

---

# **Théorème de superposition de Kolmogorov et son application à la décomposition de fonctions multivariées**

**Pierre-Emmanuel Leni, Yohan D. Fougerolle et Frédéric Truchetet**

*Laboratoire LE2I, UMR CNRS 5158, IUT Le Creusot,  
<http://vision.u-bourgogne.fr/le2i/>,  
<prénom>.<nom>@u-bourgogne.fr*

---

*RÉSUMÉ. Dans cet article, nous nous intéressons à la décomposition de fonctions multidimensionnelles en sommes et compositions de fonctions monodimensionnelles. Nous rappelons l'existence d'une telle décomposition par le théorème de Kolmogorov et présentons une des plus récentes techniques de construction de ces fonctions monodimensionnelles. Nous appliquons cette méthode, proposée par Sprecher, à des images en niveau de gris, et présentons plusieurs résultats de décomposition. Nous synthétisons nos observations, et présentons plusieurs pistes de recherche.*

*ABSTRACT. In this paper, we deal with the decomposition of multivariate functions into sums and compositions of univariate functions. We recall that such a decomposition exists in the Kolmogorov's superposition theorem, and present one of the most recent constructive algorithm of these univariate functions. We apply this method, introduced by Sprecher, and present several results of decomposition for grey level images. We synthesise our observations, before presenting several research perspectives.*

*MOTS-CLÉS : Théorème de Superposition de Kolmogorov, décomposition de fonctions multidimensionnelles, réseau de neurones, traitement du signal, analyse d'image*

*KEYWORDS: Kolmogorov superposition theorem, multidimensional functions decomposition, neural network, signal processing, image analysis*

---

## 1. Introduction

En 1900, Hilbert conjectura que les équations algébriques de haut degré ne peuvent pas être résolues par des fonctions construites comme substitution de fonctions continues de deux variables. En 1957, Kolmogorov contredit cette hypothèse en présentant son théorème de superposition (TSK), qui permet d'écrire toute fonction multivariée comme sommes et compositions de fonctions monovariées. Toutefois, Kolmogorov ne proposa pas de méthode pour la construction de ces fonctions monodimensionnelles et démontra seulement leur existence. L'objectif de ce travail est d'exprimer des fonctions multidimensionnelles complexes en éléments simples (des fonctions monodimensionnelles). La décomposition de signaux N-dimensionnels en signaux 1D offre une alternative à l'approche consistant à étendre des méthodes classiques de traitement de signal 1D ou 2D (décomposition en ondelettes, transformée de Fourier) à des signaux multidimensionnels. Les aspects fréquemment abordés du TSK concernent son écriture sous la forme d'un réseau de neurones à une couche cachée, comme l'illustre la figure 1, ainsi que la calculabilité et les méthodes de construction des fonctions monodimensionnelles. Parmi les travaux dédiés à l'étude du TSK, Sprecher a proposé un algorithme de reconstruction exacte des fonctions monodimensionnelles dans (Sprecher, 1996) et (Sprecher, 1997). Sprecher décrit explicitement la méthode de construction des fonctions monodimensionnelles, ce qui apporte des éléments pour la compréhension du théorème (la notion de pavage par exemple) et permet une implémentation directe.

Ce document est structuré en 4 sections. Dans la seconde, nous présentons le théorème de superposition et plusieurs notations. La section 3 est consacrée à l'algorithme de Sprecher, et dans la dernière, nous concluons et dégageons plusieurs perspectives de recherche.

Nos contributions sont l'explication synthétique de l'algorithme de Sprecher et l'application du théorème de superposition à des images en niveaux de gris, en utilisant l'algorithme de Sprecher dans le cas de fonctions bivariées.

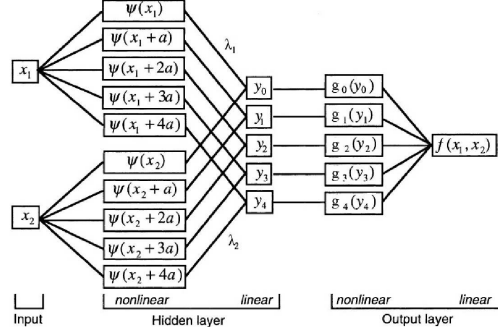
## 2. Présentation du théorème de Kolmogorov

Le TSK, reformulé et simplifié par Sprecher dans (Sprecher, 1972) s'écrit :

**Théorème 1** (Théorème de superposition de Kolmogorov). *Toute fonction continue définie sur l'hypercube identité  $[0, 1]^d$  noté  $I^d$   $f : I^d \rightarrow \mathbb{R}$  peut être écrite comme sommes et compositions de fonctions continues monodimensionnelles :*

$$\begin{cases} f(x_1, \dots, x_d) = \sum_{n=0}^{2^d} g_n(\xi(x_1 + na, \dots, x_d + na)) \\ \xi(x_1 + na, \dots, x_d + na) = \sum_{i=1}^d \lambda_i \psi(x_i + an), \end{cases} \quad [1]$$

où  $\psi$  est une fonction continue,  $\lambda_i$  et  $a$  des constantes.  $\psi$  est appelée fonction interne et  $g(\xi)$  fonction externe.



