Une équivalence entre algorithmes multiniveaux et algorithmes de descente par blocs

Guillaume LAUGA[1] Elisa RICCIETTI[1] Luis BRICEÑO-ARIAS[2] Nelly PUSTELNIK[3] Paulo GONÇALVES[1]

[1]ENS de Lyon, CNRS, Université Claude Bernard Lyon 1, Inria, LIP, UMR 5668, 69342, Lyon cedex 07, France

[2]Departamento de Matemática, Universidad Técnica Federico Santa María, Santiago, Chile

[3]Ens de Lyon, CNRS, Laboratoire de Physique, F-69342, Lyon, France

Résumé – Dans cet article, nous présentons une analogie entre algorithmes multiniveaux et algorithmes de descente par blocs pour un problème de restauration d'images. Les algorithmes multiniveaux utilisent une hiérarchie d'approximations du problème de minimisation à résoudre pour en accélérer la solution. Nous montrons que dans certains cas cette hiérarchie est telle que l'algorithme multiniveau est équivalent à un algorithme de descente par blocs. Cette équivalence ouvre de nouvelles possibilités théoriques pour les algorithmes multiniveaux et pratiques pour les algorithmes de descente par blocs.

Abstract – In this article, we present an analogy between multilevel algorithms and block descent algorithms for an image restoration problem. Multilevel algorithms use a hierarchy of approximations to the minimization problem to be solved, in order to speed up the solution. In some cases, we show that this hierarchy can be constructed in such a way that the multilevel algorithm is equivalent to a block descent algorithm. This equivalence opens up new theoretical perspectives for multilevel algorithms and practical perspectives for block-coordinate descent algorithms.

1 Introduction

En restauration d'images dans un contexte de bruit Gaussien, le problème de restauration peut s'écrire comme la minimisation de la somme d'une fonction d'attache aux données quadratique et d'une fonction de régularisation $g \circ \mathrm{D}$:

$$\widehat{\mathbf{x}} \in \underset{\mathbf{x}\in\mathbb{R}^N}{\operatorname{Argmin}}\, F(\mathbf{x}) = \frac{1}{2}\|\mathrm{A}\mathbf{x} - \mathbf{z}\|_2^2 + \lambda g\left(\mathrm{D}\mathbf{x}\right), \qquad (1)$$

où $\mathbf{z} := \mathrm{A}\bar{\mathbf{x}} + \epsilon$ est l'image dégradée, A représente l'opérateur linéaire de dégradation, ϵ le bruit de mesure Gaussien, D un opérateur linéaire, et λ le paramètre de régularisation. L'augmentation de la dimension de ces problèmes de restauration appelle à la définition d'algorithmes d'optimisation de plus en plus efficaces. Parmi les méthodes possibles, les algorithmes proximaux par blocs se distinguent par le faible coût en calcul et en mémoire par itération [14, 13, 20, 12].

Méthodes d'accélération. Les méthodes par blocs ont démontré leur efficacité sur de nombreux problème d'optimisation en apprentissage [14, 13, 20, 12]. En imagerie, leur efficacité est moins claire et est limitée à certains contextes. Les images astronomiques qui contiennent peu de points sources sur un fond uniforme sont bien adaptées à ce type de méthode [19]. Dans un cadre plus général, on peut jouer sur les propriétés de la régularisation [16, 2] afin de définir des blocs. D'autre part, les méthodes multiniveaux, qui utilisent une hiérarchie d'approximations de la fonction objective pour en accélérer sa résolution, sont à même de traiter des images de grande taille [15, 9, 8, 6]. Les algorithmes multiniveaux montrent des similarités avec les approches par blocs en alternant entre itérations à plus faible complexité qui agissent sur des versions à basse résolution de l'image (niveaux grossières) et itérations sur toute l'image (niveau fin) pour résoudre le problème dans sa dimension originale. Elles présentent également des différences structurelles apparentes comme la présence d'un terme de cohérence du premier ordre qui n'est pas nécessaire dans l'approche par bloc. Dès lors, il devient intéressant de se poser la question des équivalences entre les deux approches, ce qui permettrait d'ouvrir de nouvelles perspectives dans les deux directions.

