Convergent plug-and-play methods for image inverse problems with explicit and nonconvex deep regularization

Samuel Hurault

université de BORDEAUX

THÈSE PRÉSENTÉE

POUR OBTENIR LE GRADE DE

DOCTEUR DE L'UNIVERSITÉ DE BORDEAUX

ECOLE DOCTORALE MATHEMATIQUES ET INFORMATIQUE

Spécialité : Mathématiques Appliquées et Calcul Scientifique

Par **Samuel Hurault**

Méthodes plug-and-play convergentes pour la résolution de problèmes inverses en imagerie avec régularisation explicite, profonde et non-convexe.

Sous la direction de : **Nicolas Papadakis**
Co-directeur : **Arthur Leclaire**

Soutenue le 27 Novembre 2023

Membres du jury :

M. Rémi Gribonval	École Normale Supérieure de Lyon	Rapporteur
M. Pierre Weiss	Institut de Mathématiques de Toulouse	Rapporteur
M. Jérôme Bolte	Université de Toulouse Capitole	Examinateur
Mme. Emilie Chouzenoux	Centre Inria de Saclay	Examinatrice
Mme. Julie Delon	Université Paris Cité	Examinatrice
M. Gabriel Peyré	École Normale Supérieure de Paris	Examinateur
M. Nicolas Papadakis	Institut de Mathématiques de Bordeaux	Directeur
M. Arthur Leclaire	Institut de Mathématiques de Bordeaux	Co-directeur

Méthodes plug-and-play convergentes pour la résolution de problèmes inverses en imagerie avec régularisation explicite, profonde et non-convexe

Résumé : Les méthodes plug-and-play constituent une classe d'algorithmes itératifs pour la résolution de problèmes inverses en imagerie, où la régularisation est effectuée par un débruiteur de bruit Gaussien. Ces algorithmes donnent de très bonnes performances de restauration, notamment lorsque le débruiteur est paramétré par un réseau de neurones profond. Cependant, l'analyse théorique de la convergence de ces méthodes reste incomplète. La plupart des résultats de convergence existants considèrent des débruiteurs non expansifs, ce qui n'est pas réaliste (ou sous-optimal), ou limitent leur analyse aux termes d'attache aux données fortement convexes. De plus, les algorithmes itératifs plug-and-play ne visent pas à minimiser une fonctionnelle explicite, ce qui peut limiter leur interprétabilité et leur contrôle numérique. Nous distinguons deux types d'algorithmes plug-and-play : les algorithmes RED, qui sont construits en supposant que le débruiteur approche le gradient du logarithme de la distribution des images propres (appelé log prior), et les algorithmes PnP, qui sont construits par approximation de l'opérateur proximal du log prior. Pour ces deux familles d'algorithmes, nous proposons de nouvelles preuves de convergence lorsqu'ils sont utilisés en conjonction avec un débruiteur spécifique. Le débruiteur proposé, appelé débruiteur "Gradient-Step", s'écrit comme une étape de descente de gradient sur un potentiel explicite et non-convexe paramétré par un réseau de neurones profond. De plus, nous démontrons que ce débruiteur peut également s'écrire comme un opérateur proximal. Nos expériences montrent que ces contraintes de paramétrisation ne compromettent les performances de débruitage. En tirant parti des résultats de convergence des algorithmes proximaux pour des problèmes non-convexes, nous démontrons que nos algorithmes RED et PnP sont des processus itératifs convergents vers des points stationnaires de fonctionnelles explicites. Certains des énoncés de convergence proposés impliquent cependant des conditions restrictives sur les paramètres du problème. Nous proposons alors une version relâchée de l'algorithme de descente de gradient proximal qui converge pour une plage de paramètres de régularisation plus large, permettant ainsi une restauration d'image plus précise. Nous appliquons nos algorithmes PnP et RED à divers problèmes inverses mal posés, tels que le défloutage, la super-résolution et l'inpainting. Nos résultats numériques valident les résultats théoriques de convergence et démontrent que nos algorithmes atteignent des performances de pointe, à la fois quantitativement et qualitativement. Ces algorithmes RED et PnP ne sont cependant pas applicables pour des observations dégradées avec un bruit de Poisson. Pour résoudre ce problème, nous proposons une généralisation du plug-and-play basé sur l'algorithme de descente de gradient Bregman (BPG). BPG remplace la distance Euclidienne par une divergence de Bregman, qui permet de mieux capturer la régularité sous-jacente d'un problème inverse donné. Nous introduisons un nouveau modèle de bruit, appelé modèle de bruit de Bregman, qui généralise le bruit Gaussien à cette nouvelle géométrie, ainsi que de nouvelles versions Bregman des algorithmes RED et PnP. Nos évaluations expérimentales, menées sur des problèmes inverses de Poisson, prouvent l'efficacité de la méthode et valident les résultats théoriques de convergence.

Mots-clés : Plug-and-Play, Problème inverse, Débruitage, Convergence, Non-convexe

Convergent plug-and-play methods for image inverse problems with explicit and nonconvex deep regularization.

Abstract: Plug-and-play methods constitute a class of iterative algorithms for imaging inverse problems where regularization is performed by an off-the-shelf Gaussian denoiser. These methods yield impressive visual results, especially when the denoiser is parameterized by a deep neural network. However, the theoretical convergence analysis of plug-and-play methods remains incomplete. Most of the existing convergence results consider nonexpansive denoisers, which is non-realistic (or suboptimal), or limit their analysis to strongly convex data-fidelity terms. Furthermore, plug-and-play iterative algorithms do not aim at minimizing any specific functional, which can limit their interpretability and numerical control. We distinguish between two types of plug-and-play algorithms: RED algorithms, which are constructed by assuming that the denoiser approximates the gradient of the logarithm of the distribution of clean images (called log prior), and PnP algorithms, which are built by approximation of the proximity operator of the log prior. For these two families of algorithms, we offer new convergence proofs when they are used in conjunction with a specific denoiser. The proposed denoiser, called Gradient-Step Denoiser, writes as a gradient descent step on an explicit and nonconvex function parameterized by a deep neural network. Additionally, we demonstrate that the Gradient-Step Denoiser can also be expressed as the proximity operator of a related but distinct nonconvex function. Besides, experiments show that this parametrization does not compromise denoising performance. Leveraging convergence results for first-order optimization algorithms in the non-convex settings, we demonstrate that the proposed RED and PnP algorithms are convergent iterative processes targeting stationary points of explicit functionals. Some of these convergence results may assume restrictive conditions on the parameters of the inverse problem. We propose a relaxed version of the Proximal Gradient Algorithm (PGD) algorithm for weakly convex optimization which converges for a wider range of regularization parameters, thus allowing more accurate image restoration. We apply our PnP and RED algorithms to various ill-posed inverse problems, such as deblurring, super-resolution, and inpainting. Our numerical results validate the convergence theory and demonstrate that our algorithms achieve state-of-the-art performance both quantitatively and qualitatively. PnP and RED methods are however not directly applicable for addressing Poisson inverse problems. To address this limit, we propose a generalization of plug-and-play using the Bregman Proximal Gradient (BPG) optimization technique. BPG replaces the Euclidean distance with a Bregman divergence, which better captures the underlying smoothness of the problem. We introduce a novel noise model, called the Bregman noise model, which extends Gaussian noise to the new Bregman geometry, as well as new Bregman versions of the RED and PnP algorithms. Our experimental evaluations, conducted on Poisson inverse problems, prove the efficiency of our method and validate the established theoretical convergence results.

Keywords: Plug-and-Play, Inverse Problem, Denoising, Convergence, Nonconvex

---

Institut de Mathématiques de Bordeaux
UMR 5251 Université de Bordeaux, 33405 TALENCE, France.

Remerciements

C'est dans le rush de la dernière semaine avant la soutenance, entre la prépararation de la slide 32 et de la slide 33, que je me lance dans l'écriture de cette section. Bien que je l'écrive au dernier moment, ce sont des paragraphes importants pour moi, et je tenais à ce qu'ils soient présents sur mon manuscrit le jour de la soutenance. C'est simplement mon sens légendaire pour l'organisation qui m'a conduit à l'écrire au dernier moment. Pour ceux d'entre vous qui lisent cette section, et qui refermeront ce manuscrit à sa fin, excusez par avance la piètre qualité de sa rédaction.

Je tiens tout d'abord à remercier mes encadrants de thèse Arthur et Nicolas. Je me considère très chanceux d'avoir pu trouver un environnement de travail à la fois bienveillant et de haut niveau où je me suis senti très vite à l'aise. Vous formez un duo d'encadrement très complémentaire et humain avec qui on peut facilement parler, aussi bien de maths que de la vie en général. Je vous remercie pour votre disponibilité au quotidien, pour votre implication et votre temps, même très tard les veilles de certaines deadlines. Je vous remercie notamment pour m'avoir laissé beaucoup de liberté, sur la direction scientifique de ma thèse, ou pour m'avoir laissé beaucoup télétravailler loin de Bordeaux.

Nicolas, en plus d'avoir été un formidable directeur de thèse, tu es un chercheur très inspirant pour moi. Au-delà du fait que nous partageons de nombreux intérêts scientifiques, je suis impressionné par ton implication et ton dévouement pour tes élèves. Avec en plus tes valeurs et tes engagements, tu représentes le chercheur solide, sympa et intègre que je souhaite devenir. Arthur, celui grâce à qui on ne laisse pas une faute d'anglais dans un preprint. Ta rigueur et ton savoir mathématique m'ont énormément aidé. Merci d'avoir toujours su répondre à mes questions (souvent naïves) de mathématiques sans jamais trop me juger.

Je veux ensuite remercier Rémi Gribonval et Pierre Weiss pour avoir rapporté ma thèse. Je suis fier que des chercheurs de votre niveau aient consacré tant de temps à lire mon travail. Je remercie également les examinateurs de ma thèse Jérôme Bolte, Emilie Chouzenoux, Julie Delon et Gabriel Peyré. Merci pour vos questions très interessantes lors de la soutenance et pour vos retours très bienveillants. Vos compliments sur mon manuscrit et ma présentation m'ont beaucoup touché.

In 2022, I had the change to visit during 3 months the group of GDP group at MIT. I would like to thank Justin for making this experience possible. I would also like to thank all the amazing people I met there. Thanks Lingxiao for all the interesting discussions and for having collaborated with me. Rickard and Marcel, my boston buddies, thanks for the all the really nice moments we spent together. Thank you Elisabetta, I am very glad I had the change to meet you.

Ensuite, j'aimerais remercier mes amis doctorants, postdoctorants et stagiaires à l'IMB. Vous avez rendu ces années de thèse vraiment sympathiques. Merci Jean, mon fidèle co-bureau. Merci d'avoir pris le temps de m'expliquer tous les nouveaux modèles génératifs, ta connaissance de la littérature est impressionnante. Merci à toute l'équipe IOP et notamment aux doctorants : Paul, Lucile, Nicoletta, Gautier, Pierre-Jean, Antoine, Marien. Je crois que les HC avec notre team du 2e étage finira bien par me manquer. Merci à toute l'équipe franco-italienne du labo. Avoir retrouvé un tel groupe d'amis en 3e année a été une vraie bouffée d'air frais. Ogni pranzo con voi è stato un momento unico e memorabile. En repensant à ce groupe, je pense aux tours de lac avec Béa, aux discussions course à pied avec Simon, aux tentatives de jeux de mots de Giorgia, aux rires d'Audrey, à la famille Gossard, au foot planifié chaque semaine avec Marco mais jamais fait, aux tapes dans le dos de Simone, aux cours de français de Camilla (Je mange, Tu mange, Il mange,

Nous mange, Vous mange, Ils mange), à la folie d'Emanu, au Français magnifique de Beppe, aux traductions de l'horoscope de Umberto, à la gentillesse de Bianca, au mangeur téméraire Julien, aux recommandations Youtube de Mathias. Mais surtout, je pense à crier : BRAVO Agathe !

J'aimerais maintenant avoir quelques mots pour ma famille. Pendant ces trois ans de thèse, on en a vécu des choses ! Merci papa et maman, pour votre soutien et votre amour inconditionnel. Papa, je suis très heureux que tu sois là en bonne santé pour m'écouter présenter ma thèse. Merci Maman pour ton dévouement et ta générosité. Merci à mes formidables frères et sœurs. Julien et Véro, les naissances de Santi et Camillo ont été les pics de bonheurs de ces trois années de thèse. Merci pour tous ces week-ends en Suisses si régénérants. Marianne, ta force de caractère et tes choix de vie seront des exemples pour mes très prochaines grandes décisions. Louise, comme dit Maman, merci d'être le rayon de soleil de la famille.

Il est maintenant temps de remercier mes amis. Malgré le temps qui passe, j'ai eu la chance de garder plusieurs groupes d'amis, ils sont si importants pour mon équilibre. D'abord les fameux Amis Historiques. Je suis fier, qu'après toutes ces années, nous soyons encore si proches. J'espère encore danser de nombreux Jean Petit. Philou, merci de rester toi-même et par la même occasion, de m'aider à rester "simple". Alex et Hugo, merci pour votre accueil toujours chaleureux dès que je monte à Paris. Sasa, ma nouvelle amie de campagne, hâte à notre prochaine activité rurale. Merci Thibaut, Thibault, Caroline, Thomas pour tous ces moments de rires et de bonne humeur. Merci Clémence pour ton soutien. Solène, tu étais un rayon de soleil à chaque fois que je rentrais en Bretagne.