**Figure 1.** Illustration de l'analogie entre le TSK et un réseau de neurones à une couche cachée, de (Sprecher, 1997).

La fonction interne  $\psi$  associe à toute composante  $x_i$  du vecteur réel  $(x_1, \dots, x_d)$  de  $I^d$  une valeur dans  $[0, 1]$ . La fonction  $\xi$  associe à chaque vecteur  $(x_1, \dots, x_d) \in I^d$  un nombre  $y_n$  dans l'intervalle  $[0, 1]$  qui servira ensuite d'arguments aux fonctions  $g_n$  pour reconstituer la fonction  $f$  par sommation.

Ce théorème permet de voir le calcul d'une fonction multivariée comme un processus en deux étapes : la construction d'une fonction interne comme une fonction de hashage qui associe les coordonnées  $x_i, i \in \llbracket 1, d \rrbracket$  de chaque dimension à un nombre unique ; et la construction d'une fonction externe  $g$  avec les valeurs correspondant à  $f$  pour ces coordonnées. La figure 3 illustre la fonction de hashage  $\xi$  définie par Sprecher dans le cas 2D. Pour appliquer le TSK en utilisant l'algorithme de Sprecher, l'espace de définition de la fonction  $f$  doit être décomposé en superposition de pavages décalés composés d'hypercubes disjoints, comme l'illustre la figure 4 dans le cas 2D.

### 3. Algorithme de Sprecher

Sprecher a proposé un algorithme pour la détermination des fonctions internes et externes dans (Sprecher, 1996) et (Sprecher, 1997), respectivement. Comme la fonction  $\psi$  initialement définie par Sprecher pour construire  $\xi$  est discontinue pour certaines valeurs d'entrée, nous utiliserons la fonction  $\psi$  garantissant continuité et monotonie définie dans (Braun *et al.*, 2007). Les autres parties de l'algorithme de Sprecher ne seront pas modifiées.

#### Définition 1. (Notations).

- $d$  est la dimension,  $d \geq 2$ .
- $m$  est le nombre de couches de fonctions,  $m \geq 2d$ .
- $\gamma$  est la base des variables  $x_i$ ,  $\gamma \geq m + 2$ .
- $a = \frac{1}{\gamma(\gamma-1)}$  est la translation entre deux couches de pavage.

–  $\lambda_1 = 1$  et pour  $2 \leq i \leq d$ ,  $\lambda_i = \sum_{r=1}^{\infty} \frac{1}{\gamma^{(i-1)(d^r-1)/(d-1)}}$  sont les coefficients de la combinaison linéaire qui sert d'argument à la fonction  $g$ .

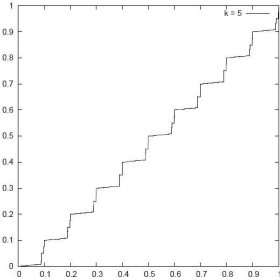
Les travaux de Sprecher portent sur la calculabilité de la fonction  $\xi$  et la définition de  $\psi$ . Plus précisément, la construction de  $\psi$  et la structure de l'algorithme sont basées sur la décomposition des nombre réels dans la base  $\gamma$ ; *i.e.* on peut écrire tout nombre décimal (noté  $d_k$ ) entre  $[0, 1]$  à  $k$  décimales sous la forme :

$$d_k = \sum_{r=1}^k i_r \gamma^{-r}, \text{ et } d_k^n = d_k + n \sum_{r=2}^k \gamma^{-r} \text{ définit un } d_k \text{ translaté.} \quad [2]$$