Contributions et plan. Dans cet article, nous établissons une équivalence entre algorithmes proximaux multiniveaux et algorithmes de descente par blocs sur un problème de restauration pénalisé sur une base d'ondelettes (D encode donc une transformée en ondelettes). Dans la section 2, nous faisons d'abord quelques rappels sur l'analyse multirésolution, mais également sur la formulation à la synthèse du problème de restauration ainsi que sa résolution par un algorithme proximal par bloc ou un algorithme multiniveau. En explicitant les opérations de chacun de ces deux algorithmes, nous montrons qu'ils effectuent les mêmes itérations. Cette équivalence est présentée section 4. Dans la section 5, nous comparons diverses mises à jour des blocs et montrons le gain significatif dans sa formulation multiniveau. Enfin nous présentons quelques conséquences de cette équivalence qui ouvre de nouvelles perspectives théoriques et pratiques pour les deux approches, multiniveaux et descente par blocs.

2 Contexte

L'analyse multirésolution en quelques mots. Soit $\mathbf{x}$, une image de taille $2^J \times 2^J$ définie sur un domaine Ω de $\mathbb{R}^2$ et telle que $\mathbf{x} \in V_J \subset L_2(\Omega)$. L'espace V_J se décompose en une somme de deux espaces orthogonaux V_{J-1} et W_{J-1} correspondant respectivement aux espaces d'approximation et

de détail à la résolution inférieure $J-1$. En décomposant récursivement l'espace d'approximation V_{J-1} selon le même schéma, alors, pour tout $0 \leq J_0 < J$, on peut écrire [11] :

$$V_J = V_{J_0} \bigoplus_{j=J_0}^{J-1} W_j.$$

Par suite, en définissant Π_{V_j} et Π_{W_j} (resp. $\Pi^*_{V_j}$ et $\Pi^*_{W_j}$ leurs adjoints) les opérateurs de projection orthogonale de $\mathbf{x}$ sur les sous-espaces respectifs V_j et W_j et en notant $\boldsymbol{a}_j$ et $\boldsymbol{d}_j$ les coefficients d'approximation et de détail correspondants, il vient :

$$\mathbf{x} = \Pi^*_{V_{J_0}} \boldsymbol{a}_{J_0} + \sum_{j=J_0}^{J-1} \Pi^*_{W_j} \boldsymbol{d}_j.$$

De plus, dans ce schéma d'analyse multirésolution, les opérateurs de projection vérifient la propriété $\Pi_{V_j}\Pi^*_{V_j} = \mathrm{Id}_{V_j}$ (resp. $\Pi_{W_j}\Pi^*_{W_j} = \mathrm{Id}_{W_j}$) où Id_{V_j} (resp. Id_{W_j}) est l'identité dans V_j (resp. W_j). Notons enfin, que si V_j est de dimension $2^j \times 2^j$, W_j est lui constitué de 3 sous-espaces de taille $2^j \times 2^j$, correspondant chacun à l'une des 3 orientations dans l'image $\{horiz., vert., diag.\}$. Le vecteur $\boldsymbol{d}_j$ est alors de taille 3×2^{2j}.

Restauration par ondelettes. Dans la suite, pour simplifier la présentation, nous nous concentrons sur la décomposition sur deux niveaux, on omet alors l'indice de résolution pour écrire $\mathbf{x} = \Pi^*_V \boldsymbol{a} + \Pi^*_W \boldsymbol{d}$. Notons que le bloc d contient les trois groupes de coefficients de détails. Ainsi $\mathrm{D} = [\Pi_V^\top, \Pi_W^\top]^\top$.