Ensuite, bien étendu, les gaziers de prépa. Alors que j'étais en pleine rédaction de la thèse, la semaine de vacances cet été avec vous a été une déconnexion incroyable. La Brute et le Truand, Alamich, Beud, Carayol, Cam, Coach, Fr6, Gandhi, Gautaf, Grutt, Guéno, Jacky, Kalamar, Keut, Lisa, Marie, Marèène, Mathil, Tang. Ce groupe est toujours plus grand, mais aussi toujours plus fort, grâce à ce que chacun apporte et représente.

Après avoir remercié les "vieux" amis, il faut maintenant que je remercie les nouveaux ! Merci à tous ceux qui ont fait de ma vie à Bordeaux une magnifique période. Malheureusement, il est fort probable que je quitte à présent cette ville. Alors que mon arrivée au milieu du 2e confinement a été assez compliquée à vivre, je repars vraiment à reculons. C'est "Bienvenue chez les Chti's" à Bordeaux en fait (comment on pourrait l'appeler, "Bienvenue chez les bobos" ?). Je tiens à sincèrement remercier tous mes amis que j'ai rencontrés ici. D'abord Marine, Théo, Adrien, et plus récemment le "Nouveau Groupe" comme dit WhatsApp. Merci Sim, ma coloc de coeur, pour nous avoir tous rapproché. Bon courage Gaspard pour la saison prochaine en L2. Mathieu, le boss de la raquette, à présent que je ne suis plus au niveau, je m'accroche à ce (de plus en plus) vieux souvenir de victoire. Hamza et Fred, merci pour votre bonne humeur. Merci à mes colocs François et Juliette, ce fut un vrai plaisir de vivre avec vous ces derniers mois. Si vous cherchez un prof de skate pour votre enfant, le meilleur vit à 30 rue Doumerc. Vous y trouverez aussi une illustratrice hors pair qui customisera le skate du môme. Enfin, merci à la communauté Adidas Runners. En intégrant ce groupe de running, j'ai trouvé une nouvelle passion, dans lequel j'ai pu m'investir de plus en plus chaque semaine. Thomas, Fabien, Marine, Laura et tous les autres, vous retrouver à chaque entraînement est un vrai plaisir. Merci Inès. Tu m'as donnée une réelle confiance en moi dans ce sport. J'espère continuer à croiser ton sourire le plus souvent possible.

Finalemente, Patri, me gustaría dedicarte unas palabras. Durante esta tesis, estabas el pilar en el que podía apoyarme. Siempre pense en ti y en tu trayectoria para orientar mis objetivos y decisiones. Es gracias a ti que me fasciné por esta comunidad de investigadores.

Aunque ahora vivamos lejos el uno del otro, al moment de hacer un balance de estos tres años, obviamente pienso primero en ti. Gracias por todo.

Résumé substantiel en français

L'objectif de cette thèse est de résoudre des problèmes inverses en imagerie en utilisant des régularisations explicites et non-convexes apprises par réseaux de neurones profonds. Nous exploiterons en particulier des algorithmes itératifs dotés de propriétés théoriques de convergence bien établies.

Nous considérons un modèle d'observation général de la forme $y = N(A(x^*))$ avec $y \in R^m$ l'image dégradée observée, $x^* \in \mathbb{R}^n$ l'image propre que nous voulons approcher, $A : \mathbb{R}^n \to \mathbb{R}^m$ l'opérateur de dégradation et $N : \mathbb{R}^m \to \mathbb{R}^m$ un opérateur stochastique qui représente le bruit de mesure. Le cas le plus courant étant A linéaire et $N(x) \sim \mathcal{N}(x, \nu^2 \operatorname{Id})$ un bruit Gaussien. Ce problème est classiquement abordé en résolvant un problème d'optimisation de la forme suivante

$$x^* \in \underset{x \in \mathbb{R}^n}{\arg\min}\, \lambda f(x) + g(x) \tag{1}$$

où f est un terme d'attache à l'observation dégradée, g un terme de régularisation et $\lambda > 0$ un paramètre de régularisation qui pondère l'importance des deux termes. Le terme de régularisation permet d'inclure de l'information a priori sur la régularité de la solution désirée. Un problème de longue date consiste à concevoir des fonctions g qui reflètent une régularité pertinente sur x, tout en développant des schémas numériques efficaces pour résoudre l'équation (2.2). Les premiers modèles d'images ont été développés avec des régularisations explicites et convexes, par exemple la norme L^2 ou la variation totale, afin de résoudre le problème (2.2) avec des algorithmes itératifs aux propriétés de convergence bien connues. Plus récemment, le deep learning a révolutionné le domaine de l'imagerie avec différents modèles de réseaux de neurones qui ont produit des résultats remarquables dans diverses applications. Il demeure un défi de déterminer comment incorporer au mieux des modèles de deep learning dans une fonction de régularisation g qui présente les propriétés nécessaires pour permettre l'utilisation de techniques d'optimisation établies.

La régularisation "plug-and-play".

La minimisation (2.2) est généralement effectuée en utilisant des algorithmes d'optimisation du premier ordre, dits proximaux. Ces algorithmes opèrent individuellement sur les deux termes de (2.2) via deux opérateurs : l'opérateur de descente de gradient $\operatorname{Id} - \tau \nabla f$ et l'opérateur proximal $\operatorname{Prox}_{\tau f}$. Par exemple, l'algorithme de descente de gradient proximale (PGD) alterne entre une opération proximale et une étape de descente de gradient sur l'une ou l'autre des fonctions f et g :

$$\text{(PGD)} \quad x_{k+1} = \operatorname{Prox}_{\tau g} \circ (\operatorname{Id} - \tau \lambda \nabla f)(x_k) \text{ ou } x_{k+1} = \operatorname{Prox}_{\tau \lambda f} \circ (\operatorname{Id} - \tau \nabla g)(x_k) \tag{2}$$

Les algorithmes HQS, ADMM ou Douglas-Rachford Splitting (DRS) sont d'autres exemples d'algorithmes proximaux.

Les méthodes Plug-and-Play établissent une connexion élégante entre ces algorithmes d'optimisation et les modèles de réseaux de neurones profonds via l'introduction d'un débruiteur d'image D_σ. Leur dérivation est basée sur l'analyse théorique des débruiteurs optimaux MAP et MMSE. Étant donné un débruiteur générique D_σ, par exemple un réseau de neurone profond, préalablement entraîné ou construit pour débruiter une image dégradée par bruit Gaussien d'écart type σ, les algorithmes Plug-and-Play sont construits en utilisant D_σ en lieu et place d'un opérateur de descente sur le terme de régularisation g :

(i) pour les algorithmes RED, le débruiteur remplace l'opérateur de descente de gradient : $D_\sigma = \mathrm{Id} - \nabla g$.

(ii) pour les algorithmes PnP, le débruiteur remplace l'opérateur proximal : $D_\sigma = \mathrm{Prox}_{\tau g}$.

Par exemple, l'algorithme PGD (2) devient avec cet échange

$$\text{(PnP-PGD)} \quad x_{k+1} = D_\sigma \circ (\mathrm{Id} - \tau\lambda\nabla f)(x_k) \tag{3}$$

$$\text{ou (RED-PGD)} \quad x_{k+1} = \mathrm{Prox}_{\tau\lambda f} \circ (\mathrm{Id} - \tau(\mathrm{Id} - D_\sigma))(x_k) \tag{4}$$

Des résultats de pointe pour divers problèmes inverses ont été obtenus avec ces procédés. Un avantage significatif de ces méthodes est qu'elles sont non supervisées et qu'un simple débruiteur Gaussien peut-être utilisé pour restaurer une variété de problèmes inverses. En particulier, le choix du débruiteur et découplé du modèle de dégradation et donc du terme d'attache aux données f.

Cependant, les algorithmes Plug-and-Play sont utilisées avec peu de garanties théoriques de convergence. En effet, comme un débruiteur générique ne s'exprime généralement pas exactement comme un opérateur proximal ou un gradient, les résultats de convergence ne suivent pas facilement. La plupart des résultats de convergence qui existent limitent leur analyse aux termes d'attache aux données fortement convexes, ce qui exclue de très nombreux problèmes, ou considèrent des débruiteurs non expansifs. Il a été déjà démontré que contraindre un débruiteur à être non expansif provoque une baisse considérable de performance. De plus, avec l'utilisation d'un débruiteur générique, la régularisation est seulement implicitement incluse via l'opération de débruitage. Ainsi, les algorithmes plug-and-play ne minimisent pas de fonctionnelle explicite de la forme (1). Cela limite fortement l'interprétabilité du résultat et le contrôle numérique de l'algorithme.

Débruiteur *Gradient-Step* pour la convergence des algorithmes RED (Chapitre 4)

Nous proposons d'utiliser un débruiteur appelé *Gradient-Step* (GS) qui s'écrit exactement comme une étape de descente de gradient sur un potentiel $g_\sigma : \mathbb{R}^n \to \mathbb{R}$ différentiable et paramétré par un réseau de neurone profond

$$D_\sigma = \mathrm{Id} - \nabla g_\sigma \tag{5}$$

Nous choisissons en particulier g_σ sous la forme

$$g_\sigma(x) = \frac{1}{2} ||x - N_\sigma(x)||^2 \tag{6}$$

avec $N_\sigma : \mathbb{R}^n \to \mathbb{R}^n$ un réseau de neurones de classe C^1. Grâce à cette écriture, nous pouvons paramétrer N_σ avec n'importe quelle architecture de réseau de neurones différentiable qui s'est montrée efficace pour le débruitage d'image, par exemple avec un UNet. Malgré le fait qu'il soit contraint d'être un champ de gradients, D_σ obtient des résultats de débruitage à l'état de l'art, et performe aussi bien que le même réseau UNet non contraint.

D_σ est appris pour débruiter simultanément à différents niveaux de bruits σ par minimisation du coût L^2 de débruitage

$$\mathbb{E}_{X\sim p_X, \xi_\sigma \sim \mathcal{N}(0,\sigma^2 I)} \left[||D_\sigma(x + \xi_\sigma) - x||^2 \right] \tag{7}$$

En utilisant le résultat de *Denoising Score Matching* (Vincent, 2011), il est alors prouvé que g_σ approche alors le log d'une version lissée (par convolution avec une Gaussienne d'écart type σ) de la vraie distribution p_X des images propres. Cela fait de g_σ un bon candidat pour régulariser un problème inverse.

Nous considérons deux algorithmes RED (RED-GD et RED-PGD) qui sont initialement construits à partir respectivement des algorithmes de descente de gradient (GD) et de gradient proximal (PGD). Avec le débruiteur GS (5), les algorithmes RED (appelés *GSRED*) prennent à nouveau la forme de véritables algorithmes d'optimisation pour optimiser $\lambda f + g_\sigma$. La fonction g_σ étant non-convexe, la preuve de la convergence des algorithmes GS-RED s'appuie alors sur les résultats connus de convergence en optimisation non-convexe. Ces résultats sont préalablement étudiés au Chapitre 3. Ils s'appuient notamment sur la propriété Kurdyka-Łojasiewicz (KŁ) que doit vérifier la fonction objectif. Après vérification des hypothèses de convergence, nous démontrons que RED-GD et RED-PGD, avec le débruiteur GS (5), convergent vers un point stationnaire de $\lambda f + g_\sigma$. Étant donné que la constante de Lipschitz du gradient de g_σ n'est pas explicite, le pas de temps de l'algorithme est automatiquement règlé par *backtracking*.

Les expériences de défloutage, super-résolution, inpainting réalisées confirment les résultats de convergence théoriques et montrent que les algorithmes GSRED atteignent des performances au niveau de l'état de l'art.

Débruiteur proximal pour la convergence des algorithmes PnP (Chapitre 5)

Afin de prouver la convergence des méthodes PnP, nous proposons un second débruiteur entraîné pour prendre exactement la forme d'un opérateur proximal. Pour cela, nous démontrons que, pour g_σ à gradient L-Lipschitz avec $L < 1$, le débruiteur *Gradient-Step* (5) est un opérateur proximal, c'est-à-dire

$$D_\sigma = \mathrm{Id} - \nabla g_\sigma = \mathrm{Prox}_{\psi_\sigma} \tag{8}$$

pour une certaine fonction $\psi_\sigma : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$. De plus, ψ_σ est $M = \frac{L}{L+1}$-faiblement convexe. Afin de contraindre ∇g_σ à être contractant, nous proposons de régulariser le coût d'apprentissage (7) avec un terme de pénalisation sur la norme spectrale de la Hessienne de g_σ. Ce coût s'écrit alors

$$\mathbb{E}_{x\sim p_X, \xi_\sigma \sim \mathcal{N}(0,\sigma^2)} \Big[||D_\sigma(x+\xi_\sigma) - x||^2 + \mu \max(\big|\big|\nabla^2 g_\sigma(x+\xi_\sigma)\big|\big|_S, 1-\epsilon) \Big] \tag{9}$$

Le débruiteur obtenu satisfait empiriquement la contrainte $L < 1$ tout en conservant de bonnes performances de débruitage.