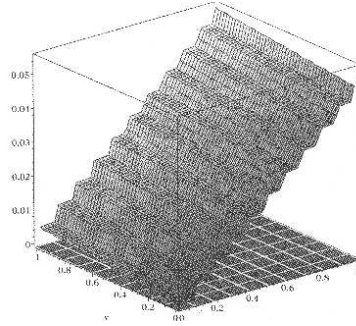
En utilisant les  $d_k$  définis en équation 2, Braun *et al.* définissent la fonction  $\psi$  par :

$$\psi_k(d_k) = \begin{cases} d_k & \text{pour } k = 1, \\ \psi_{k-1}(d_k - \frac{i_k}{\gamma^k}) + \frac{i_k}{\gamma^{\frac{d^k-1}{d-1}}} & \text{pour } k > 1 \text{ et } i_k < \gamma - 1, \\ \frac{1}{2}(\psi_k(d_k - \frac{1}{\gamma^k}) + \psi_{k-1}(d_k + \frac{1}{\gamma^k})) & \text{pour } k > 1 \text{ et } i_k = \gamma - 1. \end{cases} \quad [3]$$

La figure 2 représente le graphe de la fonction  $\psi$  sur l'intervalle  $[0, 1]$ . La fonction  $\xi$  est calculée par combinaison linéaire des réels  $\lambda_i$  et de la fonction  $\psi$  appliquée à chaque composante  $x_i$  de la valeur d'entrée. La figure 3 représente la fonction  $\xi$  sur l'espace  $[0, 1]^2$ .



**Figure 2.** Graphe de la fonction  $\psi$  pour la base  $\gamma = 10$ , de (Braun *et al.*, 2007).



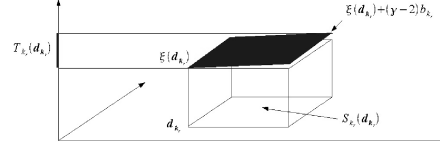
**Figure 3.** Fonction de hashage  $\xi$  pour  $d = 2$  et  $\gamma = 10$ , de (Brattka, 2004).

Sprecher a montré que des intervalles disjoints  $I$  étaient associés à des intervalles disjoints  $\psi(I)$ . Cette propriété de séparation est utilisée pour obtenir les intervalles définissant un pavage incomplet de  $[0, 1]$ . Cette méthode est étendue à un espace  $d$ -dimensionnel en effectuant le produit cartésien des intervalles  $I$ . Afin de couvrir tout l'espace, le pavage est translaté plusieurs fois d'une constante  $a$ , ce qui produit les différentes couches finales du pavage. On obtient ainsi  $2d + 1$  couches : le pavage

original constitué par des hypercubes disjoints ayant des images disjointes par  $\psi$ , et  $2d$  translatsés d'une constante  $a$  le long de chaque dimension. La figure 4 représente une coupe du pavage d'un espace 2D : on voit les  $2d + 1 = 5$  pavages différents, décalés par  $a$ , qui se superposent. Pour un espace 2D, un hypercube est associé à



**Figure 4.** Vue en coupe d'un pavage dans le cas 2D avec  $\gamma = 10$  (5 couches différentes). De (Brattka, 2004).



**Figure 5.** La fonction  $\xi$  met en correspondance chaque pavé à un intervalle  $T_k$  dans  $[0, 1]$ .

un couple  $\mathbf{d}_{k_r} = (d_{k_r,1}, d_{k_r,2})$ . L'hypercube  $S_{k_r}(\mathbf{d}_{k_r})$  est associé à un intervalle de valeur  $T_{k_r}(\mathbf{d}_{k_r})$  par la fonction  $\xi$ . La figure 5 montre le passage d'un hypercube  $S$  à un intervalle  $T$  par la fonction  $\xi$  dans le cas 2D.

Une fois les fonctions internes  $\psi$  et  $\xi$  déterminées, il reste à évaluer les fonctions externes  $g_n$ . Or, ces fonctions ne peuvent pas être déterminées directement. Sprecher propose la construction de  $r$  fonctions  $g_n^r$  dont la somme converge vers la fonction externe  $g_n$ . L'algorithme construit itérativement en trois étapes une fonction externe  $g_n^r$ . La fonction  $f_r$  définit l'erreur d'approximation, qui tend vers 0 quand  $r$  croît. L'algorithme est initialisé avec  $f_0 = f$  et  $r = 1$ .