Sous l'hypothèse que g est séparable, on peut réécrire le problème (1) sous sa forme à la synthèse en utilisant la décomposition en ondelettes de $\mathbf{x}$ comme :

$$\left(\widehat{\boldsymbol{a}}, \widehat{\boldsymbol{d}}\right) \in \underset{\boldsymbol{a}\in V, \boldsymbol{d}\in W}{\mathrm{Argmin}} \; \Psi(\boldsymbol{a}, \boldsymbol{d}) := \frac{1}{2}\|\mathrm{A}\left(\Pi^*_V \boldsymbol{a} + \Pi^*_W \boldsymbol{d}\right) - \mathbf{z}\|^2 + \lambda_a g_a(\boldsymbol{a}) + \lambda_d g_d(\boldsymbol{d}). \quad (2)$$

On obtient la solution du Problème (1) à partir de la solution du Problème (2) avec la relation :

$$\widehat{\mathbf{x}} = \Pi^*_V \widehat{\boldsymbol{a}} + \Pi^*_W \widehat{\boldsymbol{d}}.$$

3 Algorithmes pour la restauration

Afin d'établir les liens entre approches multiniveaux et approches par blocs nous commençons par présenter les itérations associées à ces deux algorithmes, pour la résolution du Problème (2).

Restauration par ondelettes : la descente par blocs. Pour minimiser Ψ, notre première stratégie est d'utiliser un algorithme de descente par blocs, en agissant sur les coefficients d'approximation et de détail selon une stratégie déterministe ou aléatoire. Partant du point $(\boldsymbol{a}^{[0]}, \boldsymbol{d}^{[0]})$, avec $\tau > 0$ et $\varepsilon_{a,n}, \varepsilon_{d,n} \in \{0, 1\}$, on définit les itérations suivantes :

$$\begin{array}{l}\text{pour } n = 0, 1, \ldots \\ \left\lfloor \begin{array}{l} \boldsymbol{u}^{[n+1]} = \mathrm{A}^*\left(\mathrm{A}\left(\Pi^*_V \boldsymbol{a}^{[n]} + \Pi^*_W \boldsymbol{d}^{[n]}\right) - \mathbf{z}\right) \\ \boldsymbol{a}^{[n+1]} = \boldsymbol{a}^{[n]} + \varepsilon_{a,n}\left(\mathrm{prox}_{\tau\lambda_a g_a}\left(\boldsymbol{a}^{[n]} - \tau\Pi_V \boldsymbol{u}^{[n+1]}\right) - \boldsymbol{a}^{[n]}\right) \\ \boldsymbol{d}^{[n+1]} = \boldsymbol{d}^{[n]} + \varepsilon_{d,n}\left(\mathrm{prox}_{\tau\lambda_d g_d}\left(\boldsymbol{d}^{[n]} - \tau\Pi_W \boldsymbol{u}^{[n+1]}\right) - \boldsymbol{d}^{[n]}\right) \end{array} \right. \end{array} \quad (3)$$

En choisissant $(\varepsilon_{a,n}, \varepsilon_{d,n}) = (1, 1)$ pour tout n, on récupère l'algorithme forward-backward (FB). En imposant que seul un des blocs soit mis à jour à chaque itération, on peut obtenir un algorithme cyclique. Les valeurs de $(\varepsilon_{a,n}, \varepsilon_{d,n})$ peuvent aussi être choisies aléatoirement, tant que $\mathsf{P}\left[(\varepsilon_{a,n}, \varepsilon_{d,n}) = (0, 0)\right] = 0$ [18].

Restauration par ondelettes : algorithme multiniveau. Nous présentons maintenant la construction d'un algorithme proximal multiniveaux pour minimiser Ψ que nous présentons à deux niveaux par simplicité. L'indice h sera utilise pour faire référence au niveau fin et l'indice H au niveau grossier.

Conformément à [9], l'opérateur de transfert d'informations I_h^H (qui envoie l'information du niveau fin au niveau grossier) est la projection Π_V sur V, et l'opérateur de prolongation (qui envoie l'information du niveau grossier au niveau fin) I_H^h est directement Π^*_V. On note Ψ_H, la fonction objectif associée au niveau grossier, une approximation de Ψ qui prend en compte que les coefficients d'approximations et que l'on définit comme suit :

$$\Psi_H(\boldsymbol{a}) = \frac{1}{2}\|\mathrm{A}\Pi^*_V \boldsymbol{a} - \Pi^*_V \Pi_V \mathbf{z}\|_2^2 + \lambda_a g_a(\boldsymbol{a}) + \langle v_H, \boldsymbol{a}\rangle \quad (4)$$

