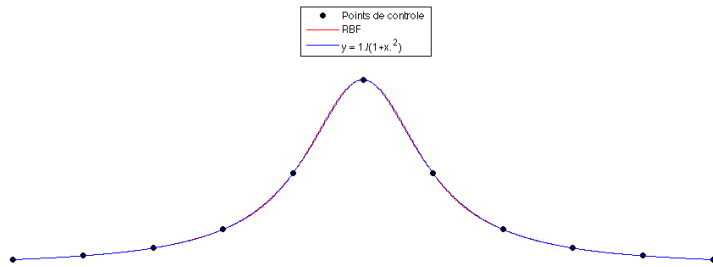
Radial Basis Function Interpolation

En 1D : $\phi(d) = e^{-d^2}$



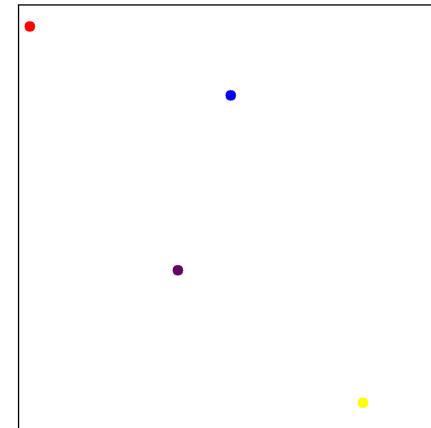
Radial Basis Function Interpolation

En 1D : $\phi(d) = \frac{1}{1 + d^2}$



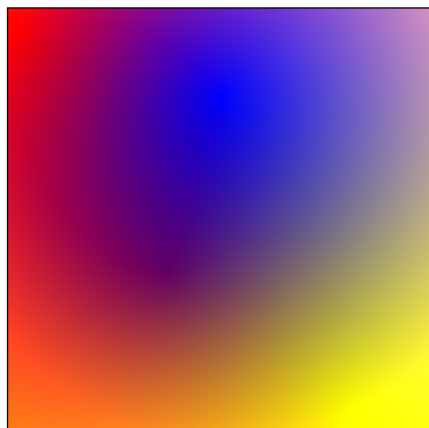
Radial Basis Function Interpolation

Exemple : interpolation 2D de couleur



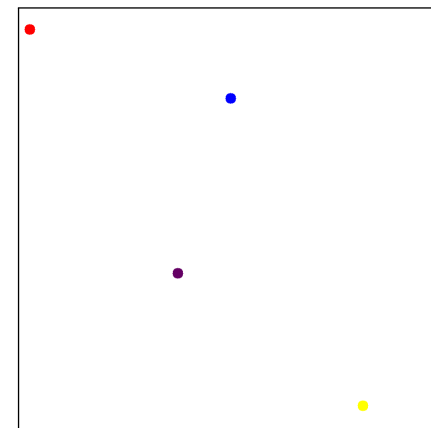
Radial Basis Function Interpolation

Exemple : interpolation 2D de couleur : $\phi(d) = d$



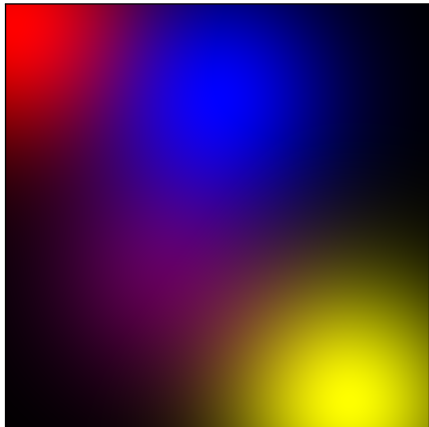
Radial Basis Function Interpolation

Exemple : interpolation 2D de couleur

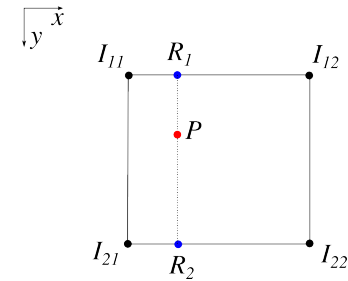


Radial Basis Function Interpolation

Exemple : interpolation 2D de couleur : $\phi(d) = e^{-\epsilon d^2}$

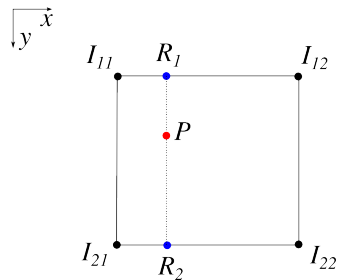


Interpolation bilinéaire



souvent utilisée pour faire de l'antialiasing

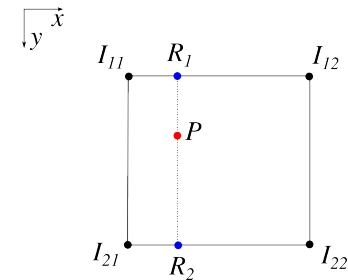
Interpolation bilinéaire



Interpolation linéaire pour R_1 et R_2 :

$$t \equiv \frac{x_2 - x_p}{x_2 - x_1} \rightarrow \begin{aligned} f(R_1) &= t.f(I_{11}) + (1 - t).f(I_{12}) \\ f(R_2) &= t.f(I_{21}) + (1 - t).f(I_{22}) \end{aligned}$$