### 3.1. première étape : détermination de la précision et construction du pavage

Avant tout calcul, nous déterminons la précision  $k_r$  pour l'étape courante. Pour deux coordonnées  $x_i$  et  $x'_i$  appartenant à deux ensembles, appartenant à la même dimension  $i$  et espacées d'une distance donnée, la distance des images des deux ensembles  $\mathbf{x}$  et  $\mathbf{x}'$  produites par  $f$  doit être inférieure au  $N^{i\text{ème}}$  de l'oscillation de  $f$ , i.e. :  $\text{si } |x_i - x'_i| \leq \frac{1}{\gamma^{k_r}}, |f_{r-1}(x_1, \dots, x_d) - f_{r-1}(x'_1, \dots, x'_d)| \leq \epsilon \|f_{r-1}\|$ . Une fois  $k_r$  déterminé, le pavage  $d_{k_r,1}^n, \dots, d_{k_r,d}^n$  est calculé par :  $\forall i \in \llbracket 1, d \rrbracket, d_{k_r,i}^n = d_{k_r,i} + n \sum_{r=2}^{k_r} \frac{1}{\gamma^r}$ .

### 3.2. deuxième étape : les fonctions internes $\psi$ et $\xi$

Pour  $n$  de 0 à  $m$ , déterminer  $\psi(d_{k_r}^n)$  et  $\xi(d_{k_r,1}^n, \dots, d_{k_r,d}^n)$  avec les équations 2 et 3.

### 3.3. troisième étape : détermination de l'erreur d'approximation

$\forall n \in \llbracket 0, m \rrbracket$ , calculer :

$$g_n^r \circ \xi(x_1 + an, \dots, x_d + an) = \frac{1}{m+1} \sum_{d_{k_r,1}^n, \dots, d_{k_r,d}^n} f_{r-1}(d_{k_r,1}, \dots, d_{k_r,d}) \theta_{d_{k_r}^n}(\xi(x_1 + an, \dots, x_d + an)),$$

où  $\theta$  est définie dans la définition 2. Ensuite, calculer :

$$f_r(x_1, \dots, x_d) = f(x_1, \dots, x_d) - \sum_{n=0}^m \sum_{j=1}^r g_n^j \circ \xi(x_1 + an, \dots, x_d + an).$$

A la fin de la  $r^{\text{ième}}$  étape, le résultat de l'approximation de  $f$  est donné par la somme des  $m+1$  couches de  $r$  fonctions  $g_n^r$  déterminées précédemment :

$$f \approx \sum_{n=0}^m \sum_{j=1}^r g_n^j \circ \xi(x_1 + an, \dots, x_d + an).$$

**Définition 2.** Définition de la fonction  $\theta$  utilisée pour le calcul de la fonction  $g_n^r$ .

$$\theta_{d_k^n}(y_n) = \sigma \left( \gamma^{\frac{d^{k+1}-1}{d-1}} (y_n - \xi(d_k^n)) + 1 \right) - \sigma \left( \gamma^{\frac{d^{k+1}-1}{d-1}} (y_n - \xi(d_k^n) - (\gamma-2)b_k) \right), \quad [4]$$

où  $\sigma$  est une fonction continue vérifiant :  $\begin{cases} \sigma(x) \equiv 0, \text{ pour } x \leq 0 \\ \sigma(x) \equiv 1, \text{ pour } x \geq 1 \end{cases}$ ,

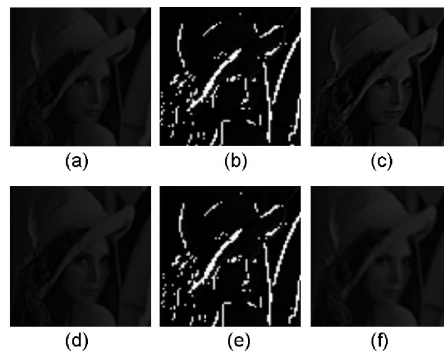
$$\text{et : } b_k = \sum_{r=k+1}^{\infty} \frac{1}{\gamma^{\frac{d^r-1}{d-1}}} \sum_{p=1}^n \lambda_p.$$

L'algorithme est très proche de l'énoncé du théorème et la fonction interne est identique pour toute fonction  $f$ . Seules les fonctions externes sont adaptées pour chaque approximation. A chaque itération, un pavage est construit. L'oscillation de la fonction d'erreur  $f_r$  tend vers 0, par conséquent, plus le nombre d'itérations est important, meilleure est l'approximation de la fonction  $f$  par les fonctions  $g$ .

### 3.4. Résultats

Nous présentons ici les résultats de la décomposition appliquée à des images en niveaux de gris, pouvant être vues comme des fonctions bidimensionnelles  $f(x, y) = I(x, y)$ . La figure 6 représente deux couches issues de l'approximation obtenue après une et deux itérations. La somme de toutes ces couches donne l'approximation de l'image originale par l'algorithme. Les points blancs visibles sur les images 6(b) et

6(e) correspondent aux valeurs négatives de la fonction externe. La figure 7 montre deux reconstructions d'une même image de départ après une et deux itérations de l'algorithme de Sprecher. Les couches issues de la décomposition sont très similaires, ce qui est normal puisque chaque couche correspond à une fraction d'un sous-échantillon de la fonction  $f$ , légèrement translatée par la valeur  $a$ . Dans le cas de fonctions 2D, on observe que la reconstruction tend très rapidement vers l'image originale. On voit d'ailleurs très peu de différence entre la première et la seconde approximation sur les figures 7(b) et 7(c).