Avec ce débruiteur proximal, les algorithmes PnP (appelés *ProxPnP*) prennent à nouveau la forme de véritables algorithmes proximaux pour optimiser $\lambda f + \phi_\sigma$. La fonction de régularisation ϕ_σ étant faiblement convexe, la preuve de la convergence des algorithmes ProxPnP s'appuie sur les résultats de convergence des algorithmes proximaux pour minimiser la somme de deux fonctions dont une est faiblement convexe. Nous montrons ainsi que les algorithmes ProxPnP-PGD et ProxPnP-DRS sont garantis de converger vers des points stationnaires de la fonction objective $\lambda f + \phi_\sigma$.

Avec la propriété précédente, D_σ s'écrit comme $\mathrm{Prox}_{\psi_\sigma}$ et non pas comme $\mathrm{Prox}_{\tau\psi_\sigma}$ (avec un pas de temps $\tau > 0$) comme utilisé dans les algorithmes proximaux (2). Ainsi, nous sommes contraints d'avoir un pas de temps fixé à $\tau = 1$. La convergence des algorithmes proximaux étudiés est conditionnée à une contrainte sur le pas de temps τ qui dépend

du paramètre de régularisation λ du problème. Ainsi, lorsque ProxPnP utilise un pas de temps $\tau = 1$, la contrainte se transforme en une limite sur la valeur du paramètre λ. Cela devient problématique lorsqu'il s'agit de restaurer une image qui présente des dégradations mineures. Dans de tels cas, on s'attendrait en effet à obtenir des solutions appropriées en utilisant une faible valeur de λ.

Nous proposons alors une relaxation de l'algorithme de descente de gradient proximal, appelée αPGD, dérivée de l'algorithme Primal Dual en version Bregman (Chambolle and Pock, 2016). Quand la contrainte de convergence de l'algorithme PGD restreint la somme de la constante de Lipschitz de ∇f et de la constante de convexité faible M de ϕ_σ, l'algorithme αPGD, quant à lui, limite le produit de ces deux termes. Ainsi, lorsque M tend vers 0, nous avons la liberté de choisir λ aussi petit que souhaité. Nous proposons en parallèle une manière de contrôler la constante de convexité faible M de ϕ_σ.

Les expériences de défloutage et de super-résolution réalisées démontrent l'efficacité de nos algorithmes ProxPnP. En particulier, Prox-αPGD remédie à la perte de performance notable de ProxPnP-PGD attribuable à la restriction sur la valeur du paramètre λ.

Généralisation Bregman des algorithmes Plug-and-Play pour la résolution de problèmes inverses de Poisson (Chapitre 6)

Dans les expériences précédentes, le bruit de mesure sur l'observation y était supposé Gaussien. Pour de nombreuses applications, par exemple en microscopie ou en astronomie, il est plus exact d'utiliser un modèle de bruit de Poisson $y \sim \mathcal{P}(Ax)$. Le terme d'attache aux données f adéquat est alors la divergence de Kullback-Leibler

$$f(x) = \sum_{i=1}^{m} y_i \log\left(\frac{y_i}{\alpha(Ax)_i}\right) + \alpha(Ax)_i - y_i. \tag{10}$$

Cependant, il est important de noter que ce terme ne possède ni un gradient Lipschitz, ni un opérateur proximal explicite, ce qui restreint l'application d'algorithmes proximaux pour la minimisation d'un objectif variationnel de la forme (1).

Pour résoudre cette problématique, une approche possible consiste à se placer dans un cadre plus général où la distance Euclidienne est remplacée par une divergence de Bregman $D_h(x, y)$. L'algorithme PGD est alors remplacé par sa version Bregman appelée BPG(Bauschke et al., 2017)

$$x^{k+1} \in \mathrm{Prox}^h_{\tau g} \circ \nabla h^*(\nabla h - \tau\lambda\nabla f)(x_k). \tag{11}$$

où Prox^h est un opérateur proximal de Bregman. L'avantage de l'algorithme BPG est que la condition ∇f L_f-Lipschitz nécessaire pour la convergence de PGD est remplacée par une condition *NoLip* :

Il existe un potentiel strictement convexe $h : \mathbb{R}^n \to \mathbb{R}$ tel que $L_f h - f$ est convexe.

Le terme d'attache aux données (10) satisfait cette condition pour h l'entropie de Burg $h(x) = \sum_{i=1}^{n} -\log x_i$.

Nous utilisons cet algorithme BPG pour dériver une version Bregman des algorithmes RED et PnP. Pour cela, nous introduisons un nouveau modèle de bruit de Bregman, qui généralise le modèle de bruit Gaussien et qui est défini, pour un paramètre γ, par la loi postérieure

$$p_{Y|X}(y|x) = \alpha(x) \exp\left(-\gamma D_h(x, y)\right). \tag{12}$$

Nos algorithmes Bregman PnP et RED sont dérivés en suivant le même raisonnement que dans le cas Euclidien, mais pour ce nouveau modèle de bruit. Ainsi, nous étudions d'abord les débruiteurs théoriques MAP et MMSE. Nous montrons que le débruiteur MAP s'écrit comme un *opérateur proximal de Bregman* et que la formule de Tweedie se généralise pour écrire le débruiteur MMSE en fonction du gradient du log prior. Maintenant, étant donné un débruiteur générique $\mathcal{B}_\gamma$, préalablement appris pour débruiter une image dégradée par bruit Bregman de paramètre γ, les algorithmes Bregman PnP (B-PnP) et RED (B-RED) sont construits en introduisant le débruiteur $\mathcal{B}_\gamma$ pour remplacer, dans BPG (11), respectivement l'opérateur proximal de Bregman $\mathrm{Prox}^h_{\tau g}$ et le gradient ∇g.

Dans ce contexte, la régularisation plug-and-play et le terme d'attache aux données sont donc liés par le choix du potentiel de Bregman h. Nous apportons donc une limite à l'argument de découplage entre ces deux termes, souvent mis en avant dans la littérature plug-and-play.

Ensuite, nous généralisons nos analyses de convergences précédemment proposées dans le contexte Euclidien. Nous proposons une généralisation Bregman du débruiteur *Gradient-Step*, qui s'écrit

$$\mathcal{B}_\gamma(y) = y - (\nabla^2 h(y))^{-1} \cdot \nabla g_\gamma(y), \tag{13}$$

avec $g_\gamma : \mathbb{R}^n \to \mathbb{R}$ un potentiel nonconvex paramétré par réseau de neurone. Équipé ce débruiteur, nous démontrons que l'algorithme B-RED converge vers un point stationnaire de la fonction $\lambda f + g_\gamma$.

De plus, après avoir prouvé un nouveau résultat de caractérisation des opérateurs proximaux Bregman, nous montrons que, si $\mathcal{B}_\gamma$ à une Jacobienne définie positive, alors il s'écrit comme un opérateur proximal de Bregman $\mathcal{B}_\gamma(y) \in \mathrm{Prox}^h_{\phi_\sigma}$ pour ϕ_γ une fonction non-convexe. C'est la généralisation Bregman du débruiteur proximal (8). La convergence de l'algorithme B-PnP, avec ce débruiteur, vers un point stationnaire de la fonction $\lambda f + \phi_\gamma$ est alors établie.

Nous appliquons nos algorithmes dans le contexte des problèmes inverses de Poisson avec h l'entropie de Burg. Le modèle de bruit Bregman suit alors une loi inverse gamma multivariée. Nous montrons que le débruiteur (13) appris sur ce modèle de bruit permet d'efficacement régulariser un problème inverse de Poisson.

Contents

Chapter 1

General Introduction

Contents

1.1 Context

Image inverse problems are the core of various scientific and technological disciplines where a digital visual representation is involved, be it a photograph, a medical scan, a satellite or microscope image. In all of these disciplines, a physical device acquires a real-world 2D scene and renders a digital representation. While a camera directly captures image pixels, a CT scanner renders a sinogram and a radio interferometer (for astronomical images) collects spatial frequencies (see Figure 1.1). The image inverse problem consists in reversing this acquisition, that is to say, in looking for a realistic input image that would best explain the observation. It is a difficult problem due to the inherent imperfections and physical limitations of the acquisition process. Essentially, observation devices are recording an altered or distorted version of the original scene. These imperfections can be caused by different factors. First, any physical acquisition device has limited resolution: a photo camera has a limited number of pixels, a CT scanner sends X-rays at a limited number of angles and a radio interferometer collects a finite number of frequencies. Also, as a by-product of image capture, any sensor creates a certain amount of noise, of different kind, on the digital image. The acquisition can also be altered for various additional reasons. For instance, a photo can be blurred due to camera shake or object motion during its acquisition, or the recorded scene can be partially occluded.

Before going further, we need to mention that a grayscale digital image consists of a matrix filled with $L \times H$ pixels. Each pixel has a real value between 0 and 1, 0 corresponding to a black pixel and 1 corresponding to a white pixel. Color images are represented by a stack of three of such matrices representing respectively the level of red R, green G and blue B. In practice, we will consider that a color image x is simply a vector of $\mathbb{R}^n$ *i.e.* a list of $n = 3LH$ real values $x = (x_1, ..., x_n)$. It is built, for example, by listing the pixel values one by one in the lexicographical order. From this representation, we can employ mathematical tools and operations for describing and solving imaging problems.

Solving an image inverse problem consists in developing techniques for reversing the

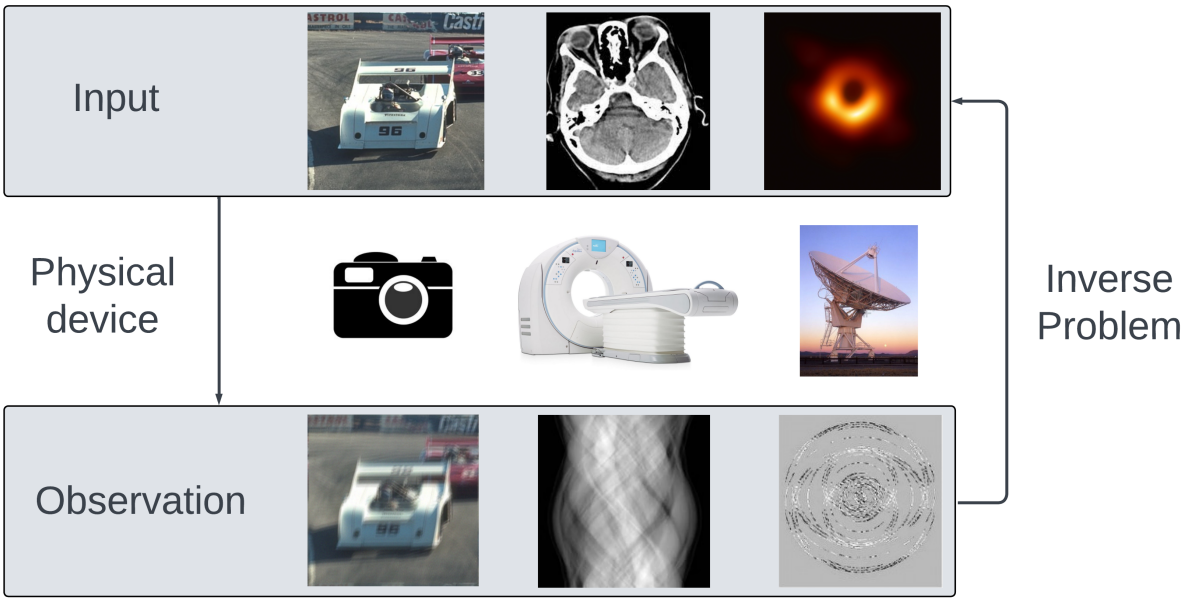


Figure 1.1: Image inverse problems

forward observation model and recovering the original scene as faithfully as possible. The degradation that takes place during the acquisition process results in data loss, and therefore, restoring the ideal image requires incorporating prior knowledge of what the ideal image should resemble.

Traditional approaches reformulate image inverse problems as optimization problems. They consider the clean image as the solution of the minimization of a well-chosen function. This function typically writes as the sum of two terms. The first one, called data-fidelity term, measures the dissimilarity between the estimation and the observation while the second one, called regularization, ensures that the output satisfies some prior knowledge about the regularity of the ideal output. Examples of typical regularization include Total Variation (TV) which encourages smoothness within the reconstructed image, or the L^1 norm within a specific basis or dictionary, which promotes sparsity in the representation. Upon minimization, the result should align with the observation while adhering to the requirements brought by the prior information. This optimization problem is addressed through iterative optimization techniques. These iterations rely on two fundamental mathematical operations: the gradient and the proximity operator. The convergence of these algorithms has been well established for convex and nonconvex objectives.

More recently, with the advent of deep learning, deep neural networks have outperformed classical methods in a variety of imaging problems. These networks are directly trained to realize the image inversion thanks to very large datasets. More precisely, when we have access to a substantial dataset of ground-truth pairs (input, observation), it becomes possible to train a neural network to approximate the optimal mapping from observation to input. Prior information is then automatically derived from the data and implicitly embedded within the learned mapping. Despite being efficient and fast, these methods come with several drawbacks. First, they require a large amount of data (thousands of images) which can be expensive, time-consuming, or even impossible to get for some inverse problems. It is also necessary to retrain the model whenever the degradation model changes. For instance, one cannot use a model trained on CT scans of a brain for acquiring other parts of the body. Moreover, deep learning models lack of interpretability. It is

difficult to understand the rationale behind a specific output. This can pose a significant limitation, especially when the output is utilized in human decision-making processes, as it is typically the case for medical images.

