Un algorithme MCMC distribué pour la résolution de problèmes inverses de grande dimension

Pierre-Antoine Thouvenin, Audrey Repetti, Pierre Chainais

Un algorithme MCMC distribué pour la résolution de problèmes inverses de grande dimension

Pierre-Antoine THOUVENIN[1], Audrey REPETTI[2], Pierre CHAINAIS[1]

[1]Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France

[2]Department of Actuarial Mathematics & Statistics, Heriot-Watt University, Edinburgh EH14 4AS, UK

[3]Institute of Sensors, Signals and Systems, Heriot-Watt University, Edinburgh EH14 4AS, UK

pierre-antoine.thouvenin@centralelille.fr, a.repetti@hw.ac.uk, pierre.chainais@centralelille.fr

Résumé – La résolution de problèmes inverses de grande dimension peut être abordée efficacement à l'aide d'algorithmes distribués. Dans la mesure où les techniques de simulation permettent d'obtenir simultanément estimateurs et intervalles de crédibilité associés, ce travail s'intéresse à l'utilisation d'une technique d'augmentation de modèle et d'algorithmes MCMC pour introduire un échantillonneur distribué. Par-delà l'architecture client-serveur communément utilisée dans la littérature, nous proposons un échantillonneur basé sur une architecture Single Program Multiple Data, pour laquelle une même tâche est affectée à l'ensemble des noeuds de calcul impliqués dans la résolution du problème. L'intérêt de cette approche est illustré dans le cadre d'un problème d'inpainting de grande taille basé sur un *a priori* de type variation totale.

Abstract – Inference in large scale inverse problems can be efficiently tackled with distributed algorithms. Since sampling algorithms allow estimates and credibility intervals to be simultaneously formed, we investigate in this work a combination of data augmentation and MCMC algorithms to design a distributed sampler. Departing from the usual client-server setting adopted in the literature, we propose a distributed sampler based on a Single Program Multiple Data architecture, where all workers are assigned the same task. Experiments conducted on a synthetic inpainting problem with a total variation prior illustrate the performance of the proposed approach.

1 Introduction

Un problème inverse consiste à estimer un ensemble de paramètres $\boldsymbol{x} \in \mathbb{R}^N$ à partir d'observations dégradées $\boldsymbol{y} \in \mathbb{R}^M$. Observations et paramètres sont reliés par un modèle de la forme

$$\boldsymbol{y} = \mathcal{D}(\boldsymbol{A}\boldsymbol{x}), \tag{1}$$

où $\boldsymbol{A} \in \mathbb{R}^{M \times N}$ représente l'opérateur de mesure et $\mathcal{D}\colon \mathbb{R}^M \to \mathbb{R}^M$ modélise toute perturbation aléatoire affectant les mesures. L'inférence requiert l'ajout d'information *a priori* sur les paramètres, conduisant à la distribution *a posteriori*

$$\pi(\boldsymbol{x} \mid \boldsymbol{y}) \propto \exp\left(-f_{\boldsymbol{y}}(\boldsymbol{A}\boldsymbol{x}) - g(\boldsymbol{B}\boldsymbol{x})\right), \tag{2}$$

où $f_{\boldsymbol{y}}\colon \mathbb{R}^M \to]-\infty, +\infty]$ est le terme d'attache aux données issu de la description statistique de $\boldsymbol{y}$, et $\boldsymbol{B} \in \mathbb{R}^{P \times N}$ et $g\colon \mathbb{R}^P \to]-\infty, +\infty]$ forment le terme d'*a priori* sur $\boldsymbol{x}$ (*e.g.*, variation totale (TV) [1]). En pratique, la difficulté du problème augmente avec le nombre d'observations et de paramètres, et requiert un algorithme de résolution pouvant passer à l'échelle. L'utilisation d'algorithmes distribués est une solution courante, en ce qu'ils permettent de diviser paramètres et observations entre plusieurs noeuds de calcul.

En optimisation, les méthodes d'éclatement (*splitting*) regroupent un grand nombre d'approches distribuées, parmi lesquelles la méthode lagrangienne Alternating Direction Method of Multipliers (ADMM) [2] et les algorithmes primaux-duaux [3]. Bien que ces approches exploitent efficacement les ressources de calcul, elles ne permettent de former qu'un estimateur au sens du *maximum a posteriori*, sans pouvoir en quantifier directement l'incertitude. Ceci s'avère problématique en l'absence de vérité terrain, par exemple en astronomie [4].