où v_H impose la cohérence du premier ordre entre deux versions lissées Ψ_μ and $\Psi_{H,\mu}$ [9, Définition 2.1] de Ψ et Ψ_H, [9, Définition 2.4] :

$$\boldsymbol{v}_H = \Pi_V \nabla\Psi_\mu(\boldsymbol{a}^{[n]}, \boldsymbol{d}^{[n]}) - \nabla\Psi_{H,\mu}(\boldsymbol{a}^{[n]}), \quad (5)$$

avec $\Psi_{H,\mu}(\cdot) = 1/2\|\mathrm{A}\Pi^*_V \cdot - \Pi^*_V \Pi_V \mathbf{z}\|_2^2 + \lambda_a g_{a,\mu} + \langle \boldsymbol{v}_H, \cdot\rangle$, où $g_{a,\mu}$, $\mu > 0$ est la version lissée de g_a (suivant le cadre de [1]).

La méthode multiniveau alterne des itérations fines et des itérations grossières. Dans la suite, nous supposons pour simplifier que nous ne calculons qu'une seule itération grossière avant de revenir au niveau fin, une extension à plusieurs itérations est directe [7, 6]. Cette itération produit l'itéré intermédiaire $\boldsymbol{a}^{[n+1/2]}$ à partir de $\boldsymbol{a}^{[n]}$. Le niveau grossier n'étant pas nécessairement lisse, on utilise une étape de gradient proximal pour optimiser Ψ_H.

L'algorithme de gradient-proximal à deux niveaux s'écrit donc comme :

$$\begin{array}{l}\text{pour } n = 0, 1, \ldots \\ \left\lfloor \begin{array}{l} \boldsymbol{a}^{[n+1/2]} = \mathrm{prox}_{\tau\lambda_a g_a}\left(\boldsymbol{a}^{[n]} - \tau\mathrm{A}^*_H\left(\mathrm{A}_H \boldsymbol{a}^{[n]} - \Pi^*_V \Pi_V \mathbf{z}\right) - \tau \boldsymbol{v}_H\right) \\ \boldsymbol{u}^{[n+1]} = \mathrm{A}^*\left(\mathrm{A}\left(\Pi^*_V \boldsymbol{a}^{[n+1/2]} + \Pi^*_W \boldsymbol{d}^{[n]}\right) - \mathbf{z}\right) \\ \boldsymbol{a}^{[n+1]} = \mathrm{prox}_{\tau\lambda_a g_a}\left(\boldsymbol{a}^{[n+1/2]} - \tau\Pi_V \boldsymbol{u}^{[n+1]}\right) \\ \boldsymbol{d}^{[n+1]} = \mathrm{prox}_{\tau\lambda_d g_d}\left(\boldsymbol{d}^{[n]} - \tau\Pi_W \boldsymbol{u}^{[n+1]}\right) \end{array} \right. \end{array} \quad (6)$$

4 Équivalence bloc-multiniveau

Dans cette section, nous démontrons l'équivalence annoncée entre les algorithmes (3) et (6).

Hypothèse 1. *On suppose que*

1. *l'opérateur de transfert d'information I_h^H est la projection Π_V sur V,*
2. *dans la définition de v_H dans* (5)*, les modèles fins et grossiers sont lissés selon la même méthode, avec un paramètre $\mu > 0$ [1],*

3. Ψ et Ψ_H sont cohérentes au premier ordre.

Le lemme suivant montre que la cohérence du premier ordre transfère au niveau grossier la contribution des coefficients de détail au gradient de l'attache aux données.