Interpolation bilinéaire



Interpolation linéaire pour P entre R_1 et R_2 :

$$u \equiv \frac{y_2 - y_p}{y_2 - y_1} \rightarrow f(P) = u.f(R_1) + (1 - u).f(R_2)$$

Interpolation bilinéaire

Soit :

$$t \equiv \frac{x_2 - x_p}{x_2 - x_1} \quad \text{et} \quad u \equiv \frac{y_2 - y_p}{y_2 - y_1}$$

$$f(P) = (1-t)(1-u)f(I_{11}) + t(1-u)f(I_{12}) + (1-t)u.f(I_{21}) + t.u.f(I_{22})$$

Interpolation bilinéaire

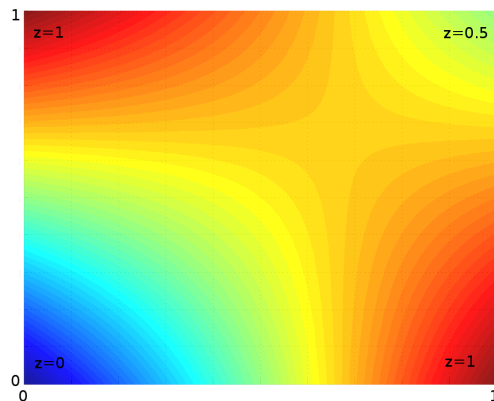
coordonnées unitaires :

$$f(x, y) = f(0, 0)(1-x)(1-y) + f(1, 0)x(1-y) + f(0, 1)(1-x)y + f(1, 1)xy$$

forme matricielle :

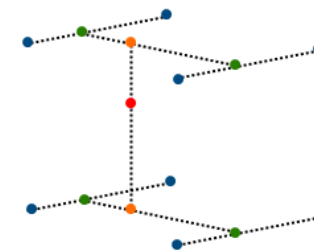
$$f(x, y) = \begin{pmatrix} 1-x & x \end{pmatrix} \begin{bmatrix} f(0, 0) & f(0, 1) \\ f(1, 0) & f(1, 1) \end{bmatrix} \begin{pmatrix} 1-y \\ y \end{pmatrix}$$

Interpolation bilinéaire



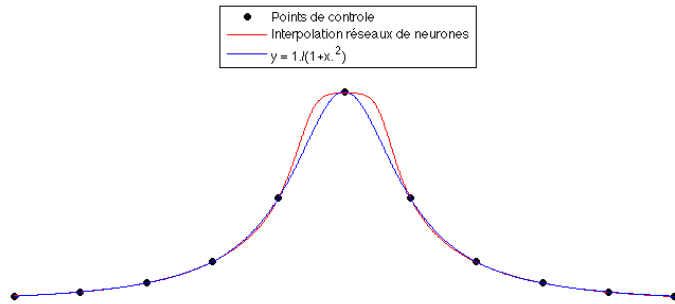
Interpolation trilineaire

Extension naturelle en 3D :



Divers

Réseaux de neurones :



→ Topologie : 1-3-3-1

→ Apprentissage : 10 000 itérations