Les méthodes de Monte Carlo par chaîne de Markov (MCMC) permettent quant à elles de former estimateurs et intervalles de crédibilité à partir d'échantillons tirés suivant la loi *a posteriori* (2) [5]. En revanche, ces algorithmes sont traditionnellement moins adaptés pour traiter des problèmes de grande dimension. Inspirés par des techniques d'optimisation, plusieurs échantillonneurs ont récemment été proposés pour pallier ce problème [6]-[8].

Néanmoins, peu d'échantillonneurs distribués ont été proposés à ce jour [9], [10]. Inspirée de l'algorithme ADMM, la technique d'augmentation de modèle asymptotiquement exacte AXDA [10] permet, en particulier, d'améliorer le passage à l'échelle d'un algorithme MCMC grâce à une stratégie de séparation des fonctions impliquées dans (2) (*i.e.*, *splitting*) [9]. Jusqu'ici, AXDA a été exclusivement utilisée pour définir des échantillonneurs basés sur une architecture client-serveur, exploitant l'indépendance conditionelle entre les blocs de paramètres issus de l'éclatement. Toutefois, les noeuds (*clients*) doivent tous communiquer avec le serveur, ce qui crée des limitations en termes de flexibilité d'implémentation (nombre limité de blocs indépendants) et de volume des communications.

Ce travail est soutenu par la Chaire IA Sherlock ANR-20-CHIA-0031-01 portée par P. Chainais, ainsi que par le programme national d'investissement d'avenir ANR-16-IDEX-0004 ULNE et la Région Hauts-de-France.

Dans cet article, nous considérons une classe de problèmes inverses pour lesquels le couplage entre les paramètres est *localisé*, c'est-à-dire pour lesquels $\boldsymbol{A}$ et $\boldsymbol{B}$ ont une structure parcimonieuse par bloc, voir Fig. 1. Cette classe recouvre plusieurs applications usuelles, telles que la déconvolution ou l'inpainting d'images, et offre une grande flexibilité dans la mise en œuvre d'algorithmes distribués. Exploitant l'approche AXDA, nous introduisons un algorithme MCMC pour lequel nous proposons une implémentation distribuée de type Single Program Multiple Data (SPMD) [11]. Comparée à une architecture client-serveur, tous les noeuds de calcul utilisés effectuent la même tâche, pour laquelle seul un petit nombre de communications entre noeuds voisins est nécessaire. De plus, pour le type de problèmes considérés, une architecture SPMD permet d'utiliser un plus grand nombre de noeuds et de réduire le volume des communications.

La classe de problèmes considérée est introduite en Section 2. Afin de simplifier la présentation, l'approche proposée est décrite en Section 3 dans le cas particulier d'un problème inverse d'inpainting, utilisé dans les expériences. Les performances de l'algorithme sont évaluées en Section 4 sur un problème d'inpainting avec *a priori* TV, en comparaison avec [10]. Conclusions et perspectives sont reportées en Section 5.

2 Formulation du problème

2.1 Structure des problèmes étudiés

Nous considérons un problème inverse avec une loi *a posteriori* de la forme (2), dont les matrices $\boldsymbol{A}$ et $\boldsymbol{B}$ sont parcimonieuses par bloc (traduisant un couplage *localisé* entre des entrées de $\boldsymbol{x}$). Le nombre d'observations M et de paramètres N sont supposés du même ordre de grandeur, ce qui nécessite de distribuer observations et paramètres. Soit $K \in \mathbb{N}^*$ le nombre de noeuds disponibles, avec $K \leq \min\{M, N, P\}$. Une répartition des colonnes de $\boldsymbol{A}$ et $\boldsymbol{B}$ entre les K noeuds est supposée possible, avec un faible recouvrement entre noeuds. Le nombre d'éléments impliqués dans les recouvrements est supposé petit comparé à $\lfloor N/K \rfloor$. De plus, le nombre de lignes pour lesquelles des éléments non-nuls affectent plusieurs blocs est supposé faible par rapport à $\min\{\lfloor M/K \rfloor, \lfloor P/K \rfloor\}$.