Lemme 1. *Supposons que l'hypothèse 1 soit vérifiée. Le terme de cohérence du premier ordre $\boldsymbol{v}_H$ dans l'équation* (5) *au point $(\boldsymbol{a}^{[n]}, \boldsymbol{d}^{[n]})$ est donné par :*

$$\boldsymbol{v}_H = \Pi_V \mathrm{A}^* \left(\mathrm{A} \Pi_W^* \boldsymbol{d}^{[n]} - \Pi_W^* \Pi_W \mathbf{z} \right). \tag{7}$$

Démonstration. Par définition de la cohérence du premier ordre entre fonctions lissées [9, Définition 2.1], v_H est donné par l'équation (5) Le second terme est directement obtenu en explicitant le gradient au niveau grossier :

$$\begin{aligned}
\nabla \Psi_{H,\mu}(\boldsymbol{a}^{[n]}) &= \nabla \left(\frac{1}{2} \| \mathrm{A}_H \boldsymbol{a}^{[n]} - \Pi_V^* \Pi_V \mathbf{z} \|_2^2 + \lambda_a g_{a,\mu}(\boldsymbol{a}^{[n]}) \right) \\
&= \mathrm{A}_H^* \left(\mathrm{A}_H \boldsymbol{a}^{[n]} - \Pi_V^* \Pi_V \mathbf{z} \right) + \lambda_a \nabla_a g_{a,\mu}(\boldsymbol{a}^{[n]}) \\
&= \Pi_V \mathrm{A}^* \left(\mathrm{A} \Pi_V^* \boldsymbol{a}^{[n]} - \Pi_V^* \Pi_V \mathbf{z} \right) + \lambda_a \nabla_a g_{a,\mu}(\boldsymbol{a}^{[n]}).
\end{aligned}$$

Quant au premier terme dans l'équation (5), on a :

$$\begin{aligned}
\nabla \Psi_{\mu}(\boldsymbol{a}^{[n]}, \boldsymbol{d}^{[n]}) &= \nabla \left(\frac{1}{2} \| \mathrm{A} \left(\Pi_V^* \boldsymbol{a}^{[n]} + \Pi_W^* \boldsymbol{d}^{[n]} \right) - \mathbf{z} \|_2^2 \right) \\
&\quad + \nabla \left(\lambda_a g_{a,\mu}(\boldsymbol{a}^{[n]}) + \lambda_d g_{d,\mu}(\boldsymbol{d}^{[n]}) \right) \\
&= \mathrm{A}^* \left(\mathrm{A} \left(\Pi_V^* \boldsymbol{a}^{[n]} + \Pi_W^* \boldsymbol{d}^{[n]} \right) - \mathbf{z} \right) \\
&\quad + \begin{bmatrix} \lambda_a \nabla_a g_{a,\mu}(\boldsymbol{a}^{[n]}) \\ \lambda_d \nabla_d g_{d,\mu}(\boldsymbol{d}^{[n]}) \end{bmatrix}.
\end{aligned}$$

Et ainsi :

$$\begin{aligned}
v_H &= \Pi_V \mathrm{A}^* \left(\mathrm{A} \left(\Pi_V^* \boldsymbol{a}^{[n]} + \Pi_W^* \boldsymbol{d}^{[n]} \right) - \mathbf{z} \right) \\
&\quad + \Pi_V \begin{bmatrix} \lambda_a \nabla_a g_{a,\mu}(\boldsymbol{a}^{[n]}) \\ \lambda_d \nabla_d g_{d,\mu}(\boldsymbol{d}^{[n]}) \end{bmatrix} \\
&\quad - \Pi_V \mathrm{A}^* \left(\mathrm{A} \Pi_V^* \boldsymbol{a}^{[n]} - \Pi_V^* \Pi_V \mathbf{z} \right) - \lambda_a \nabla_a g_{\mu}(\boldsymbol{a}^{[n]}).
\end{aligned}$$

Comme V et W sont orthogonaux, $\Pi_V \left(\lambda_d \nabla g_\mu(\boldsymbol{d}^{[n]}) \right) = 0$. Dès lors :

$$\boldsymbol{v}_H = \Pi_V \mathrm{A}^* \left(\mathrm{A} \Pi_W^* \boldsymbol{d}^{[n]} - \Pi_W^* \Pi_W \mathbf{z} \right),$$

où nous avons utilisé le fait que

$$\mathbf{z} - \Pi_V^* \Pi_V \mathbf{z} = \Pi_W^* \Pi_W \mathbf{z}.$$

□

On peut désormais écrire complètement l'étape de gradient-proximal au niveau grossier à l'itération n :