Plug-and-play algorithms (Venkatakrishnan et al., 2013; Romano et al., 2017) were built to take advantage from both strategies. They are iterative algorithms where the iterated operator is constructed using a deep neural network. Compared to classical methods, this deep neural network replaces either the gradient (RED algorithms) or the proximity operator (PnP algorithms) of the regularization function. Therefore, it should implicitly bring prior information. This deep neural network is previously trained to denoise images corrupted with artificial Gaussian noise. Plug-and-play methods are unsupervised and have shown state-of-the-art visual performance for various image restoration problems. However, when incorporating an off-the-shelf generic denoiser within an iterative minimizing scheme, one a priori loses its guarantees of convergence as well as its interpretability. Indeed, plug-and-play methods do not minimize any explicit function. To ensure convergence of plug-and-play schemes, many works have proposed specific parametrizations of the plugged deep denoisers. However, existing results require either unverifiable or suboptimal hypotheses, or assume restrictive conditions on the parameters of the inverse problem.

1.2 Contributions and Outline

In this work, we propose new types of plug-and-play methods in which the denoiser is realized as a gradient descent step or as a proximal step on a function parameterized by a deep neural network. Exploiting convergence results for optimization algorithms in the nonconvex setting, we show that the proposed algorithms are convergent iterative schemes that target stationary points of explicit global functionals.

In Chapter 2, we start by introducing more formally image inverse problems, convex and nonconvex optimization and plug-and-play methods. We will make a distinction between two kinds of plug-and-play algorithms that we will refer to PnP and RED, according to the fact that the plugged denoiser approximates the MAP or MMSE estimator.

In Chapter 3, we present a variety of tools and results that will be the building blocks for analyzing the convergence of plug-and-play algorithms. In particular, we give convex and nonconvex convergence results for the most common proximal optimization algorithms (PGD, DRS, ADMM or Primal-Dual). We also review in this chapter the existing literature on plug-and-play convergence.

Our main contributions are contained in Chapters 4, 5 and 6.

In Chapter 4, we introduce a new deep denoiser, called Gradient-Step denoiser, which writes as a gradient descent step on an explicit and nonconvex function parameterized by a deep neural network. We show that it is possible to learn such a deep denoiser while not compromising its performance. We prove that the proposed denoiser ensures convergence of the RED algorithms (RED-PGD (2.102) and RED-GD (2.102)) towards a critical point of an explicit objective function. Moreover, compared to other plug-and-play methods, the resulting RED algorithms reach state-of-the-art restoration performance on deblurring and super-resolution problems.

In Chapter 5, under additional constraints, we show that the Gradient-Step denoiser writes as a proximity operator of another nonconvex function. We then prove that PnP algorithms (*e.g.* PnP-PGD (2.109) and PnP-DRS (2.111)) also converge towards explicit stationary points. In this chapter, we additionally introduce a new optimization scheme, reminiscent of Proximal Gradient Descent, which converges for a wider ranger of

regularization parameters, thus allowing for a more accurate image restoration.

In Chapter 6, we extend our PnP and RED methods in a more general Bregman framework. Using a Bregman divergence instead of the Euclidean distance enables to better capture the smoothness properties of the inverse problem, which proves particularly useful when the observation is corrupted with Poisson noise. This extension is based on the definition of a Bregman noise model that generalizes Gaussian noise. We propose a denoiser that extends the Gradient-Step denoiser to this Bregman geometry, and show that this denoiser can also write as a Bregman proximity operator. After proving new convergence results for the Bregman Proximal Gradient (BPG) algorithm in the nonconvex setting, we demonstrate that our Bregman PnP and RED methods converge towards explicit stationary points. Eventually, we experimentally verify the efficiency of the approach for Poisson image deblurring.

1.3 List of publications

Hurault S., Leclaire A., Papadakis N.
Gradient step denoiser for convergent plug-and-play.
In *International Conference on Learning Representations (ICLR). (2022)*

Hurault S., Leclaire A., Papadakis N. (2022)
Proximal denoiser for convergent plug-and-play optimization with nonconvex regularization.
In *International Conference on Machine Learning (ICML).* (2022)

Ballester C., Bugeau A., Hurault S., Parisotto S., Vitoria P.
An Analysis of Generative Methods for Multiple-Image Inpainting.
In *Handbook of Mathematical Models and Algorithms in Computer Vision and Imaging, 1-48. (2022)*

Hurault S., Chambolle A., Leclaire A., Papadakis N.
A relaxed proximal gradient descent algorithm for convergent plug-and-play with proximal denoiser.
In *International Conference on Scale Space and Variational Methods in Computer Vision (SSVM).* (2023)

Hurault S., Kamilov U., Leclaire A., Papadakis N.
Convergent Bregman Plug-and-Play Image Restoration for Poisson Inverse Problems.
In *Neural Information Processing Systems (Neurips).* (2023)

Li L., Hurault S., Salomon J.
Self-consistent velocity matching of probability flows.
In *Neural Information Processing Systems (Neurips).* (2023)

Chapter 2

Mathematical introduction to Plug-and-Play Image Restoration

Contents

2.1 Variational image restoration

Image restoration (IR) consists in estimating an unknown image (or signal) $x \in \mathcal{X} \subseteq \mathbb{R}^n$ given an observation $y \in \mathcal{Y} \subseteq \mathbb{R}^m$ which has been degraded (*e.g.* with blur, noise, sampling). The *forward model* is the map used to represent the physics behind this degradation. From the clean signal $x \in \mathcal{X}$, the forward map usually takes the form

$$y = N(A(x)) \tag{2.1}$$

where $A : \mathcal{X} \to \mathcal{Y}$ is deterministic and $N : \mathcal{Y} \to \mathcal{Y}$ is stochastic and characterizes the noise affecting the measurements. Generally, and for the rest of this manuscript, A is assumed linear. For example, for image inpainting, A is diagonal with binary values, and for image deblurring, A represents the convolution with a blur kernel. More complex degradations (*e.g.* image haze removal, single photon lidar) could nonetheless involve a non-linear operator A . The most typical noise model is additive white Gaussian noise *i.e.* $N(x) = x + \xi$ with ξ a realization of $\mathcal{N}(0, \sigma^2 \,\mathrm{Id})$ a Gaussian distribution with zero mean and standard deviation σ.

The image inverse problem consists in estimating $x \in \mathcal{X}$ given the observation y, the forward operator A and the noise model N. An inverse problem is ill-posed in the sense of Hadamard if it does not have a unique solution or if the solution does not change continuously with respect to the input. This is typically the case if A is not invertible or if its eigenvalues are too small in absolute value, which makes the noise explode in the pseudo-inverse estimation. In order to address ill-posedness, a typical approach is to incorporate regularization by reformulating the image inverse problem as a minimization problem of the form

$$\hat{x} = \arg\min_x f(x) + \lambda g(x) \tag{2.2}$$

where $f : \mathcal{X} \to \mathbb{R}$, called data-fidelity term, measures the distance to a degraded observation y, for example the L^2 distance $f(x) = \frac{1}{2} ||A(x) - y||^2$, and $g : \mathcal{X} \to \mathbb{R}$ is a regularization term weighted by a parameter $\lambda > 0$.

Including a regularization term g can permit to cope with the ill-posed nature of the inverse problem by assuming a priori knowledge about the ground-truth solution, and thus restricting the null space of the observation model A. The choice of the regularization depends on the properties we wish to impose on the solution. For instance, one can use the L^1 norm $g(x) = ||x||_1$ to enforce sparsity, or the Total Variation $g(x) = ||Dx||_1$ for promoting spacial smoothness. More details on the choices of the regularization and data-fidelity terms will be given in Sections 2.1.1 and 2.1.2.

This variational formulation can also be obtained via a *Bayesian approach*. Assume that x is the realization of a random variable X with prior distribution $p_X(x)$: we assume a prior distribution on $(\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n))$ that admits a density w.r.t. to the Lebesgue measure on $\mathbb{R}^n$. This prior depends on the type of image data considered. In the general case, it represents the distribution of clean natural images, but it could also represent a more specific kind of signal, for example medical or astronomical data. Let y be the realization of Y with forward conditional distribution $p_{Y|X}(y|x) = p_N(y|A(x))$ with p_N the noise distribution. The Maximum A Posteriori (MAP) estimation maximizes the posterior probability distribution $p_{X|Y}(x|y)$

$$\hat{x}^{MAP} = \arg\max_{x \in \mathcal{X}} p_{X|Y}(x|y). \tag{2.3}$$

By Bayes formula, the posterior distribution is given by (with $p_Y(y)$ the marginal distribution of Y)

$$p_{X|Y}(x|y) = \frac{p_X(x) p_{Y|X}(y|x)}{p_Y(y)} \tag{2.4}$$

and the MAP writes as the minimization of the sum of two functions

$$\hat{x}^{MAP} = \arg\min_{x \in \mathcal{X}} - \log p_{Y|X}(y|x) - \log p_X(x). \tag{2.5}$$

Compared to (2.2), the MAP estimator relates the data-fidelity term to the conditional distribution $p_{Y|X}$ and the regularization term to the prior distribution p_X. For example, assuming Gaussian noise $p_N(y|x) \propto \exp\left(-\frac{||x-y||^2}{2\sigma^2}\right)$, we obtain again the L^2 data-fidelity. Moreover, assuming a Laplacian image prior $p_X(x) \propto \exp\left(-\lambda ||x||_1\right)$, we get back the L^1 regularizer. However, as advocated by Gribonval and Nikolova (2021), although the variational formulation (2.2) can be derived via the MAP estimator, a variational formulation is *not necessarily* linked to a Bayesian formulation of the form (2.5). For instance, the L^1 regularization can be chosen to promote sparsity, but it does not mean that the Laplacian prior is a good distribution to represent the true image prior p_X.

Once the image restoration task has been formulated as a minimization problem of the form (2.2), one needs to choose an appropriate *optimization algorithm* to minimize the sum of two functions. Typically, it consists in picking the most suitable algorithm in a list of well-known iterative schemes, for example Gradient Descent (GD), Proximal Gradient Descent (PGD) or Alternating Direction Method of Multipliers (ADMM). The choice of the minimization algorithm mostly depends on the regularity of the functions f and g, which may or may not fit its conditions required for convergence. Therefore, before proceeding, one must need to precisely know which are the minimal conditions for convergence of the iterative schemes of interest. These convergence results will be recalled in Section 2.3. In practice, assuming that the forward model (*i.e.* the data term f) is given and fixed, in order to achieve *performant image restoration with convergence guarantees*, the choice of both the regularization function g and the minimization scheme has to be done jointly. For example, as we will see later, minimization algorithms have stronger convergence properties when applied to *convex* functions, which advocates for choosing convex regularizers. On the other hand, very efficient regularizers may be *nonconvex*. Actually, under the general MAP formulation (2.5), the regularizer is the true (but intractable) negative log image prior $-\log p_X(x)$ which is potentially highly nonconvex.

2.1.1 A variety of image priors

Classical priors Regularization is crucial since it tackles the ill-posedness of the IR task by bringing a priori knowledge on the solution. A long-standing problem consists in designing explicit prior functions g that reflect a relevant regularity prior on x while allowing for efficient numerical schemes to solve (2.2). Following the variational approach, research has first been dedicated to find, by hand, potentials g that are minimized when a desired property of a clean image is satisfied. Among the most classical priors, one can single out total variation Rudin et al. (1992), L^2 norm (Tikhonov) Tikhonov (1963), or sparsity in a given dictionary, for example in the wavelet representation Mallat (2009). Note that the choice of an appropriate regularizing potential often depends on the kind of degradation that is treated or on the type of structures we wish to recover in the image. For instance, Total Variation (with MAP estimation) favors piecewise constant regions at the cost of loosing textures details. Designing a prior g that is efficient for a variety of images and degradations is a difficult task. With the advent of big data, a more recent line of work tries to learn g from a dataset of "clean images" $\{x_i\}$ which can be understood as realizations of X.

Among learned priors, from the idea that a clean image should be sparse in a given dictionary $D \in \mathbb{R}^{n\times p}$, the first strategies consisted in learning these sparsity-promoting dictionaries (Mallat and Zhang, 1993; Aharon et al., 2006; Mairal et al., 2009, 2008). This

can be formulated as the following optimization problem

$$(\hat{D}, \hat{r}) = \underset{D \in \mathcal{C}, r \in \mathbb{R}^p}{\arg\min} \sum_i l(c_i, Dr_i) + \lambda ||r||_0 \tag{2.6}$$

where $l : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is some loss function (*e.g.* the L^2 or L^1 norm) and $\mathcal{C}$ is a constraint set for D (*e.g.* with columns $||d_j|| \leq 1$ (Olshausen and Field, 1997)). Alternatively, the nonconvex term $||r||_0$ can be replaced by the convex (and also sparsifying) $||r||_1$ norm to make the minimization problem (2.6) convex in D and r (Tibshirani, 1996; Chen et al., 2001).

A more general prior learning strategy consists in trying to approximate the true image prior p_X (or its log) with a parametric potential. The main advantage is that, in theory, following the MAP formulation (2.5), such a learned prior can be used to efficiently regularize any image inverse problem represented by $f(x) = -\log p_{Y|X}(y|x)$. *The prior and the observation model are then decoupled.* For $\mathcal{X}$ representing natural images, the distribution p_X is however very complex and irregular in a high-dimensional space $\mathbb{R}^n$.

