
AMS X02 : Méthodes numériques avancées et calcul haute performance pour la simulation de phénomènes complexes

Encadrants :

- Marc Massot
- Laurent Series



Multi-Résolution adaptative

Hugo Negrel

Février 2023

Table des matières

| | | |
|----------|------------------------------------|----------|
| 1 | Introduction | 3 |
| 2 | Décomposition multi-échelle | 3 |
| 3 | Conclusion | 9 |

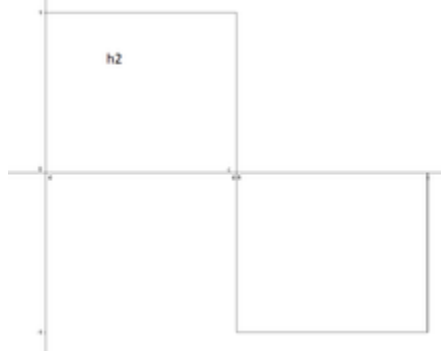


FIGURE 1 – Ondelette de Haar

1 Introduction

Ce TP porte sur la résolution multi-échelle, c'est à dire la description d'une fonction de carré intégrable à l'aide d'une base de fonctions bien choisies, les fameuses ondelettes de Haar. Il s'agit donc de décomposer une fonction initiale en somme d'ondelette, sur une certaine grille de précision $J = 8$. Pour cela, nous avons 4 fonctions à disposition pour tester cette décomposition multi-échelle à l'aide d'ondelette.

2 Décomposition multi-échelle

La décomposition multi-échelle va être employée sur 4 fonctions distinctes avec les ondelettes de Haar. On traite ici des fonctions définies sur un intervalle continu, mais on considère immédiatement que les fonctions ne sont définies, par un processus de moyennisation, que sur des intervalles de longueurs non déterminées. D'où l'adjectif 'adaptatif'. On rappelle que la décomposition se fait en représentant la fonction en deux quantités : un vecteur c_{j_0} représentant la fonction en moyenne, et un vecteur $d_{j,k}$ représentant les détails, exploité avec l'ondelette de Haar. (figure 1)

ϕ est la fonction d'échelle duale, tandis que ψ est l'ondelette duale. En toute généralité, on peut représenter une fonction en décomposition multi-échelle sous la forme suivante :

$$f = \sum_k c_{j_0,k} \phi_{j_0,k} + \sum_{j \geq j_0} \sum_k d_{j,k} \psi_{j,k}$$

$\phi_{j,k}$ est la fonction ϕ dilatée de 2^{-j} et translatée de k . On rappelle que l'opérateur de projection permet de passer d'un niveau de précision j_0 au niveau $j_0 + 1$ sur l'espace des moyennes de f (première somme), et non des détails (deuxième). Toute la subtilité de la compression réside dans le choix du seuil de troncature des détails, c'est à dire dans le choix des $d_{j,k}$ que l'on va enlever dans la deuxième somme. Notons enfin le point fondamental suivant : la donnée d'un vecteur U_J sur une grille de niveau J et la donnée d'un vecteur U_{j_0} sur une grille moins fine $j_0 \geq J$ avec ses détails du niveau j_0 jusqu'au niveau J sont tout à fait équivalents. Le procédé de compression joue sur cette équivalence de manière plus ou moins subtile. On rappelle qu'on a le résultat suivant pour l'estimation en norme du détail :

$$d_\lambda \leq C 2^{-sj} |u|_{C^s([-1,1])}$$

où $|\lambda| = j$. La régularité de la fonction est donc directement liée au contrôle du détail d_λ via le paramètre s . Les fonctions qui vont servir d'exemple sont les suivantes :

$$f_1(x) = \exp(-50x^2)$$

$$f_2(x) = \begin{cases} 1 - 2|x| & \text{si } -1/2 < x < 1/2 \\ 0 & \text{sinon.} \end{cases}$$