$$\begin{aligned}
\boldsymbol{a}^{[n+1/2]} &= \mathrm{prox}_{\tau \lambda_a g_a} \left(\boldsymbol{a}^{[n]} - \tau \mathrm{A}_H^* \left(\mathrm{A}_H \boldsymbol{a}^{[n]} - \Pi_V^* \Pi_V \mathbf{z} \right) - \tau \boldsymbol{v}_H \right) \\
&= \mathrm{prox}_{\tau \lambda_a g_a} \left(\boldsymbol{a}^{[n]} - \tau \Pi_V \mathrm{A}^* \left(\mathrm{A} \Pi_V^* \boldsymbol{a}^{[n]} - \mathbf{z} + \mathrm{A} \Pi_W^* \boldsymbol{d}^{[n]} \right) \right) \\
&= \mathrm{prox}_{\tau \lambda_a g_d} \left(\boldsymbol{a}^{[n]} - \tau \Pi_V \mathrm{A}^* \left(\mathrm{A} \left(\Pi_V^* \boldsymbol{a}^{[n]} + \Pi_W^* \boldsymbol{d}^{[n]} \right) - \mathbf{z} \right) \right)
\end{aligned}$$

Par conséquent, le pas de gradient proximal au niveau grossier est exactement égal à un pas de gradient proximal au niveau fin, pour les coefficients d'approximation. Les algorithmes multiniveaux peuvent ainsi s'inscrire dans le formalisme par blocs en alternant entre la mise à jour de $\boldsymbol{a}$ seul, puis de $\boldsymbol{a}$ et $\boldsymbol{d}$ ensemble. Ainsi, nous avons le lemme suivant :

Lemme 2. *L'algorithme à deux niveaux défini dans l'équation* (6) *est équivalent à l'algorithme à deux blocs défini dans l'équation* (3) *en choisissant $\varepsilon_{a,2n} = 1$ et $\varepsilon_{d,2n} = 0$, puis $\varepsilon_{a,2n+1} = 1$ et $\varepsilon_{d,2n+1} = 1$ pour tout n.*

Quelques conséquences. Étudier la convergence des algorithmes par blocs nous donnent donc la convergence de leur équivalent multiniveau [7] pour des choix spécifiques de ε. Par exemple, les résultats de convergence des algorithmes par blocs obtenus dans [7] dans un cadre non lisse et non convexe, sont transposables à leur équivalent multiniveau.

Il a aussi été démontré que les algorithmes multiniveaux peuvent accélérer la résolution des problèmes de restauration d'images [5, 3, 17, 9, 10, 8, 4, 15]. Il est donc aussi intéressant pour un algorithme par blocs de suivre des règles de mise à jour inspirées de ces algorithmes multiniveaux.

Dans tous les cas, la convergence vers un minimiseur de l'algorithme résultant, est garantie sous l'hypothèse que Ψ est convexe et que τ est choisi par rapport à la constante de Lipschitz associée au gradient partiel des blocs mis à jours [6, 7]. Ce résultat de convergence ne s'applique pas directement à la suite $(\boldsymbol{a}^{[n]}, \boldsymbol{d}^{[n]})_{n \in \mathbb{N}}$: cette convergence n'est possible que si $\varepsilon_{a,n} = \varepsilon_{d,n} = 1$ pour tout n. Par exemple pour l'algorithme "multiniveau" ci-dessus la sous-séquence convergente est $(\boldsymbol{a}^{[2n]}, \boldsymbol{d}^{[2n]})_{n \in \mathbb{N}}$.

5 Expériences numériques

Dans cette section, nous étudions les performances de l'algorithme multiniveau (par blocs) obtenues en le comparant aux algorithmes par blocs de la littérature.

Contexte expérimental. Dans cette section, nous considérons la restauration du Cameraman, de taille 1024×1024. L'observation est dégradée par un flou gaussien et un bruit gaussien additif (voir Figure 1). La régularisation sera effectuée dans une base d'ondelettes de Haar.