Une telle structure est illustrée en Fig. 1. Un grand nombre d'opérateurs usuels en traitement d'images satisfont ces hypothèses, tels que les opérateurs de convolution, d'échantillonnage compressé et de sélection [12, Sections 3.4 and 13.3].

Enfin, le bruit est supposé tel que K blocs d'observations statistiquement indépendants peuvent être formés.

2.2 Modèle

Vraisemblance. D'après la structure introduite en Section 2.1, les observations $\boldsymbol{y}$ peuvent être partitionnées en K blocs statistiquement indépendants $\boldsymbol{y} = (\boldsymbol{y}_k)_{1 \leq k \leq K}$ conditionnellement à $\boldsymbol{x}$, tels que $\boldsymbol{y}_k \in \mathbb{R}^{M_k}$ et $M = \sum_k M_k$. L'opérateur $\mathcal{D}$ est alors séparable sous la forme $(\mathcal{D}_k(\cdot))_{1 \leq k \leq K}$, et (1) peut se décomposer en

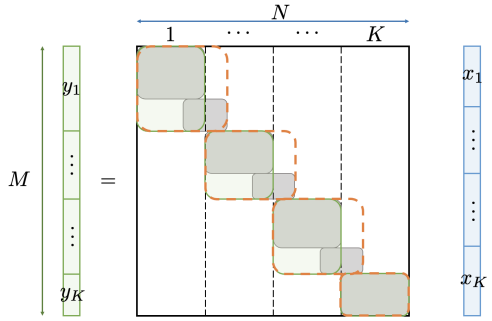

FIGURE 1 – Structure parcimonieuse par bloc de la matrice $\boldsymbol{A}$ (3). Les colonnes de $\boldsymbol{A}$ sont divisées en groupes contigus (pointillés noirs), avec un recouvrement de petite taille par rapport à $\lfloor N/K \rfloor$. Les sous-blocs $\boldsymbol{A}_k$ de (3) sont entourés en orange, et $\boldsymbol{C}_k$ extrait les composantes de $\boldsymbol{x}$ requises pour former $\boldsymbol{y}_k$.

$$\boldsymbol{y}_k = \mathcal{D}_k(\boldsymbol{A}_k \boldsymbol{C}_k \boldsymbol{x}), \tag{3}$$

où $\boldsymbol{C}_k \in \mathbb{R}^{\widetilde{N}_k \times N}$ extrait les $\widetilde{N}_k$ entrées consécutives de $\boldsymbol{x}$ nécessaires pour former $\boldsymbol{y}_k$, $\boldsymbol{A}_k \in \mathbb{R}^{M_k \times \widetilde{N}_k}$ avec $N \leq \sum_k \widetilde{N}_k$ [1]. Les matrices $\boldsymbol{A}_k$ et $\boldsymbol{C}_k$ sont illustrées sur la Fig. 1. La structure de $\mathcal{D}$ implique que $f_{\boldsymbol{y}}$ est additivement séparable

$$(\forall \boldsymbol{x} \in \mathbb{R}^N)(\forall \boldsymbol{y} \in \mathbb{R}^M) \quad f_{\boldsymbol{y}}(\boldsymbol{A}\boldsymbol{x}) = \sum_{k=1}^{K} f_{\boldsymbol{y}_k}(\boldsymbol{A}_k \boldsymbol{C}_k \boldsymbol{x}), \tag{4}$$

avec, pour tout $k \in \{1, \ldots, K\}$, $f_k \colon \mathbb{R}^{M_k} \to]-\infty, +\infty]$.

Les bloc d'observations $\boldsymbol{y}_k$ et de paramètres associés $\boldsymbol{x}_k$ peuvent tous deux être gérés par le même noeud k. En effet, la structure décrite en Section 2.1 garantit que $\boldsymbol{x}_k$ contient la majorité des composantes de $\boldsymbol{C}_k \boldsymbol{x}$. Les éléments manquants sont obtenus en communiquant avec les noeuds voisins de k.

La décomposition (3) recouvre en particulier le cas du bruit multiplicatif, de Poisson, et additif gaussien avec une matrice de covariance diagonale (par blocs).