$$f_3(x) = 1 - \sqrt{|\sin \frac{\pi}{2} x|}$$

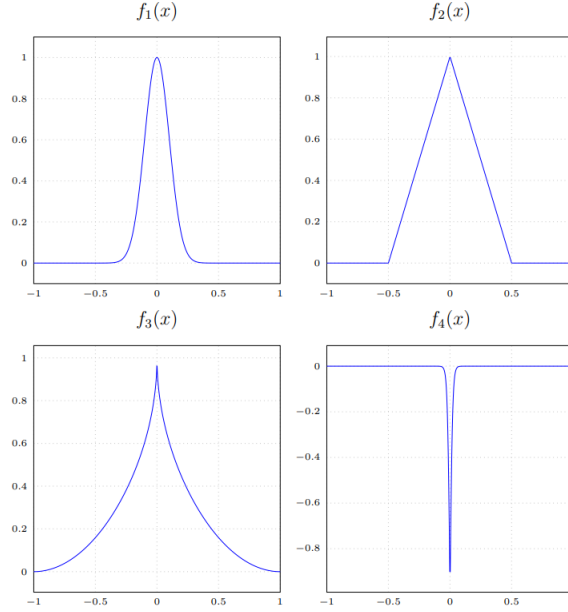


FIGURE 2 – Les quatres fonctions tests utilisées

et

$$f_4(x) = \tanh(50|x|) - 1$$

On peut rapidement commenter les paramètres de la multi-résolution. Il y a le niveau minimum et maximal (noté j jusqu'ici), qui représente la profondeur de l'arbre Λ qui sera utilisé. On a aussi le 'stencil', qui est l'ensemble des indices $\gamma = (j, k) \in \Lambda$ utilisés pour l'opération de prédiction. Pour faire simple, si on connaît la fonction en moyenne à l'échelle j , on va essayer de la prédire à l'échelle $j + 1$ plus fine. Il y a aussi le 'threshold', qui définit le seuil de détail que l'on ne souhaite pas conserver dans notre décomposition multi-échelle. Pour ce cas simple, l'arbre utilisé sera une grille dyadique, c'est à dire de la forme $I_{j,k} = [2^{-j}k, 2^{-j}(k + 1)]$. L'intervalle $[-1, 1]$ sera divisé en sous intervalle de cette forme. Sur la figure 2 sont représentées les quatres fonctions tests :

Ces fonctions sont toutes très régulières, surtout f_1 , qui est bien évidemment $C^\infty([-1, 1])$. Cependant, f_2 n'est pas dérivable en 0, $-1/2$ et $1/2$, tandis que f_3 et f_4 ne sont pas dérivables en 0 uniquement. Les figures 3,4,5 et 6 représentent les résultats de la compression pour les différentes fonctions.

Bien évidemment, plus l'arbre Δ a de feuille au plus haut niveau, moins il y aura de compression, puisque l'on sera alors au maximum de précision possible et l'on garde alors toute l'information initiale. On remarque de suite que f_1 , bien que la plus régulière sur l'ensemble du domaine, est la fonction pour laquelle la compression est la moins efficace, ce qui peut sembler contre-intuitif. L'erreur L^2 est aussi la plus importante dans ce cas. On aurait tendance à dire que cela ait dû à la présence de point d'inflexion et de forte pente. La vitesse de décroissance des coefficients dans la suite de l'expansion de Taylor doit être corrélée au niveau de détail requis pour décrire au mieux la fonctions, et donc minimiser l'erreur entre la fonction compressé et la fonction 'intacte'. La fonction f_2 peut nous induire en ce sens. En effet, la fonction est évidemment polynomiale d'ordre 1 par morceau. Ainsi, on observe que mis-à-part les points anguleux, le niveau de l'arbre reste bas, et la fonction a pu être compressé sans aucune perte d'information du tout. De plus, la fonction a le plus haut taux de compression des fonctions tests. On peut en tirer donc que les polynômes se compressent très bien, de par la nature des ondelettes mis en jeu et de la simplicité de la fonction elle-même, mais les points non dérivables restent cependant délicats. La fonction f_3 a un taux de compression et une erreur L^2 comparable à f_1 , ce qui traduit, même si régulière, la non approximabilité de la fonction par des polynômes de petit degré. Encore une fois, on constate que l'erreur se concentre autour des forts gradients. L'erreur semble monter, puis descendre brusquement, puis monter de nouveau. Cela est sans doute explicable assez facilement : l'erreur monte car l'arbre n'a pas encore monter de niveau pour augmenter le nombre de détail, puis redescend soudainement lorsque cela arrive. Cette montée de niveau soudaine a pour effet d'augmenter efficacement la précision et donc de diiminuer l'erreur. On constate bien la corrélation entre les sauts sur la figure de niveau et d'erreur. En ce qui concerne f_4 , malgré une singularité en 0 qui est compensé par un arbre au niveau maximal 8, la compression fonctionne très bien, avec une petite erreur L^2 . On peut en déduire alors qu'il peut être