Before the deep learning era, it was proposed with EPLL Zoran and Weiss (2011); Hurault et al. (2018) to simplify the learning problem by working on lower-dimensional image patches *i.e.* very small images (typically 8×8) extracted from the input image. From a large collection of patches ($\approx 10^6$) extracted from a dataset of natural images, and seen as realizations of a random variable denoted by X_p, the idea is to approximate the true patch prior p_{X_p} with $p^\theta_{X_p}$ parameterized by a set of parameters θ. The full image log prior $\log p_X$ in (2.5) is then approximated via summing the learned patch log prior values across all patches:

$$\log p_X(x) \approx \sum_{\text{patches } x_p^k \in x} \log p_{X_p}(x_p^k) \approx \sum_{\text{patches } x_p^k \in x} \log p^\theta_{X_p}(x_p^k) \tag{2.7}$$

In practice, $p^\theta_{X_p}$ is modeled in Zoran and Weiss (2011) as a Gaussian Mixture Model (GMM) with 200 mixture components. The parameters θ of the GMM (means, covariances, weights) are trained using the Expectation Maximization (EM) algorithm. EPLL is one of the state-of-the-art methods among non-deep image restoration methods, and compared to other efficient classical non-local methods such as BM3D, it has the advantage to provide an explicit prior and thus to be readily applicable for a variety of inverse problems formulated as (2.5). Even though very promising, the performance of the EPLL or dictionary learning algorithm has recently been largely overtaken by deep learning based priors. Indeed, compared to deep neural network architectures, the dictionary and GMM models have limiting representation power.

Deep priors More recently, deep generative models make use of recent deep learning advances to directly approximate the full prior with p^θ_X parameterized by a neural network. Among them, latent-based models propose to simplify the problem by introducing a latent variable Z in $\mathbb{R}^p$ with simple prior p_Z, typically $\mathcal{N}(0, \mathrm{Id})$. The parameters θ parameterize the posterior $p^\theta_{X|Z}$ via an image generator $G^\theta : \mathbb{R}^p \to \mathbb{R}^n$. For example, Generative Adversarial Networks (GAN) (Goodfellow et al., 2014) or Normalizing Flows (NF) Rezende and Mohamed (2015) models use the pushforward measure through G^θ denoted by $p^\theta_{X|Z} = G^\theta \# p_Z$ while Variational Auto-Encoders (VAE) Kingma and Welling (2013) use a Gaussian centered on $G^\theta(z)$ *i.e.* $p^\theta_{X|Z} = \mathcal{N}(G^\theta(z), \Sigma^\theta(z))$. In this case, the image prior

$$p^\theta_X(x) = \int_z p^\theta_{X|Z}(x|z) p_Z(z) dz \tag{2.8}$$

is however intractable. These models then resort to different strategies to optimize θ by realizing Maximum Likelihood Estimation (MLE) on the image data

$$\mathbb{E}_{x\sim X}\left[-\log p_X^\theta(x)\right]. \tag{2.9}$$

GANs make use of a discriminator neural network, NFs parameterize G^θ to be bijective and VAEs introduce an encoder neural network to approximate the intractable posterior $p_{Z|X}$ inside a variational inference formulation.

Once trained, this prior can be used to regularize inverse problems. Depending on the generative model, different variational formulations of the inverse problems have been proposed. Using normalizing flows, by bijectivity of G^θ, the image prior is tractable:

$$p_X^\theta(G^\theta(z)) = p_Z(z)|\det J_{G^\theta}(z)|^{-1}, \tag{2.10}$$

and the MAP problem (2.5) can be rewritten as an optimization problem with respect to the latent z (Helminger et al., 2020; Whang et al., 2020)

$$x^{MAP} = G^\theta\left(\arg\min_z -\log p_{Y|X}(y|G^\theta(z)) - \log p_X^\theta(G^\theta(z))\right). \tag{2.11}$$

Without bijectivity of G^θ, *i.e.* with a GAN or a VAE, directly integrating the generative prior inside the MAP (2.5) is difficult. Instead, a workaround proposed by Bora et al. (2017) consists in estimating a MAP w.r.t. the latent variable z

$$z^{MAP} = \arg\max_z p_{Z|Y}p(z|y) \tag{2.12}$$

and then choosing $x = G^\theta(z^{MAP})$. (2.12) can be shown to write as (see (Gonzalez et al., 2019, Appendix B.1))

$$z^{MAP} = \arg\max_z -\log p_{Y|X}(y|G^\theta(z)) - \log p_Z(z). \tag{2.13}$$

Note that the regularization term is different between (2.13) and (2.11). Both optimization problems are highly nonconvex due to the nonconvexity of G^θ. Moreover, the landscape of the learned p_X^θ is often irregular and can suffer from mode collapse, *i.e.* most of the mass is centered on a single kind of image. With such irregularities, it has been observed (Daras et al., 2021; Saharia et al., 2022) that iterative optimization applied to these objectives may not converge or easily fall in bad local minima.

Energy-Based Models (EBM) (Song and Kingma, 2021) directly consider a parametric probability density in the image space using a potential $E^\theta : \mathbb{R}^n \to \mathbb{R}$ via

$$p_X^\theta(x) = \frac{\exp(-E^\theta(x))}{Z^\theta}, \tag{2.14}$$

where Z^θ is a normalizing constant such that $\int p_X^\theta(x) = 1$. This constant is however intractable and training via maximizing of the log-likelihood w.r.t. θ is again impossible. Instead of trying to approximate this constant via MCMC, *Score Matching* (SM) consists in matching the score $\nabla \log p_X(x)$ with $\nabla \log p_X^\theta(x) = -\nabla E^\theta(x)$, which does not involve the intractable Z^θ. Equality of the scores implies equality of the (normalized) distributions p_X^θ

and p_X. Different score matching methods (Hyvärinen and Dayan, 2005; Vincent, 2011; Song et al., 2020b) have been proposed to minimize the intractable divergence

$$\mathbb{E}_{p_X}\left[\left|\left|\nabla \log p_X^\theta(x) - \nabla \log p_X(x)\right|\right|^2\right]. \tag{2.15}$$

Among them, *Denoising Score Matching (DSM)* (Vincent, 2011) shows that minimizing this divergence on noisy data comes back to training a denoising model with the L^2 loss. Thus, an image denoiser can be used to approximate the score $\nabla \log p_X(x)$. This result will be explained in more details in Section 2.4.1. In this manuscript, we will explore the power of this *denoising prior* for regularizing image inverse problems. A review of existing algorithms using this prior is given in Section 2.4.

Instead of the energy, *score-based models* directly parameterize the score with a network $s_\theta(x)$, trained to approximate $\nabla \log p_X(x)$. An energy-based model $E^\theta(x)$ can be turned to a score model via $s_\theta(x) = -\nabla E^\theta(x)$. *Diffusion models* Ho et al. (2020); Song et al. (2020a) are score-based models with well-chosen parameterization. They became recently extremely efficient and popular as a powerful generative model. Score-based models have close connections with the RED algorithms that will be further analyzed in this manuscript. This connection will be analyzed with further details in the perspectives section, Chapter 7.

2.1.2 A variety of data-fidelity terms

In this section, we review the most common data-fidelity terms f appearing in the literature. They usually write as a distance, in the general sense, between the degraded observation y and the current estimate x:

$$f(x) = d(Ax, y). \tag{2.16}$$

Under a general variational formulation (2.2), one can freely choose the data-fidelity term, depending on the type of signal considered or on the regularity we wish to have on f. The most standard data-fidelity term is the L^2 distance

$$f(x) = \frac{1}{2}\left|\left|Ax - y\right|\right|^2 \tag{2.17}$$

which is infinitely differentiable, convex and verifies $\nabla^2 f = A^T A$. The smoothness and strong convexity of f then depends on the eigenvalues of the operator $A^T A$. For most of the linear inverse problems (*e.g.* deblurring, super-resolution, in painting ...) $A^T A$ has bounded eigenvalues, and it is common practice to normalize f with the spectral norm of $A^T A$ so that f has Lipschitz constant 1. Except for particular applications such as denoising, $A^T A$ is generally singular and f may not be strongly convex. In applications such as Positron Emission Tomography (PET) or astronomical CCD cameras (Bertero et al., 2009), where images are obtained by counting particles (photons or electrons) and have positive values, it is also frequent to choose for d the Kullback-Liebler divergence $f(x) = \mathrm{KL}(y, Ax)$ or $f(x) = \mathrm{KL}(Ax, y)$ which is convex but has non-Lipschitz gradient at 0.

On the other hand, under the Bayesian formulation (2.5), the choice of the data-fidelity is not free but inherently linked to the noise distribution p_N chosen to model the degradation via

$$f(x) = -\log p_{Y|X}(y|x) = -\log p_N(y|Ax). \tag{2.18}$$

Assuming Gaussian noise, we have $p_{Y|X}(y|x) \propto \exp\left(-\frac{||Ax-y||^2}{2\sigma^2}\right)$ and thus we retrieve the L^2 data-fidelity term. Similarly, assuming Laplace noise $p_{Y|X}(y|x) \propto \prod_{i=1}^m \exp\left(-\gamma|(Ax)_i - y_i|\right)$, we get a nonconvex L^1 data-fidelity term. Finally, assuming Poisson noise, $p_{Y|X}(y|x) \propto \prod_{i=1}^m (Ax)_i^{y_i} \exp(-(Ax)_i)$, the log-likelihood writes

$$-\log p_{Y|X}(y|x) = \sum_{i=1}^m (Ax)_i - y_i \log((Ax)_i) = KL(y, Ax) + \text{const} \tag{2.19}$$

and we retrieve the Kullback-Liebler data-fidelity term. Note that the Kullback-Liebler in the reversed order $KL(Ax, y)$ can also be used as a data-fidelity term in a variational formulation (Bolte et al., 2018) but it does not correspond to a known noise distribution $p_{Y|X}$. Finally, in the noiseless case $y = Ax$, the data-fidelity term is the *nonsmooth* indicator function of $A^{-1}(\{y\}) = \{x \mid Ax = y\}$: $f(x) = \imath_{A^{-1}(\{y\})}$ (which, by definition, equals 0 on $A^{-1}(\{y\})$ and $+\infty$ elsewhere). This setup is used for image inpainting in Chapter 4.

In the general case, one can associate a choice of a distance $d(x, y)$ in a general variational formulation (2.2), with the choice of a noise model in a Bayesian formulation (2.5) via

$$p_N(y|x) \propto \exp(-d(x, y)) \tag{2.20}$$

provided $\int \exp(-d(x, y))dx < \infty$. If d writes as a Bregman divergence, when well-defined, this noise model, referred to "Bregman noise model" in Chapter 6, belongs to the exponential family of distributions (Banerjee et al., 2005).

2.2 Some useful definitions and properties

Before going further, we need to introduce some basic concepts that will be useful in the rest of our analysis. In this section, we consider $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$. Following Rockafellar and Wets (2009), we take the convention to always define f on $\mathbb{R}^n$ by allowing the value $\pm\infty$. We are interested in the minimization of such a function f, which will therefore be assumed lower-bounded, and thus allowing only the value $+\infty$. Everything said about minimization can however be translated to maximization, with $-\infty$ taking the part of $+\infty$. Following this path, we denote the domain of f by

$$\text{dom}\, f = \{x \in \mathbb{R}^n, f(x) \neq +\infty\}. \tag{2.21}$$

We first recall that f is *proper* if $\text{dom}(f) \neq \emptyset$ and *lower semicontinuous (lsc)* if $\forall x \in \mathbb{R}^n$ $\liminf_{y \to x} f(y) \geqslant f(x)$. Moreover, it is *coercive* if $\lim_{||x|| \to +\infty} f(x) = +\infty$. We also recall the definition of convexity.

Definition 1 (Convexity, strong and weak convexity)**.**

- *f is convex if $\forall (x, y) \in \text{dom}\, f \times \text{dom}\, f$ and $\forall \lambda \in (0, 1)$,*

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y). \tag{2.22}$$

  If the inequality is strict, then f is strictly convex.

- *For $\alpha \in \mathbb{R}$, f is α-convex if $x \to f(x) - \frac{\alpha}{2}||x||^2$ is convex. In particular, if $\alpha > 0$, f is called α-strongly convex and if $\alpha < 0$, for $M = -\alpha$, f is called M-weakly convex.*

2.2.1 Subdifferential

As we will deal with both convex and nonconvex functions, we need to give some details on different notions of the subdifferential. We first define the standard version of the subdifferential and then introduce more specific versions for nonconvex functions.