Prior. Nous supposons que la fonction g admet une décomposition additive par blocs similaire à celle de f

$$(\forall \boldsymbol{x} \in \mathbb{R}^N) \quad g(\boldsymbol{B}\boldsymbol{x}) = \sum_{k=1}^{K} g_k(\boldsymbol{B}_k \boldsymbol{D}_k \boldsymbol{x}), \tag{5}$$

où, pour $k \in \{1, \ldots, K\}$, $g_k \colon \mathbb{R}^{P_k} \to]-\infty, +\infty]$, $\boldsymbol{B}_k \in \mathbb{R}^{P_k \times \overline{N}_k}$, et $\boldsymbol{D}_k \in \mathbb{R}^{\overline{N}_k \times N}$ est un opérateur de sélection. Plusieurs *a priori* usuels admettent la structure (5), tels que la TV [1] ou la contrainte de positivité.

3 Approche proposée

Pour simplifier la présentation, l'approche proposée est directement introduite dans le cas du problème d'inpainting traité en Section 4 (couplage entre les paramètres induit uniquement par le prior) en présence de bruit blanc gaussien. La matrice $\boldsymbol{A}$ est alors un opérateur de sélection, et la stratégie d'éclatement n'est appliquée qu'au terme d'*a priori* [2]. Dans ce cas, (4)

1. Lorsque $N = \sum_k \widetilde{N}_k$, $\boldsymbol{A}$ est diagonale par blocs (à une permutation des lignes près), et n'induit aucun couplage entre blocs de paramètres.

2. Dans le cas d'un bruit non gaussien, il peut être nécessaire d'introduire des variables auxiliaires à la fois dans le terme de vraisemblance et l'*a priori*.

Algorithm 1: Échantillonneur distribué proposé.

Entrées: $(\alpha, \beta, \tau) \in]0, +\infty[^3$, $N_{\text{MC}} \in \mathbb{N}^*$

$\boldsymbol{x}^{(0)} \in \mathbb{R}^N$, $(\boldsymbol{z}^{(0)}, \boldsymbol{u}^{(0)}) \in (\mathbb{R}^P)^2$

$\nu = \alpha\beta(\alpha+\beta)^{-1}$; $\eta = 0.99\alpha$, $\gamma = 0.99(1/\sigma^2 + \|\boldsymbol{B}^\mathsf{T}\boldsymbol{B}\|_2/\alpha)^{-1}$

Pour $t = 0$ **à** $N_{MC} - 1$ **faire**

  Pour *chaque noeud* $k \in \{1, \dots, K\}$ **faire** en parallèle

    // Mise à jour de $\boldsymbol{x}_k$ avec PSGLA [8]

    Communications requises par $\boldsymbol{D}_k$ pour calculer $\nabla_{\boldsymbol{x}_k} h_k$;

    $\boldsymbol{x}_k^{(t+1)} = \boldsymbol{x}_k^{(t)} - \gamma \nabla_{\boldsymbol{x}_k} h_k\big(\boldsymbol{x}^{(t)}, (\boldsymbol{z}_k^{(t)}, \boldsymbol{u}_k^{(t)})\big) + \sqrt{2\gamma}\boldsymbol{\xi}_k$,

    avec $\boldsymbol{\xi}_k \sim \mathcal{N}(\boldsymbol{0}, \mathbf{I}_{N_k \times N_k})$;

    Communications nécessaires au calcul de $\boldsymbol{B}_k \boldsymbol{D}_k \boldsymbol{x}^{(t+1)}$;

    // Mise à jour de $\boldsymbol{z}_k$ avec PSGLA [8]

    $\boldsymbol{z}_k^{(t+1)} = \operatorname{prox}_{\eta g_k}\big(\boldsymbol{z}_k^{(t)} - \frac{\eta}{\alpha}(\boldsymbol{z}_k^{(t)} - \boldsymbol{B}_k\boldsymbol{D}_k\boldsymbol{x}^{(t+1)} - \boldsymbol{u}_k^{(t)}) + \sqrt{2\eta}\boldsymbol{\zeta}_k\big)$, avec $\boldsymbol{\zeta}_k \sim \mathcal{N}(\boldsymbol{0}, \mathbf{I}_{P_k \times P_k})$;

    // Tirer $\boldsymbol{u}_k$ suivant sa conditionnelle

    $\boldsymbol{u}_k^{(t+1)} \sim \mathcal{N}\big(\frac{\nu}{\alpha}(\boldsymbol{z}_k^{(t+1)} - \boldsymbol{B}_k\boldsymbol{D}_k\boldsymbol{x}^{(t+1)}), \nu \mathbf{I}_{P_k \times P_k}\big)$;