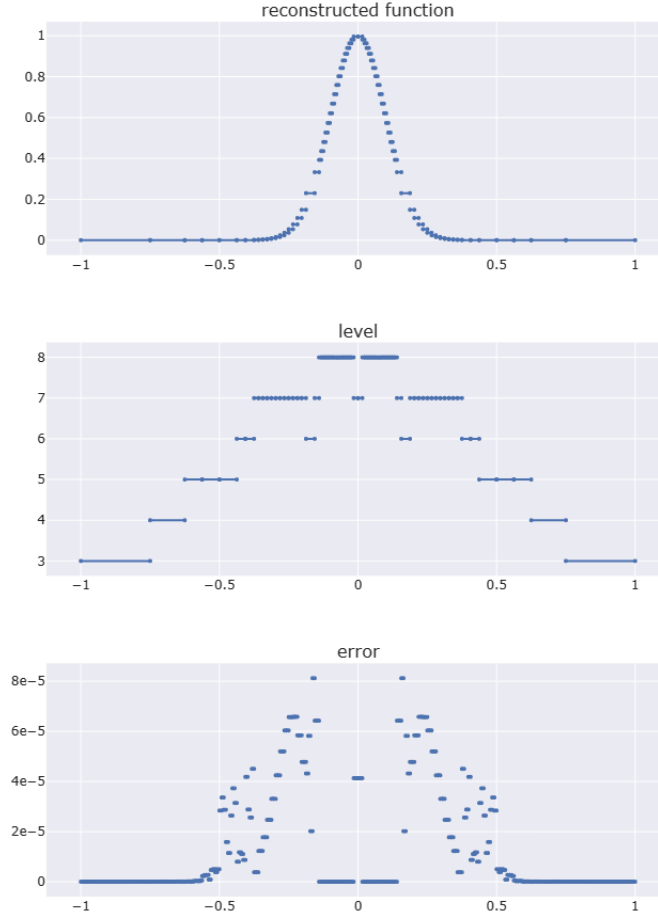


FIGURE 3 – Compression de f_1 . Nombre de feuille : 76. Taux de compression : 0.703125. Norme L^2 de l'erreur : $1.490300\text{e-}05$

assez délicat de prévoir à l'avance si la compression par ondelette va être assez efficace ou non. Les fonctions f_1 et f_3 sont très régulières et ne sont pas les plus efficaces pour autant.

On pourrait maintenant faire évoluer le seuillage pour voir à quel point l'erreur L^2 et le taux de compression augmente pour chacune des fonctions. On remarque en premier lieu que le taux de compression pour f_2 est constant, étant donné que le seuil ne supprime aucun détail, ces derniers étant déjà nuls. Comme on pouvait aussi s'y attendre, en diminuant le seuil, le taux de compression diminue aussi puisqu'on garde plus d'information, et peu de détails sont supprimés par le seuil. Le même raisonnement s'applique pour la norme L^2 , qui diminue au fur et à mesure que le taux de compression diminue.