Definition 2 (Subdifferential (Bauschke and Combettes, 2011a))**.** *Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ proper. The* subdifferential *of f is the point-to-set operator ∂f defined by*

$$\forall x \in \mathbb{R}^n, \partial f(x) = \{\omega \in \mathbb{R}^n, \forall y \in \mathbb{R}^n, f(y) - f(x) - \langle \omega, y - x \rangle \geq 0\}. \tag{2.23}$$

Let $x \in \mathbb{R}^n$, f is said *subdifferentiable* at x if $\partial f(x) \neq 0$. The elements of $\partial f(x)$ are the *subgradients* of f at x. $\partial f(x)$ is a closed and convex set (Bauschke and Combettes, 2011a, Proposition 16.4). The domain of the subdifferential stands for

$$\mathrm{dom}(\partial f) = \{x \in \mathbb{R}^n, \partial f(x) \neq \emptyset\}, \tag{2.24}$$

and verifies $\mathrm{dom}(\partial f) \subset \mathrm{dom}(f)$. If f convex is differentiable on its domain, $\mathrm{dom}(\partial f) = int\,\mathrm{dom}(f)$ and for $x \in int\,\mathrm{dom}(f)$, $\partial f(x) = \{\nabla f(x)\}$. An important property of the subdifferential is that it characterizes global minimizers of a proper function:

Theorem 1 (Fermat's rule (Bauschke and Combettes, 2011a, Theorem 16.3))**.** *Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ proper. Then*

$$\mathrm{Argmin}\, f = \mathrm{zeros}(\partial f) = \{x \in \mathbb{R}^n, 0 \in \partial f(x)\}. \tag{2.25}$$

Subdifferential of a nonconvex function

When f is nonconvex, the previous notion of subdifferential may not be informative enough. Indeed, for minimizing a nonconvex function, there is usually no hope to target a global minimum. Instead, one can hope to reach a point that is critical/stationary for some generalized notion of subdifferential. There exists a variety of generalized notions of subdifferentials for nonconvex functions (*e.g.* Clarke subdifferential, Fréchet subdifferential, limiting subdifferential). Depending on the properties of the function and the type of iterative scheme we are using, the appropriate choice of subdifferential may vary.

To study the convergence of proximal iterative schemes for minimizing a nonconvex function, Attouch et al. (2013) use as notion of subdifferential the *limiting subdifferential*, also called *general subgradient* in Rockafellar and Wets (2009):

$$\partial^{lim} f(x) = \left\{\omega \in \mathbb{R}^n, \exists x_k \to x, f(x_k) \to f(x), \omega_k \to \omega, \omega_k \in \hat{\partial} f(x_k)\right\} \tag{2.26}$$

with $\hat{\partial} f$ the *Fréchet subdifferential* of f (also called *regular subgradient* in Rockafellar and Wets (2009)) defined as

$$\hat{\partial} f(x) = \left\{\omega \in \mathbb{R}^n, \liminf_{y \to x} \frac{f(y) - f(x) - \langle \omega, y - x \rangle}{||x - y||} \geq 0\right\}. \tag{2.27}$$

The three introduced notions of subdifferential verify for $x \in \mathrm{dom}\, f$

$$\partial f(x) \subset \hat{\partial} f(x) \subset \partial^{lim} f(x). \tag{2.28}$$

For f *convex*, the three subdifferentials coincide (Rockafellar and Wets, 2009, Proposition 8.12). For f of class $\mathcal{C}^1$ but not necessarily convex, the Fréchet and limiting subdifferential coincide with the usual concept of gradient $\hat{\partial} f(x) = \partial^{lim} f(x) = \{\nabla f(x)\}$. This generalized notion of subdifferential gives birth to generalized notions of *critical point* or *stationary point i.e.* $x \in \mathbb{R}^n$ such that

$$0 \in \partial^{lim} f(x). \tag{2.29}$$

A necessary (but not sufficient) condition for $x \in \mathbb{R}^n$ to be a local minimizer of a nonconvex function f is $0 \in \hat{\partial} f(x)$ and thus $0 \in \partial^{lim} f(x)$. Last but not least, we give two important properties of the limiting subdifferential.

Proposition 1 (Sum rule (Rockafellar and Wets, 2009, 8.8(c)))**.** *If $F = f + g$ with f of class $\mathcal{C}^1$ and g proper. Then for $x \in \mathrm{dom}(g)$,*

$$\partial^{lim} F(x) = \nabla F(x) + \partial^{lim} g(x) \tag{2.30}$$

Proposition 2 (Closeness of the limiting subdifferential)**.** *For a sequence $(x_k, \omega_k) \in \mathrm{Graph}(\partial^{lim} f)$, if $(x_k, \omega_k) \to (x, \omega)$ and $f(x_k) \to f(x)$, then $(x, \omega) \in \mathrm{Graph}(\partial^{lim} f)$.*

where $\mathrm{Graph}(F)$ stands for the graph of a point-to-set mapping

$$\mathrm{Graph}\, F = \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^m, y \in F(x)\}. \tag{2.31}$$

For additional details on the notion of limiting subdifferential, we refer to (Rockafellar and Wets, 2009, Chapter 8). In the rest of the manuscript, except if explicitly mentioned, the subdifferential ∂f of a proper function f corresponds to the standard subdifferential from Definition 2. In different parts of the manuscript, we will denote for simplicity ∂f as the limiting subdifferential $\partial^{lim} f$. But in that case, it will be explicitly mentioned.

2.2.2 Proximity operator

An important operator that will be used across the manuscript is the proximity operator.

Definition 3 (Proximity operator)**.** *For $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$, the proximity operator of f, denoted Prox_f is the point-to-set mapping Prox_f defined by*

$$\forall x \in \mathbb{R}^n, \mathrm{Prox}_f(x) = \arg\min_{z \in \mathbb{R}^n} f(z) + \frac{1}{2} ||x - z||^2. \tag{2.32}$$

For f proper, lsc, bounded from below, and $x \in \mathrm{dom}\, f$, $\mathrm{Prox}_f(x)$ is nonempty. If f is M-weakly convex with $M < 1$, it is single-valued. We say that f is *proximable* if its proximity operator can be computed in closed-form. By optimality of the proximity operator, we get directly the following characterization,

Proposition 3. *Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ proper lsc. For $x \in \mathrm{dom}\, f$,*

$$z \in \mathrm{Prox}_f(x) \Leftrightarrow x - z \in \partial f(z). \tag{2.33}$$

This property is the origin of the notation

$$\mathrm{Prox}_{\tau f}(x) = (\mathrm{Id} + \tau \partial f)^{-1}(x). \tag{2.34}$$

Note that if f is **nonconvex**, using a specific notion of subdifferential such as the limiting subdifferential (defined in Section 2.2.1), we only have the implication $\Longrightarrow$ in (2.33).

Interpretations of the proximity operator:

(i) First, the proximity operator is a generalization of the **projection** on a set. Indeed, if f is the indicator of C a non-empty closed set, then $\text{Prox}_f(x)$ corresponds to the projection of x on C.

(ii) Second, from Proposition 3, we have that for f convex and for $\tau > 0$,

$$\text{Fix}(\text{Prox}_{\tau f}) = \text{zeros}(\partial f). \tag{2.35}$$

Therefore, minimizing f can be done by finding fixed points of the proximity operator, for instance, via the fixed-point iterations, called *proximal point algorithm*

$$x_{k+1} = \text{Prox}_{\tau f}(x_k). \tag{2.36}$$

For differentiable f, using Proposition 3, these iterations can be rewritten as

$$x_{k+1} = x_k - \tau \nabla f(x_{k+1}). \tag{2.37}$$

A proximal step can then be seen as a step of **implicit gradient descent** for minimizing f. We will study in Section 4.2.2 the convergence of this algorithm for convex and nonconvex f.

(iii) Another useful interpretation of the proximity operator is via the **Moreau envelope** or Moreau-Yosida regularization $e_\lambda f$ of a proper lsc function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$, with parameter $\lambda > 0$, defined as

$$e_\lambda f(x) = \inf_{z \in \mathbb{R}^n} f(z) + \frac{1}{2\lambda} ||x - z||^2 \tag{2.38}$$

It is the inf-convolution between f and $x \to \frac{1}{2\lambda} ||x||^2$. It is everywhere a lower bound of f and for convex f, the stationary points of f and $e_\lambda f$ coincide. Moreover, for convex f, $e_\lambda f$ is convex and continuously differentiable with gradient (Rockafellar and Wets, 2009, Theorem 2.26)

$$\nabla e_\lambda f(x) = \frac{1}{\lambda} \left(x - \text{Prox}_{\lambda f}(x)\right). \tag{2.39}$$

By firm nonexpansivity of $\text{Prox}_{\lambda f}$ (see Section 3.1.3), $x - \text{Prox}_{\lambda f}(x)$ is then 1-Lipschitz and $\nabla e_\lambda f$ is $\frac{1}{\lambda}$-Lipschitz. With this result, for convex f, $\text{Prox}_{\lambda f} = \text{Id} - \lambda \nabla e_\lambda f$ can then be seen as a step of explicit gradient descent on the Moreau envelope of f.

(iv) Eventually, with relation (2.34), the proximity operator writes as the **resolvent** of the point-to-set mapping ∂f (given an operator $T : \mathbb{R}^n \to \mathbb{R}^n$ and $\lambda < 0$, the resolvent is defined has $(\text{Id} + \lambda T)^{-1}$). For f proper, ∂f is a monotone operator (Combettes, 2018). We recall that a multivalued operator A defined on $\mathbb{R}^n$ is monotone if and only if

$$\forall x, y \in \mathbb{R}^n, \hat{x} \in A(x), \hat{y} \in A(y), \langle x - y, \hat{x} - \hat{y} \rangle \geq 0 \tag{2.40}$$

Combettes (2018) proposes to generalize the theory for convex optimization problems to monotone inclusion problems. Indeed, monotone operator theory is used in many areas above optimization such as partial differential equations, evolution equations and inclusions, and nonlinear equations. More details on this generalization are given in the perspectives Chapter 7.

Characterizations of the proximity operator

We also report important results characterizing proximity operators as gradients of convex potentials. Moreau Moreau (1965) first showed that the proximity operator of a **convex** function corresponds to a nonexpansive (1-Lipschitz) gradient of convex function.

Theorem 2 ((Moreau, 1965, Corollary 10.c)). *An operator $T : \mathbb{R}^n \to \mathbb{R}^n$ is the proximity operator of a proper lsc convex function $\phi : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$* i.e. *$f = \text{Prox}_\phi$ if and only if the following conditions hold jointly:*

(i) T is nonexpansive i.e. *1-Lipschitz.*

(ii) there exists a proper lsc convex function $\psi : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ such that for each $x \in \mathbb{R}^n$, $T(x) \in \partial\psi(x)$.

Gribonval and Nikolova (2020) then extended this result to the characterization of the proximity operator of **nonconvex** functions by removing the nonexpansivity condition.

Theorem 3 (Gribonval and Nikolova (2020)). *An operator $T : \mathbb{R}^n \to \mathbb{R}^n$ is the proximity operator of a function $\phi : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$* i.e. *$T(x) \in \text{Prox}_\phi(x)$ for each $x \in \mathbb{R}^n$, if and only if there exists a proper lsc convex function $\psi : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ such that for each $x \in \mathbb{R}^n$, $T(x) \in \partial\psi(x)$. Moreover, T of class $\mathcal{C}^k$ is equivalent to ψ of class $\mathcal{C}^{k+1}$ and T L-Lipschitz is equivalent to ψ $\left(1 - \frac{1}{L}\right)$-weakly convex.*

Therefore, one can fully characterize continuous proximity operators as gradients of convex functions. These results will be essential for showing the convergence of plug-and-play algorithms with denoisers that write as gradient of convex functions (Chapter 5).

2.2.3 Convex conjugate

In this section, we define the notion of *convex conjugate* and state some basic properties that will be helpful in this manuscript, in particular for the analysis of the primal-dual algorithm in Section 2.3.4. We refer to (Bauschke and Combettes, 2011a, Chapter 13) for details and proofs.

Definition 4 (Convex conjugate). *Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$. The convex (or Fenchel or Legendre-Fenchel) conjugate of f is $f^* : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ defined by, for $u \in \mathbb{R}^n$*

$$f^*(u) = \sup_{x \in \mathbb{R}^n} \langle x, u \rangle - f(x). \tag{2.41}$$

It is a proper convex function. Moreover, for proper convex lsc f, the biconjugate of f is itself:

Proposition 4 (Fenchel-Moreau (Bauschke and Combettes, 2011a, Theorem 13.32)). *Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ proper. Then f is lsc and convex if and only if $f^{**} = f$.*

We deduce from this result that any proper lsc convex function f satisfies $\forall x \in \mathbb{R}^n$

$$f(x) = \sup_{u \in \mathbb{R}^n} \langle x, u \rangle - f^*(u). \tag{2.42}$$

This relation is particularly useful to decouple f from a composition with some operator $K : \mathbb{R}^n \to \mathbb{R}^m$

$$f \circ K(x) = \sup_{u \in \mathbb{R}^n} \langle K(x), u \rangle - f^*(u). \tag{2.43}$$

As an immediate consequence of the definition of the conjugate, we have the following Fenchel-Young inequality and characterization of the subdifferential

Proposition 5 (Fenchel-Young)**.** *Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ proper. Then $\forall x, y \in \mathbb{R}^n$*

$$f(x) + f^*(y) \geq \langle x, y \rangle \tag{2.44}$$

with equality if and only if $y \in \partial f(x)$.

It will be also convenient to characterize the subdifferential of the convex conjugate with the following property.

Proposition 6. *Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ proper lsc convex. Then,*

$$u \in \partial f(x) \Leftrightarrow x \in \partial f^*(u). \tag{2.45}$$

Finally, an important result on the proximity operator of the convex conjugate is given by Moreau's identity.