Sorties: $(\boldsymbol{x}^{(t)}, \boldsymbol{z}^{(t)}, \boldsymbol{u}^{(t)})_{1 \le t \le N_{\text{MC}}}$

s'écrit, pour tout $\boldsymbol{x} \in \mathbb{R}^N$ et $\boldsymbol{y} \in \mathbb{R}^M$

$$f_{\boldsymbol{y}}(\boldsymbol{A}\boldsymbol{x}) = \frac{1}{2\sigma^2} \sum_{k=1}^{K} \|\boldsymbol{A}_k \boldsymbol{x}_k - \boldsymbol{y}_k\|_2^2, \tag{6}$$

où $\boldsymbol{A}_k \in \mathbb{R}^{M_k \times N_k}$ est un opérateur de sélection agissant sur le bloc $\boldsymbol{x}_k$, et σ^2 est la variance du bruit blanc gaussien.

Éclatement basé sur l'approche AXDA. En considérant des noyaux gaussiens, l'application d'AXDA à (2) conduit à un modèle approché de loi *a posteriori*

$$\widetilde{\pi}\big(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{u} \mid \boldsymbol{y}\big) \propto \exp\big(-h(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{u})\big), \tag{7}$$

où $h(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{u}) = f_{\boldsymbol{y}}(\boldsymbol{A}\boldsymbol{x}) + g(\boldsymbol{z}) + \frac{1}{2\alpha}\|\boldsymbol{B}\boldsymbol{x} - \boldsymbol{z} + \boldsymbol{u}\|_2^2 - \frac{1}{2\beta}\|\boldsymbol{u}\|_2^2$, et $(\alpha, \beta) \in]0, +\infty[^2$ et $(\boldsymbol{z}, \boldsymbol{u}) \in (\mathbb{R}^P)^2$ sont des variables auxiliaires associées au prior. Les hypothèses (3)–(5) permettent de réécrire h sous la forme

$$h(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{u}) = \sum_{k=1}^{K} h_k(\boldsymbol{x}, \boldsymbol{z}_k, \boldsymbol{u}_k), \tag{8}$$

avec, pour tout $k \in \{1, \dots, K\}$,

$$h_k(\boldsymbol{x}, \boldsymbol{z}_k, \boldsymbol{u}_k) = \frac{1}{2\sigma^2}\|\boldsymbol{y} - \boldsymbol{A}_k\boldsymbol{x}_k\|_2^2 + g_k(\boldsymbol{z}_k) + \frac{1}{2\alpha}\|\boldsymbol{B}_k\boldsymbol{D}_k\boldsymbol{x} - \boldsymbol{z}_k + \boldsymbol{u}_k\|_2^2 + \frac{1}{2\beta}\|\boldsymbol{u}_k\|_2^2. \tag{9}$$

Algorithme distribué proposé. La forme de la loi *a posteriori* $\widetilde{\pi}$ définie par (7)-(8)-(9) permet d'échantillonner les blocs $\boldsymbol{x}_k$, $\boldsymbol{z}_k$ et $\boldsymbol{u}_k$ sur un unique noeud k, après une étape de communication impliquant un petit nombre de noeuds voisins (d'après les hypothèses sur la structure de $\boldsymbol{B}$, voir Section 2.1). Cette configuration est parfaitement adaptée à une architecture SPMD [11], dans laquelle les noeuds effectuent une même tâche qui ne nécessite que quelques communications entre des noeuds voisins.

Nous proposons un algorithme compatible avec cette architecture, dont la structure globale est celle d'un échantillonneur de Gibbs : chacune des variables impliquées dans (8) est tirée successivement, conditionnellement aux autres. Pour traiter les variables dont la distribution conditionnelle n'est pas standard ou requiert des communications, nous proposons d'utiliser le noyau de transition de l'échantillonneur de Langevin PSGLA [8]. La procédure est décrite dans l'Algorithme 1.

Pour tout $k \in \{1, \dots, K\}$, le noyau de transition PSGLA (Algorithme 1 ligne 7) nécessite de former le gradient partiel de h_k par rapport à $\boldsymbol{x}_k$ (9). Une étape de communication est alors nécessaire, en raison du couplage induit entre plusieurs blocs de paramètres par la matrice $\boldsymbol{B}$. Une étape similaire est requise pour échantillonner $\boldsymbol{z}_k$ et $\boldsymbol{u}_k$ aux étapes 9 et 10. L'Algorithme 1 reste toutefois applicable dans le cas où $\boldsymbol{C}_k$ n'est pas une matrice identité, moyennant des communications complémentaires.