On peut ensuite modifier le cardinal du stencil (figure 9). Il était jusqu'ici défini à 1. En augmentant le stencil, on augmente aussi le nombre de feuille, et mécaniquement leurs niveaux moyens. Le taux de compression diminue aussi, car l'opérateur de prédiction aura besoin du niveau $j - 1$. Cela sous-tend à conserver d'avantage d'information et donc à moins compresser. Ainsi, pour f_2 et f_4 , le taux de compression diminue. Ce n'est cependant pas aussi simple, puisque pour f_1 et f_3 , le taux de compression augmente. Pour un stencil de 5, f_1 obtient même le meilleur taux de compression, alors qu'elle a le pire avec un stencil de 1. Augmenter le stencil améliore donc grandement l'efficacité de l'opérateur de prédiction dans ce cas. On peut le comprendre de la manière suivante : pour f_2 et f_4 , un stencil de 1 était déjà optimal, l'augmenter ne fait qu'encombrer la compression car on doit stocker plus de données pour appliquer l'opérateur de prédiction. Par contre, pour f_1 et f_3 , cela rend la compression plus efficace par le biais de l'opérateur de projection.

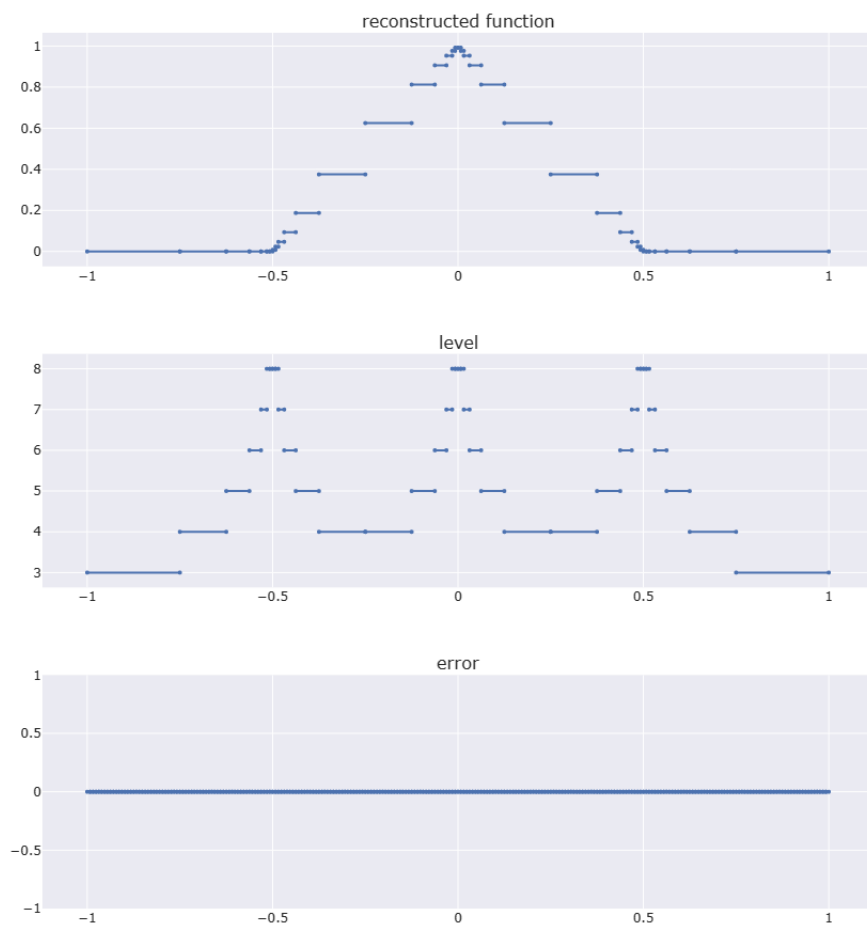


FIGURE 4 – Compression de f_2 . Nombre de feuille : 38. Taux de compression : 0.8515625. Norme L^2 de l'erreur : 0.0