**Figure 6.** (a) et (b) La première couche ( $n = 0$ ) après une et deux itérations ( $r = 1, r = 2$ ) respectivement. (c) Somme de (a) et (b), reconstruction partielle donnée par la première couche. (d) et (e) La dernière couche ( $n = 5$ ) après une et deux itérations ( $r = 1, r = 2$ ) respectivement. (f) Somme de (d) et (e), reconstruction partielle donnée par la dernière couche.



**Figure 7.** (a) Image originale. (b) et (c) Reconstruction après une et deux itérations respectivement.

#### 4. Conclusion et perspectives

Nous avons abordé le problème de la décomposition de fonctions multivariées en utilisant le théorème de superposition de Kolmogorov. Nous avons présenté l'algorithme proposé par Sprecher qui construit des fonctions internes et externes proches de la formulation du théorème : pour toute fonction  $f$  à décomposer, une fonction

interne unique est utilisée, et seules les fonctions externes sont adaptées. Nous avons ensuite appliqué cet algorithme à la décomposition de fonctions bidimensionnelles particulières : des images en niveaux de gris. Nos résultats sur des images montrent que l'algorithme converge rapidement vers l'image originale.

Ce premier travail préliminaire a permis de constater plusieurs aspects particulièrement intéressants de ce type de décomposition. Sprecher *et al.* ont montré dans (Sprecher *et al.*, 2002) qu'en inversant la fonction  $\xi$ , il était possible d'obtenir une courbe de remplissage de l'espace. Par conséquent, il est possible de passer d'une image à un signal 1D : grâce à la bijectivité de la fonction  $\psi$ , on obtient pour chaque pixel de l'image une valeur dans  $[0, 1]$ . Plusieurs questions restent ouvertes : comment contrôler la fonction de remplissage ? Peut-on obtenir un balayage différent dans certaines zones de l'image ? Le principal inconvénient de l'algorithme de Sprecher est sa rigidité : les fonctions internes  $\psi$  ne peuvent pas être modifiées ; la courbe de remplissage de l'espace, c'est-à-dire le balayage de l'image, est toujours la même. Igel'nik *et al.* ont récemment proposé, dans (Igel'nik *et al.*, 2003), un algorithme pour approximer les fonctions internes et externes, offrant une plus grande flexibilité. Il permet notamment de faire varier le nombre de couches de pavage et de générer plusieurs fonctions internes : on obtient plusieurs balayages différents. Dans de futurs travaux, nous nous intéresserons à l'approche d'Igel'nik, à la construction des fonctions internes et externes et leur influence sur les courbes de remplissage ; notre objectif étant l'obtention d'un balayage adapté aux données.

## 5. Bibliographie

- Brattka V., « Du 13-ième problème de Hilbert à la théorie des réseaux de neurones : aspects constructifs du théorème de superposition de Kolmogorov », *L'héritage de Kolmogorov en mathématiques. Éditions Belin, Paris*.p. 241-268, 2004.
- Braun J., Griebel M., « On a constructive proof of Kolmogorov's superposition theorem », *Constructive approximation*, 2007.
- Hecht-Nielsen R., « Kolmogorov's mapping neural network existence theorem », *Proceedings of the IEEE International Conference on Neural Networks III, New York*.p. 11-13, 1987.
- Igel'nik B., Parikh N., « Kolmogorov's Spline Network », *IEEE transactions on neural networks*, vol. 14, n° 4, p. 725-733, 2003.
- Sprecher D. A., « An improvement in the superposition theorem of Kolmogorov », *Journal of Mathematical Analysis and Applications*, n° 38, p. 208-213, 1972.
- Sprecher D. A., « A Numerical Implementation of Kolmogorov's Superpositions », *Neural Networks*, vol. 9, n° 5, p. 765-772, 1996.
- Sprecher D. A., « A Numerical Implementation of Kolmogorov's Superpositions II », *Neural Networks*, vol. 10, n° 3, p. 447-457, 1997.
- Sprecher D. A., Draghici S., « Space-filling curves and Kolmogorov superposition-based neural networks », *Neural Networks*, vol. 15, n° 1, p. 57-67, 2002.