4 Expériences sur données synthétiques

Les performances de l'algorithme proposé sont évaluées sur des données synthétiques issues du modèle d'inpainting (6).

4.1 Cadre des expériences

Modèle. Le problème d'inpainting (6) est traité à l'aide d'un *a priori* TV. Celui-ci s'exprime sous la forme (5) en choisissant $\boldsymbol{B}$ comme l'opérateur de gradient discret ($P = 2N$) et $g = \tau\|\cdot\|_{2,1}$, avec $\tau > 0$ [1, Section 3.A]). La loi *a posteriori* du modèle obtenu admet une factorisation de la forme (3)-(5).

Évaluation des performances. Les performances sont évaluées en termes de temps de calcul et de qualité de l'estimateur de l'erreur quadratique moyenne minimum (MMSE), noté $\widehat{\boldsymbol{x}}$. La qualité d'estimation est quantifiée à l'aide du rapport signal sur bruit (SNR) de reconstruction. Une comparaison est conduite avec l'approche [10, Section V-B]. [3]

Expériences. Deux expériences sont considérées, basées sur $M = \lfloor 0.6N \rfloor$ observations corrompues par un bruit blanc gaussien, dont la variance assure un SNR de 40 dB.

a) *Passage à l'échelle (strong scaling) :* l'Algorithme 1 a été appliqué à la reconstruction d'une image de taille $N = 256 \times 256$ avec $K \in \{1, 4, 8, 16\}$.

b) *Inpainting en grande dimension :* la reconstruction d'une image $N = 1024 \times 1024$ est étudiée avec $K = 16$. L'approche proposée avec $K = 1$ et [10] ne permettant pas d'obtenir un résultat en temps raisonnable, la comparaison n'est conduite qu'en terme du temps de calcul par itération, évalué sur 10 itérations.

L'échantillonneur [10] est appliqué dans la configuration suggérée par ses auteurs, *i.e.*, $(\alpha, \beta, \tau) = (9, 1, 0.2)$, avec $\text{N}_{\text{MC}} = 5 \times 10^3$ échantillons, dont $\text{N}_{\text{bi}} = 200$ de chauffe. L'approche proposée utilise $(\alpha, \beta, \tau) = (9, 1, 0.2)$, $\text{N}_{\text{MC}} = 10^4$ et $\text{N}_{\text{bi}} = 5 \times 10^3$. Les expériences ont toutes été conduites sur un ordinateur de la grille de calcul de l'université de Lille [4], équipé

3. Dans la mesure où tous les blocs sont liés, noter que [10] ne peut bénéficier ici d'une implémentation distribuée client-serveur.

4. https://hpc.univ-lille.fr/

TABLE 1 – Passage à l'échelle. Le facteur d'accélération est donné par rapport à l'Algorithme 1 avec $K = 1$.

Méthode (nombre de noeuds K)	[10, Section V-B] (1)	Prop. (1)	Prop. (4)	Prop. (8)	Prop. (16)
Temps par itération moyen ($\pm$ écart-type) (10^{-3} s)	65.56 ($\pm$ 2.08)	6.07 ($\pm$ 0.42)	3.50 ($\pm$ 0.21)	1.93 ($\pm$ 0.77)	**1.08** ($\pm$ 2.35)
Accélération par itération	0.19	1	3.49	6.33	**11.30**
Temps de calcul (s)	262.20	61.04	17.50	9.63	**5.38**
SNR de reconstruction (dB)	23.33	23.45	**23.48**	23.44	**23.48**

FIGURE 2 – Inpainting d'image $N = 1024 \times 1024$. De gauche à droite : vérité terrain, observations et estimateur MMSE.

de deux processeurs Intel Xeon E5-2695 v4 series de 18 cœurs chacun, fonctionnant à 2.1 GHz. Un noeud k correspond ici a un processus associé à un cœur physique. L'algorithme a été implanté en Python grâce à la librairie `mpi4py` [13].