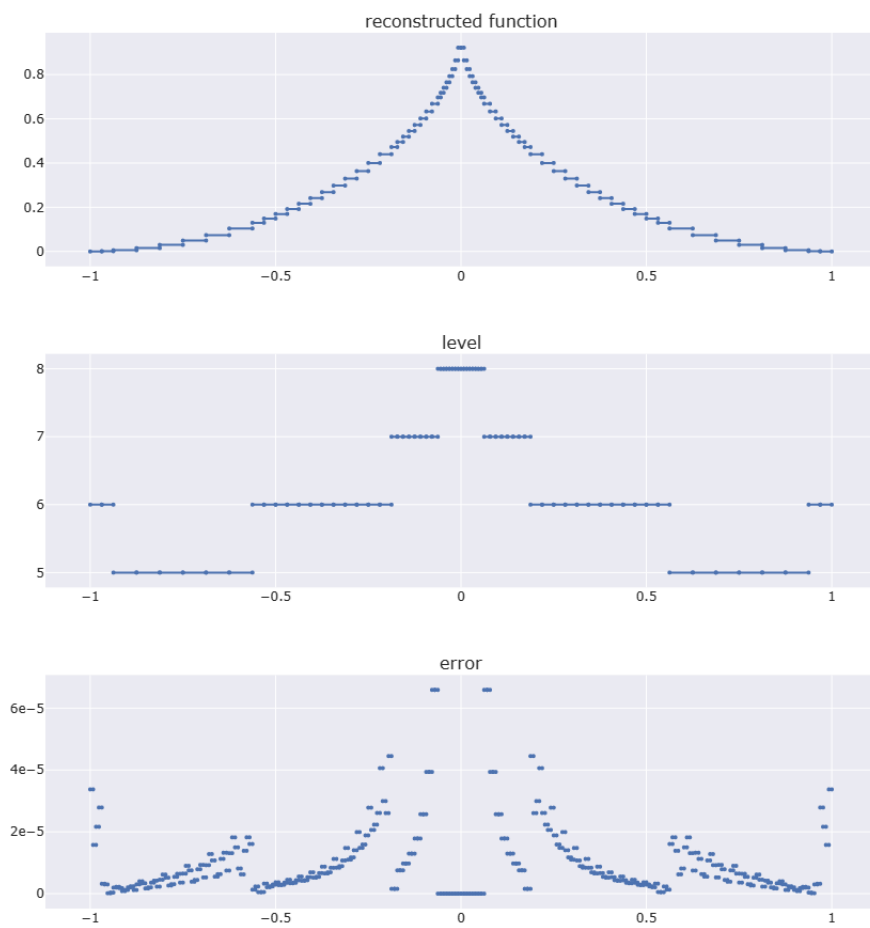


FIGURE 5 – Compression de f_3 . Nombre de feuille : 72. Taux de compression : 0.71875. Norme L^2 de l'erreur : 8.92837×10^{-6}