Algorithmes de références. Nous comparons notre algorithme multiniveau à l'algorithme forward-backward et deux versions de l'algorithme FB par blocs : une version cyclique où un bloc sera mis à jour à chaque itération dans un ordre prédéfini, et une version aléatoire ou le bloc à mettre à jour est tiré au sort à chaque itération. Nous comparons ces algorithmes à notre algorithme multiniveau. L'algorithme à 2 niveaux proposé va mettre à jour 8 fois les coefficients d'approximations, avant de faire 2 mises à jour complètes. Se contenter d'une seule mise à jour sur les coefficients d'approximation (comme dans l'équivalence présentée), conduit à des résultats tout juste meilleurs que ceux du FB.

Résultats. Les résultats des expériences numériques sont présentés en Figure 1. Le paramètre de régularisation est choisi pour maximiser le rapport signal à bruit (RSB) à convergence. L'algorithme multiniveau est plus rapide que toute les autres méthodes. Les deux autres algorithmes par blocs ont des comportements à peu près équivalents, inférieur à FB. Le multiniveau peut exploiter la connaissance *a priori* de ses itérations pour éviter des projections/prolongations superflues.

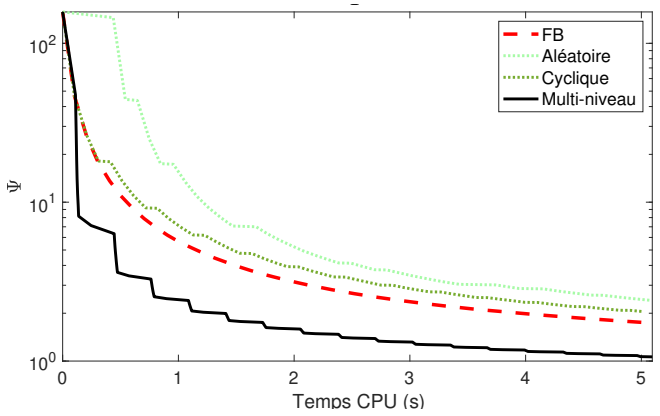

FIGURE 1 : Comparaison de la convergence de l'algorithme multiniveau par rapport à FB et des algorithmes de descente par blocs pour le problème de défloutage régularisé par une ondelette de Haar sur deux niveaux. Image du cameraman de taille 1024×1024. Dégradation : $\sigma_{\text{bruit}} = 0.001$. Noyau de flou gaussien de taille 20×20 et d'écart type 3.6. $\lambda_a = 1 \times 10^{-10}$, $\lambda_d = 1 \times 10^{-4}$.

6 Conclusion

Dans cet article, nous avons montré une équivalence entre algorithmes multiniveaux et algorithmes de descente par blocs. De cette équivalence, on peut tirer des enseignements pour les approches multiniveaux et les approches par blocs.

Construction des modèles grossiers – Pour obtenir l'équivalence, le modèle grossier choisi contient une forme de cohérence du second ordre avec le niveau fin. Ainsi la Hessienne de l'attache aux données du niveau grossier est une approximation de Galerkin de la Hessienne de l'attache aux données du niveau fin ($\Pi_V \mathrm{A}^* \mathrm{A} \Pi_V^*$ par exemple). Ainsi, le niveau grossier incorpore l'information de la courbure du niveau fin.

Règles de mise à jour des blocs – Il est souvent proposé dans les approches par blocs d'exploiter les différences entre les constantes de Lipschitz pour accélérer la résolution. Dans notre cas les constantes de Lipschitz sont égales, ainsi l'algorithme multiniveau exploite plutôt la différence d'importance, vis-à-vis de la solution, des blocs.

Généralisation à plusieurs échelles – L'équivalence présentée ici est généralisable si l'on décompose sur plus de niveaux de résolutions (cf [6, Chapitre 7]). Il reste cependant à étudier plus en profondeur la convergence en pratique d'une mise à jour des blocs par cette règle multiniveau.

Perspectives – L'équivalence démontrée ici est limitée à un contexte assez précis. Pour la généraliser il faudrait pouvoir définir une notion générale de hiérarchie qui permet de faire apparaître, comme pour les ondelettes, une importance de certains blocs plutôt que d'autres.

Références

[1] A. BECK et M. TEBOULLE : Smoothing and First Order Methods : A Unified Framework. *SIAM Journal on Optimization*, 22(2):557–580, janvier 2012.