Proposition 7 (Moreau's identity (Bauschke and Combettes, 2011a, Theorem 14.3))**.** *Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ proper lsc convex. Then for $\lambda > 0$, and $x \in \mathbb{R}^n$,*

$$\operatorname{Prox}_{\lambda f^*}(x) = x - \lambda \operatorname{Prox}_{\frac{1}{\lambda} f}(\frac{x}{\lambda}). \tag{2.46}$$

2.3 First-order optimization algorithms

In this section, we review the most standard first-order algorithms for minimizing the sum of two functions

$$\min_{x \in \mathbb{R}^n} F(x) = f(x) + g(x) \tag{2.47}$$

with $f, g : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ proper, lower semi-continuous and lower-bounded. In the general case, and except if explicitly mentioned, f and g are *nonconvex* and *nonsmooth*. For nonconvex F, instead of looking for a global or local minimizer, we will rather look for a *critical point i.e.* a point x such that $0 \in \partial F(x)$ for some predefined notion of subdifferential. The convergence of each algorithm will be analyzed in detail in Chapter 3, Section 3.2.

2.3.1 Proximal Gradient Descent

The Proximal Gradient Descent (PGD) (or Forward-Backward) algorithm consists in alternating an explicit gradient step on a smooth f and implicit gradient descent step on a potentially nonsmooth g. For an initial point $x_0 \in \mathbb{R}^n$ and a fixed stepsize $\tau > 0$, it corresponds to the following fixed-point iterations

$$\textbf{(PGD)} \quad x_{k+1} \in \operatorname{Prox}_{\tau g} \circ (x_k - \tau \nabla f(x_k)) \,. \tag{2.48}$$

In the general case, we do not assume g (and f) convex. Therefore, $\operatorname{Prox}_{\tau g}$ is potentially multivalued. Note that the following Proximal Point (PP) and Gradient Descent (GD) algorithms are particular cases of the PGD algorithm where one of the two functions is zero:

$$\textbf{(PP)} \quad x_{k+1} \in \operatorname{Prox}_{\tau g}(x_k) \tag{2.49}$$

$$\textbf{(GD)} \quad x_{k+1} = x_k - \tau \nabla f(x_k) \tag{2.50}$$

We verify that a fixed-point of the PGD operator corresponds to a critical point of the objective function F.

Proposition 8. *Assume f L_f-smooth. Then, for $\tau > 0$,*

$$\text{zeros}(\partial F) = \text{Fix}(\text{Prox}_{\gamma g}(\text{Id} - \gamma \nabla f)). \tag{2.51}$$

where $\text{Fix}(T) = \{x \in \mathbb{R}^n, x \in T(x)\}$ denotes the fixed points of the set-valued mapping T.

Proof.

$$\begin{aligned} 0 \in \partial F(x) &\Leftrightarrow 0 \in \nabla f(x) + \partial g(x) && (2.52)\\ &\Leftrightarrow -\gamma \nabla f(x) \in \partial \tau g(x) && (2.53)\\ &\Leftrightarrow x - \gamma \nabla f(x) \in (\text{Id} + \partial \tau g)(x) && (2.54)\\ &\Leftrightarrow x \in \text{Prox}_{\tau g}(x - \tau \nabla f(x)). && (2.55) \end{aligned}$$

□

Remark 1. *For nonconvex g, when using as subdifferential a different notion, such as the limiting subdifferential, the last equivalence is not true and we rather have*

$$\text{Fix}(\text{Prox}_{\gamma g}(\text{Id} - \gamma \nabla f)) \subset \text{zeros}(\partial F) \tag{2.56}$$

2.3.2 Half Quadratic Splitting

The *Half Quadratic Splitting* (HQS) algorithm consists in alternating proximal steps on f and g.

$$\textbf{(HQS)} \quad x_{k+1} \in \text{Prox}_{\tau g} \circ \text{Prox}_{\tau f}(x_k). \tag{2.57}$$

HQS does not target a minimizer of $F = f + g$ but of $e_\tau f + g$ where $e_\tau f$ stands for the Moreau envelope of f with parameter τ (defined in Section 2.2.2). Indeed, assuming f convex, using (2.39), HQS writes as PGD iterations

$$x_{k+1} = \text{Prox}_{\tau g} \circ (\text{Id} - \tau \nabla e_\tau f)(x_k), \tag{2.58}$$

with $\nabla e_\tau f$ $\frac{1}{\tau}$-Lipschitz continuous. It thus takes the form of a PGD algorithm for minimizing

$$\min_{x \in \mathbb{R}^n} e_\tau f(x) + g(x) = \min_{x,z \in \mathbb{R}^n} f(z) + g(x) + \frac{1}{2\tau} ||x - z||^2 \tag{2.59}$$

2.3.3 Douglas-Rachford Splitting / ADMM

The *Douglas-Rachford Splitting (DRS)* algorithm (Douglas and Rachford, 1956) minimizes $f + g$ by alternating proximal steps on f and g via the *reflected proximity* operator

$$\text{Rprox}_f = 2\,\text{Prox}_f - \text{Id}\,. \tag{2.60}$$

In its most general form, the algorithm writes, for $\beta \in (0, 1]$,

$$\textbf{(DRS)} \quad x_{k+1} \in \left(\beta\, \text{Rprox}_{\tau g} \circ \text{Rprox}_{\tau f} + (1 - \beta)\, \text{Id}\right)(x_k) \tag{2.61}$$

where, in the nonconvex case, $\text{Rprox}_g \circ \text{Rprox}_f$ is a composition of point-to-set mappings. $\beta = \frac{1}{2}$ corresponds to the standard version of Douglas-Rachford Splitting, while $\beta = 1$ is

generally referred to *Peaceman-Rachford splitting* (PRS). This algorithm can be rewritten as

$$\textbf{(DRS)} \quad \begin{cases} y_{k+1} & \in \mathrm{Prox}_{\tau f}(x_k) \\ z_{k+1} & \in \mathrm{Prox}_{\tau g}(2y_{k+1} - x_k) \\ x_{k+1} & = x_k + 2\beta(z_{k+1} - y_{k+1}). \end{cases} \tag{2.62}$$

In the convex setting, it is well known that DRS is equivalent to the *Alternating Direction Method of Multipliers* (ADMM) applied to the dual problem of (2.47) (Gabay, 1983). We now show, in the general nonconvex case, the primal equivalence between DRS with $\beta = \frac{1}{2}$ and ADMM algorithm, after swapping the order between f and g steps

$$\textbf{(ADMM)} \quad \begin{cases} z_{k+1} & \in \mathrm{Prox}_{\tau g}(y_k - u_k) \\ y_{k+1} & \in \mathrm{Prox}_{\tau f}(u_k + z_{k+1}) \\ u_{k+1} & = u_k + z_{k+1} - y_{k+1}. \end{cases} \tag{2.63}$$

Lemma 1. *DRS* (2.62) *with* $\beta = \frac{1}{2}$ *and ADMM* (2.63) y_k *and* z_k *iterations are equivalent.*

Proof. Starting from the DRS iterations (2.62), using $x_k = u_k + z_k$,

$$\begin{cases} y_{k+1} & \in \mathrm{Prox}_{\tau f}(u_k + z_k) \\ z_{k+1} & \in \mathrm{Prox}_{\tau g}(2y_{k+1} - (u_k + z_k)) = \mathrm{Prox}_{\tau g}(y_{k+1} - (u_k + z_k - y_{k+1})) \\ u_{k+1} & = u_k + z_k - y_{k+1} \end{cases} \tag{2.64}$$

which also writes, by inverting the last two steps,

$$\begin{cases} y_{k+1} & \in \mathrm{Prox}_{\tau f}(u_k + z_k) \\ u_{k+1} & = u_k + z_k - y_{k+1} \\ z_{k+1} & \in \mathrm{Prox}_{\tau g}(y_{k+1} - u_{k+1}) \end{cases} \tag{2.65}$$

or again

$$\begin{cases} z_{k+1} & \in \mathrm{Prox}_{\tau g}(y_k - u_k) \\ y_{k+1} & \in \mathrm{Prox}_{\tau f}(u_k + z_{k+1}) \\ u_{k+1} & = u_k + z_{k+1} - y_{k+1}. \end{cases} \tag{2.66}$$

□

We verify in the convex case that a fixed-point of the DRS operator corresponds to a critical point of the objective function F.

Proposition 9. *Let* f *and* g *convex. For* $\tau > 0$,

$$\mathrm{zeros}(\partial F) = \mathrm{Prox}_{\tau f}\left(\mathrm{Fix}\left(\mathrm{Rprox}_{\tau g} \circ \mathrm{Rprox}_{\tau f}\right)\right). \tag{2.67}$$

Proof.

$0 \in \partial F(x)$
$\Leftrightarrow 0 \in \partial\tau f(x) + \partial\tau g(x)$
$\Leftrightarrow \exists z \in \mathbb{R}^n$ such that $-z \in \partial\tau g(x)$ and $z \in \partial\tau f(x)$
$\Leftrightarrow \exists y \in \mathbb{R}^n$ such that $x - y \in \partial\tau g(x)$ and $y - x \in \partial\tau f(x)$
$\Leftrightarrow \exists y \in \mathbb{R}^n$ such that $2x - y \in (\mathrm{Id} + \partial\tau g)(x)$ and $y \in (\mathrm{Id} + \partial\tau f)(x)$
$\Leftrightarrow \exists y \in \mathbb{R}^n$ such that $x = \mathrm{Prox}_{\tau g}(2x - y)$ and $x = \mathrm{Prox}_{\tau f}(y)$
$\Leftrightarrow \exists y \in \mathbb{R}^n$ such that $x = \mathrm{Prox}_{\tau g} \circ \mathrm{Rprox}_{\tau f}(y)$ and $x = \mathrm{Prox}_{\tau f}(y)$
$\Leftrightarrow \exists y \in \mathbb{R}^n$ and $z = \mathrm{Rprox}_{\tau f}(y)$ such that $2x - z = \mathrm{Rprox}_{\tau g} \circ \mathrm{Rprox}_{\tau f}(y)$ and $x = \mathrm{Prox}_{\tau f}(y)$

For $x = \text{Prox}_{\tau f}(y)$ and $z = \text{Rprox}_{\tau f}(y)$, we have $2x - z = y$. Thus we get,

$$0 \in \partial F(x) \Leftrightarrow \exists y \in \mathbb{R}^n \text{ such that } y \in \text{Fix}\left(\text{Rprox}_{\tau g} \circ \text{Rprox}_{\tau f}\right) \text{ and } x = \text{Prox}_{\tau f}(y).$$

□

We thus get a critical point of F by estimating a fixed point of the DRS operator $T_\tau = \text{Rprox}_{\tau g} \circ \text{Rprox}_{\tau f}$ via the fixed-point iterations (2.61) **and then** by applying $\text{Prox}_{\tau f}$ to this fixed point. This is the reason why it is useful to write the DRS iterations as (2.62). With the notations of (2.62), if we show that the sequence (x_k) converges, then $y_k = \text{Prox}_{\tau f}(x_k)$ converges towards a critical point of F.

2.3.4 Primal-dual

Primal-Dual (PD) algorithms target solutions of the more general problem:

$$\min_{x \in \mathbb{R}^n} f(Kx) + g(x), \tag{2.68}$$

where $K \in \mathbb{R}^{n \times m}$ is a linear operator and f, g are still proper, lower semi-continuous and lower-bounded. Primal-dual is particularly useful when the proximal operators of both f and g are simple to compute, but the proximal operator of $f \circ K$ is not easily accessible.

Assume f and g **convex**, using the convex conjugate of $f(K\cdot)$, problem (2.68) has the equivalent *primal-dual* formulation

$$\min_{x \in \mathbb{R}^n} \max_{y \in \mathbb{R}^m} \langle K^* y, x \rangle - f^*(y) + g(x). \tag{2.69}$$

Chambolle and Pock (2011) proposed the following algorithm to solve (2.69), which is called *Chambolle-Pock* (CP) or *Primal-Dual Hybrid Gradient* (PDHG). For stepsizes $\tau > 0$, $\sigma > 0$, and relaxation parameter $\beta \in [0, 1]$, it writes

$$\left\{ \begin{array}{lcl} y_{k+1} & = & \text{Prox}_{\sigma f^*}(y_k + \sigma K \bar{x}_k) \\ x_{k+1} & = & \text{Prox}_{\tau g}(x_k - \tau K^* y_{k+1}) \\ \bar{x}_{k+1} & = & x_{k+1} + \beta(x_{k+1} - x_k). \end{array} \right. \tag{2.70}$$

For convex f and g, it is shown in (Chambolle and Pock, 2011, Theorem 1) that, for $\tau\sigma \, ||K^* K|| < 1$, the iterates (x_k, y_k) given by the algorithm (2.70) converge towards (x^*, y^*) solution of the primal-dual problem (2.69).

Eventually, the primal-dual algorithm (2.70) can be written in a fully primal version using Moreau's identity (Proposition 7). We then get rid of the convex conjugate and the algorithm is well-defined for **nonconvex** f and g functions, and writes

$$\left\{ \begin{array}{ll} z_{k+1} & \in \text{Prox}_{\frac{1}{\sigma} f}(\frac{1}{\sigma} y_k + K \bar{x}_k) \\ y_{k+1} & = y_k + \sigma(K \bar{x}_k - z_{k+1}) \\ x_{k+1} & \in \text{Prox}_{\tau g}(x_k - \tau K^* y_{k+1}) \\ \bar{x}_{k+1} & = x_{k+1} + \beta(x_{k+1} - x_k). \end{array} \right. \tag{2.71}$$

2.4 Image restoration with Gaussian denoising priors

In this section, we come back to the image denoising task. Denoising Gaussian noise is the simplest and the most studied inverse problem in imaging. There exists a wide variety of efficient generic Gaussian denoisers *i.e.* explicit operators $D_\sigma : \mathbb{R}^n \to \mathbb{R}^n$ built to approximate x given $y = x + \xi$ with ξ white Gaussian noise with standard deviation σ. In this section, we will see that such a generic denoiser can be used as a prior for regularizing more general inverse problems. After presenting the MMSE and MAP theoretical denoising estimators in Section 2.4.1, we will introduce in Section 2.4.2 the RED and PnP algorithms, two family of iterative schemes for solving image inverse problems with Gaussian denoising priors.