4.2 Résultats

Passage à l'échelle. Les résultats de la Table 1 montrent que l'approche proposée est entre 5 et 60 fois plus rapide que [10], au prix d'une qualité de reconstruction légèrement inférieure. Le facteur d'accélération est quasi idéal jusqu'à $K = 8$, et commence à saturer à partir de $K = 16$. Ces résultats montrent que l'Algorithme 1 est une alternative très intéressante à [10] pour traiter des problèmes inverses de grande dimension.

Inpainting en grande dimension. L'échantillonneur [10] nécessiterait environ 4h pour générer $\mathrm{N_{MC}} = 10^4$ échantillons, avec 1.35 s en moyenne par itération. Moins d'une heure serait requise par l'Algorithme 1 pour $K = 1$, avec 0.23 s par itération. L'estimateur MMSE de la Fig. 2, de SNR $= 26.60$ dB, a été obtenu en 90 s avec $K = 16$ (17.93 ms par itération). Cela correspond à une accélération d'un facteur 75 par rapport à [10], et 13 par rapport au cas $K = 1$.

5 Conclusion et perspectives

Une application originale de la technique d'augmentation AXDA a été proposée dans cet article pour résoudre des problèmes inverses de structure parcimonieuse par blocs. Pour former efficacement des estimateurs et en quantifier l'incertitude, nous avons introduit un échantillonneur distribué, combinant noyaux de transition d'un échantillonneur de Gibbs et de Langevin. Une architecture SPMD a été adoptée pour l'implantation de l'algorithme, offrant plus de flexibilité dans la distribution des données par rapport à une architecture client-serveur. Des expériences sur un problème synthétique d'inpainting ont illustré l'intérêt de cette approche, avec un facteur d'accélération quasi proportionnel au nombre noeuds utilisés. Par la suite, une extension au cas asynchrone sera étudiée, ainsi que des applications à des problèmes plus généraux.

Références

[1] L. Condat, « A Generic Proximal Algorithm for Convex Optimization—Application to Total Variation Minimization, » *IEEE Signal Process. Lett.*, t. 21, n° 8, p. 985-989, 2014.

[2] S. Boyd et al., « Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers, » *Foundations and Trends® in Machine Learning*, t. 3, n° 1, p. 1-122, 2011.

[3] L. Condat, « A Primal-Dual Splitting Method for Convex Optimization Involving Lipschitzian, Proximable and Linear Composite Terms, » *J. Opt. Soc. America*, t. 158, p. 460-479, 2013.

[4] X. Cai et al., « Uncertainty quantification for radio interferometric imaging – I. Proximal MCMC methods, » *MNRAS*, t. 480, n° 3, p. 4154-4169, 2018.

[5] C. P. Robert et al., *Monte Carlo Statistical Methods* (Springer Texts in Statistics), 2e éd. Springer, 2010.

[6] M. Pereyra, « Proximal Markov Chain Monte Carlo Algorithms, » *Stat. Comput.*, t. 26, n° 4, p. 745-760, 2016.

[7] A. Durmus et al., « Efficient Bayesian Computation by Proximal Markov Chain Monte Carlo : When Langevin Meets Moreau, » *SIAM J. Imaging Sci.*, t. 11, n° 1, p. 473-506, 2018.

[8] A. Salim et al., « Primal Dual Interpretation of the Proximal Stochastic Gradient Langevin Algorithm, » in *Adv. in Neural Information Processing Systems*, t. 33, 2020, p. 3786-3796.

[9] L. J. Rendell et al., « Global Consensus Monte Carlo, » *J. Comput. and Graph. Stat.*, t. 30, n° 2, p. 249-259, 2021.

[10] M. Vono et al., « Split-and-augmented Gibbs sampler - Application to large-scale inference problems, » *IEEE Trans. Signal Process.*, t. 67, n° 6, p. 1648-1661, 2019.

[11] F. Darema, « The SPMD Model : Past, Present and Future, » in *Recent Advances in Parallel Virtual Machine and Message Passing Interface* Y. Cotronis et al., Berlin, Heidelberg, 2001, p. 1-1.

[12] S. Mallat, *A Wavelet Tour of Signal Processing*, 3rd Edition. Boston : Academic Press, 2009.

[13] L. Dalcin et al., « mpi4py : Status Update After 12 Years of Development, » *IEEE Comput. Sci. Eng.*, t. 23, n° 4, p. 47-54, 2021.