[2] E. CHOUZENOUX, J.-C. PESQUET et A. REPETTI : A block coordinate variable metric forward–backward algorithm. *Journal of Global Optimization*, 66(3):457–485, 2016.

[3] S. W. FUNG et Z. WENDY : Multigrid Optimization for Large-Scale Ptychographic Phase Retrieval. *SIAM Journal on Imaging Sciences*, 13(1):214–233, janvier 2020.

[4] V. HOVHANNISYAN, P. PARPAS et S. ZAFEIRIOU : MAGMA : Multilevel Accelerated Gradient Mirror Descent Algorithm for Large-Scale Convex Composite Minimization. *SIAM Journal on Imaging Sciences*, 9(4):1829–1857, janvier 2016.

[5] A. JAVAHERIAN et S. HOLMAN : A Multi-Grid Iterative Method for Photoacoustic Tomography. *IEEE Transactions on Medical Imaging*, (3):696–706, mars 2017.

[6] G. LAUGA : *Multilevel proximal methods and application to image restoration*. phdthesis, École Normale Supérieure de Lyon, décembre 2024.

[7] G. LAUGA, L. BRICEÑO-ARIAS, E. RICCIETTI, N. PUSTELNIK et P. GONÇALVES : A flexible block-coordinate forward-backward algorithm for non-smooth and non-convex optimization. *to appear*, 2025.

[8] G. LAUGA, A. REPETTI, E. RICCIETTI, N. PUSTELNIK, P. GONÇALVES et Y. WIAUX : A multilevel framework for accelerating uSARA in radio-interferometric imaging. *In 2024 32nd European Signal Processing Conference (EUSIPCO)*, pages 2287–2291, 2024.

[9] G. LAUGA, E. RICCIETTI, N. PUSTELNIK et P. GONÇALVES : IML FISTA : A Multilevel Framework for Inexact and Inertial Forward-Backward. Application to Image Restoration. *SIAM Journal on Imaging Sciences*, 17(3):1347–1376, 2024.

[10] G. LAUGA, E. RICCIETTI, N. PUSTELNIK et P. GONÇALVES : Multilevel Fista For Image Restoration. *IEEE ICASSP, Rhodes, Greece*, 4-10 June 2023.

[11] S. MALLAT : *A wavelet tour of signal processing*. Elsevier, 1999.

[12] Y. NESTEROV : Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

[13] J. NUTINI, I. LARADJI et M. SCHMIDT : Let's make block coordinate descent converge faster : faster greedy rules, message-passing, active-set complexity, and superlinear convergence. *Journal of Machine Learning Research*, 23(131):1–74, 2022.

[14] J. NUTINI, M. SCHMIDT, I. LARADJI, M. FRIEDLANDER et H. KOEPKE : Coordinate descent converges faster with the gauss-southwell rule than random selection. *In Proc. ICML'15*, 2015.

[15] P. PARPAS : A Multilevel Proximal Gradient Algorithm for a Class of Composite Optimization Problems. *SIAM Journal on Scientific Computing*, 39(5):S681–S701, 2017.

[16] B. PASCAL, N. PUSTELNIK, P. ABRY et J.-C. PESQUET : Block-Coordinate Proximal Algorithms for Scale-Free Texture Segmentation. *In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1253–1257, Calgary, AB, avril 2018. IEEE.

[17] J. PLIER, F. SAVARINO, M. KOČVARA et S. PETRA : First-Order Geometric Multilevel Optimization for Discrete Tomography. *In Scale Space and Variational Methods in Computer Vision*, volume 12679, pages 191–203. Springer International Publishing, Cham, 2021.

[18] S. SALZO et S. VILLA : Parallel random block-coordinate forward–backward algorithm : a unified convergence analysis. *Mathematical Programming*, 193(1):225–269, mai 2022.

[19] Y. SUN, J. LIU et U. KAMILOV : Block Coordinate Regularization by Denoising. *In Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[20] S. J. WRIGHT : Coordinate descent algorithms. *Mathematical programming*, 151(1):3–34, 2015.