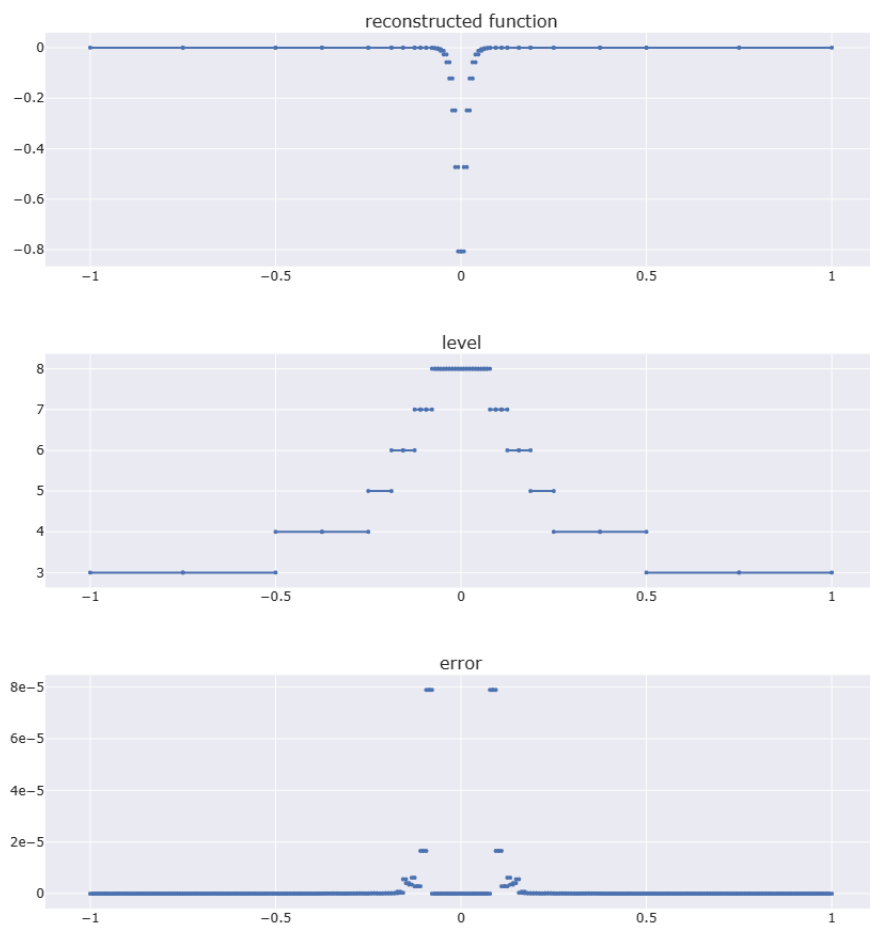


FIGURE 6 – Compression de f_4 . Nombre de feuille : 40. Taux de compression : 0.84375. Norme L^2 de l'erreur : 5.84659×10^{-6}

| epsilon | Colonne2 | Colonne3 | Colonne4 | Colonne5 |
|----------|---------------|----------|---------------|---------------|
| | f1 | f2 | f3 | f4 |
| 1,00E-02 | 0.00010343619 | | 0 4.59697e-05 | 3.136029e-05 |
| 1,00E-03 | 1.490300e-05 | | 0 8.92837e-06 | 5.84659e-06 |
| 1,00E-04 | 9.521111e-07 | | 0 1.22189e-06 | 2.5858066e-07 |
| 1,00E-05 | 7.245047e-08 | | 0 1.29056e-07 | 6.264594e-08 |

FIGURE 7 – Tableau des erreurs L^2

| epsilon | Colonne2 | Colonne3 | Colonne4 | Colonne5 |
|----------|-----------|-----------|-----------|-----------|
| | f1 | f2 | f3 | f4 |
| 1,00E-02 | 0.828125 | 0.8515625 | 0.8203125 | 0.8671875 |
| 1,00E-03 | 0.703125 | 0.8515625 | 0.71875 | 0.84375 |
| 1,00E-04 | 0.5546875 | 0.8515625 | 0.5546875 | 0.8203125 |
| 1,00E-05 | 0.4921875 | 0.8515625 | 0.328125 | 0.8125 |

FIGURE 8 – Tableau des taux de compression

| stencil | Colonne2 | Colonne3 | Colonne4 | Colonne5 |
|----------|-----------|-----------|-----------|-----------|
| | f1 | f2 | f3 | f4 |
| 1,00E+00 | 0.703125 | 0.8515625 | 0.71875 | 0.84375 |
| 2,00E+00 | 0.7890625 | 0.75 | 0.7890625 | .828125 |
| 4,00E+00 | 0.84375 | .6171875 | 0.7890625 | 0.78125 |
| 5,00E+00 | 0.8515625 | .609375 | 0.7578125 | 0.7578125 |

FIGURE 9 – Tableau du taux de compression en fonction du stencil

3 Conclusion

En conclusion, la compression permet de ne garder qu'une partie limitée de l'information stockée. Le taux de compression, de 70% au moins, se retrouve dans le nombre de feuille utilisé dans l'arbre. On en a à la base $256 = 2^8$, pour en avoir moins de 80 après compression. La régularité de la fonction n'est pas nécessairement synonyme de compression efficace. On pourra retenir en général que les fonctions en polynômes par morceaux sont celles qui se décrivent le plus efficacement par les ondelettes de Haar. Les fonctions avec beaucoup de point d'inflexions, qui peuvent décrire un comportement polynomial assez complexe, se décrivent par conséquent moins bien. La situation peut cependant s'arranger sur le stencil, brique élémentaire pour la construction d'un opérateur de projection bien définie.