2.4.1 MMSE and MAP Gaussian denoisers

Consider a clean image $x \in \mathcal{X} = \mathbb{R}^n$, realization of the random variable X with prior distribution p_X. Assume the observation $y \in \mathbb{R}^n$ is obtained from x by adding random Gaussian noise ξ of standard deviation σ *i.e.* $y = x + \xi$ with ξ realization of $\mathcal{N}(0, \sigma^2 \mathrm{Id})$. We denote by G_σ the Gaussian probability density of $\mathcal{N}(0, \sigma^2 \mathrm{Id})$, and $p_\sigma = p_Y = p_X * G_\sigma$ the marginal distribution of Y. We now describe the MAP and MMSE denoisers which are two optimal estimators $y \to \hat{x}(y)$, intractable in the general case.

The Minimum Mean Square Error (MMSE) (or posterior mean) estimator. It selects the mean of the posterior.

$$D_\sigma^{MMSE}(y) = \mathbb{E}[X|Y = y] = \int_{x \in \mathbb{R}^n} x p_{X|Y}(x|y) dx. \tag{2.72}$$

It is the optimal Bayes estimator for the L^2 loss *i.e.*

$$D_\sigma^{MMSE} = \underset{D_\sigma \text{ measurable}}{\arg\min} \mathbb{E}_{(X,Y)} \left[||D_\sigma(Y) - X||^2 \right]. \tag{2.73}$$

An important well-known property of the MMSE Gaussian denoiser is that it is related to the *score* $\nabla_y \log p_\sigma$ via Tweedie's identity.

Proposition 10 (Tweedie's identity (Efron, 2011))**.** *With the previous notation, the MMSE estimator for the Gaussian denoising problem verifies*

$$D_\sigma^{MMSE} = \mathrm{Id} + \sigma^2 \nabla_y \log p_\sigma. \tag{2.74}$$

Proof. Using that $\int_{x \in \mathbb{R}^n} p_{X|Y}(x|y) dx = 1$ together with Bayes formula, we can derive

$$\forall y \in \mathbb{R}^n, \quad D_\sigma^{MMSE}(y) = y + \int_{x \in \mathbb{R}^n} (x - y) p_{X|Y}(x|y) dx \tag{2.75}$$

$$= y + \int_{x \in \mathbb{R}^n} (x - y) \frac{p_{Y|X}(y|x) p_X(x)}{p_Y(y)} dx. \tag{2.76}$$

Meanwhile, using that $p_\sigma(y) = p_Y(y) = \int_x p_{Y|X}(y|x) p_X(x) dx = \int_x G_\sigma(y - x) p_X(x) dx$, and that $\nabla_y G_\sigma(y - x) = \frac{1}{\sigma^2}(x - y) G_\sigma(y - x)$, we get

$$\forall y \in \mathbb{R}^n, \quad \nabla_y \log p_\sigma(y) = \frac{1}{\sigma^2} \int_x \frac{(x - y) G_\sigma(x - y) p_X(x) dx}{\int_{\hat{x}} G_\sigma(y - \hat{x}) p_X(\hat{x}) d\hat{x}} \tag{2.77}$$

$$= \frac{1}{\sigma^2} \int_{x \in \mathbb{R}^n} (x - y) \frac{p_{Y|X}(y|x) p_X(x)}{p_Y(y)} dx. \tag{2.78}$$

Combining (2.76) and (2.78), we get the desired identity. □

Denoising Score Matching (DSM) provides a related result. It states that training a parametric denoiser by minimization of the L^2 denoising loss comes back to the approximation of the right term of (2.74) with this denoiser.

Proposition 11 (Denoising Score Matching (Vincent, 2011))**.** *Given a parametric estimator $D_\sigma^\theta : \mathbb{R}^n \to \mathbb{R}^n$ with parameters θ,*

$$\mathbb{E}_{Y\sim p_\sigma}\left[\left|\left|D_\sigma^\theta(Y) - (Y + \sigma^2 \nabla \log p_\sigma(Y))\right|\right|^2\right] = \mathbb{E}_{X\sim p_X, Y\sim p_\sigma}\left[\left|\left|D_\sigma^\theta(Y) - X\right|\right|^2\right] + \text{ const.} \tag{2.79}$$

where the constant is w.r.t θ.

Using Tweedie's identity, the DSM result rewrites as

$$\mathbb{E}_{Y\sim p_\sigma}\left[\left|\left|D_\sigma^\theta(Y) - D_\sigma^{MMSE}(Y)\right|\right|^2\right] = \mathbb{E}_{X\sim p_X, Y\sim p_\sigma}\left[\left|\left|D_\sigma^\theta(Y) - X\right|\right|^2\right] + \text{ const.} \tag{2.80}$$

This indicates that *training a neural network D_σ^θ to denoise Gaussian noise by minimizing the L^2 denoising loss (second term in (2.79)) comes back to training this network to approximate the MMSE denoiser.*

Using Tweedie's formula, we can also interpret the MMSE Gaussian denoiser as a step of explicit gradient descent, with stepsize σ^2, on $-\log p_\sigma$, where $p_\sigma = p_X * G_\sigma$ is a Gaussian-smoothed version of p_X.

Lemma 2 (Gribonval (2011, Lemma A.1))**.** *$p_\sigma = p_X * G_\sigma$ verifies $\frac{1}{2\sigma^2}\left|\left|x\right|\right|^2 + \log p_\sigma$ strictly convex.*

With this property, from the convergence analysis of the Proximal Gradient Descent algorithm in the nonconvex setting realized in Section 3.2 (refer to Theorem 7 and Remark 8), iterating the MMSE denoiser can then be shown to converge towards a stationary point of $\log p_\sigma$.

Finally, note that, from formula (2.77), considering the limit case where X is discrete over a dataset $\{x_i\}_i$ and $p_X \approx \sum_i \delta_{x_i}$ (even if strictly speaking, in that case, there is no density w.r.t. Lebesgue measure), the MMSE corresponds to a weighted average of all the data points $\{x_i\}_i$, with weights $\exp(-\frac{1}{\sigma^2}\left|\left|y - x_i\right|\right|^2)$.

The Maximum A Posteriori (MAP) estimator selects a maximum of the posterior distribution.

$$D_\sigma^{MAP}(y) \in \arg\max_{x\in\mathcal{X}} p_{X|Y}(x|y) = \arg\min_{x\in\mathcal{X}} -\log p_{Y|X}(y|x) - \log p_X(x). \tag{2.81}$$

It is not a Bayesian estimator in the sense that it does not minimize a Bayes risk. In the Gaussian noise setting, the posterior distribution is $p_{Y|X}(y|x) = G_\sigma(y-x) \propto \exp\left(-\frac{||x-y||^2}{2\sigma^2}\right)$ and the MAP estimator (2.5) writes

$$D_\sigma^{MAP}(y) \in \arg\min_{x\in\mathbb{R}^n} \frac{1}{2}\left|\left|x - y\right|\right|^2 - \sigma^2 \log p_X(x) \tag{2.82}$$

$$\in \text{Prox}_{-\sigma^2 \log p_X}(y) \tag{2.83}$$

From the different interpretations of the proximity operator proposed in Section 2.2.2, several observations on the MAP denoiser can be done.

First, considering again the case where X is discrete, the MAP realizes the projection of the input y on the data space.

In the more realistic absolutely continuous setting, we can interpret the MAP denoiser as a step of implicit gradient descent, with stepsize σ^2, on the true negative log image prior $-\log p_X$. Using the convergence analysis of the Proximal Gradient descent algorithm in the nonconvex setting (refer to Theorem 7) with $f = 0$ and $g = -\log p_X$, iterating the MAP denoiser monotonically increases $\log p_X$ and could be shown to converge towards a stationary point of $\log p_X$.

It is also interesting to mention that (although unrealistic in the general case) assuming $-\log p_X$ convex, we have from identity (2.39) that, like the MMSE, the MAP denoiser also realizes a step of *explicit* gradient descent on a smoothed version of $-\log p_X$, which is here given by the Moreau-Yosida regularization

$$g_\sigma^{MAP}(y) = e_{\sigma^2}(-\log p_X)(y) = \inf_x g(x) + \frac{1}{2\sigma^2}\left|\left|x-y\right|\right|^2 \tag{2.84}$$

In comparison, recall that the MMSE case realizes a step of gradient descent on

$$g_\sigma^{MMSE} = -\log(p_X * G_\sigma) \tag{2.85}$$

g_σ^{MAP} verifies

$$\exp(-g_\sigma^{MAP}(y)) = \sup_x \exp(-g(x)) \exp(-\frac{1}{2}\left|\left|x-y\right|\right|^2) = \left|\left|p_X G_\sigma(.-y)\right|\right|_\infty . \tag{2.86}$$

i.e.

$$g_\sigma^{MAP}(y) = \log\left(-\left|\left|p_X G_\sigma(.-y)\right|\right|_\infty\right) \tag{2.87}$$

In comparison, g_σ^{MMSE} verifies

$$\exp(-g_\sigma^{MMSE}(y)) = \int_x p_X(x) G_\sigma(x-y)dx = \left|\left|p_X G_\sigma(.-y)\right|\right|_1 . \tag{2.88}$$

i.e.

$$g_\sigma^{MMSE}(y) = \log\left(-\left|\left|p_X G_\sigma(.-y)\right|\right|_1\right) \tag{2.89}$$

Therefore, the MAP and MMSE denoisers perform an explicit step of gradient descent on a potential given by the log of, respectively, the L^1 and L^∞ norm of $x \to p_X(x)G_\sigma(x-y)$.

Related to the distinction between MAP and MMSE, Gribonval (2011) shows that the Gaussian noise MMSE with prior p_X is a MAP, but with another prior, which is related but different from p_X:

Proposition 12. *There is $\phi_\sigma : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ such that the MMSE Gaussian denoiser* (2.72) *is the unique solution of a variational optimization problem of the form*

$$D_\sigma^{MMSE}(y) = \operatorname*{arg\,min}_{x\in\mathbb{R}^n} \frac{1}{2}\left|\left|x-y\right|\right|^2 + \phi_\sigma(x) = \mathrm{Prox}_{\phi_\sigma}(y) \tag{2.90}$$

Proof. This result can be proved using the characterization, from the same authors, of continuous proximity operators as gradients of convex functions, given in Theorem 3. Indeed, from Tweedie's identity (2.74) $D_\sigma^{MMSE} = \nabla\psi_\sigma$ with $\psi_\sigma : x \to \frac{1}{2}\left|\left|x\right|\right|^2 + \sigma^2 \log p_\sigma(x)$. Moreover, with Lemma 2, ψ_σ is convex. The fact that the MMSE is a proximity operator then follows from Theorem 3. □

In the general case, both MAP and MMSE denoisers remain theoretical estimators without tractability. However, given a generic Gaussian denoiser $D_\sigma : \mathbb{R}^n \to \mathbb{R}^n$, one can assume that D_σ approximates one of the two estimators *i.e.* given a noisy image $y \in \mathbb{R}^n$,

$$D_\sigma(y) \approx D_\sigma^{MMSE}(y) = y + \sigma^2 \nabla \log p_\sigma(y) \tag{2.91}$$

$$\text{or } D_\sigma(y) \approx D_\sigma^{MAP}(y) = \text{Prox}_{-\sigma^2 \log p_X}(y). \tag{2.92}$$

Therefore, depending on this choice, a generic denoiser provides an approximation of the true log (smoothed) prior via its gradient or its proximity operator.

2.4.2 RED and PnP algorithms for image restoration

Given the previous interpretation, we now introduce how a generic denoiser can be used to regularize image inverse problems. Recall the form of the MAP formulation of a general inverse problem (2.5)

$$\hat{x}^{MAP} = \arg\min_{x \in \mathcal{X}} -\log p_{Y|X}(y|x) - \log p_X(x) \tag{2.93}$$

$$= \arg\min_{x \in \mathcal{X}} \lambda f(x) + g(x) \tag{2.94}$$

with $f \propto -\log p_{Y|X}(y|.)$ and $g = -\log p_X$ and where we chose, for simplicity, the regularization parameter $\lambda > 0$ to be in front of f. In order to apply the first order algorithms presented in Section 2.3 to minimize the MAP objective (2.94), we need to calculate precisely the gradient or the proximity operator of $-\log p_X$. The idea of the following class of plug-and-play methods is to replace this gradient or Prox by a generic off-the-shelf denoiser. In the rest of this section, we make the distinction between gradient or proximal based plug-and-play algorithms. The former is called "Regularization by Denoising" (RED) while for the latter we keep the acronym PnP.

Regularization by Denoising (RED) algorithms

Regularization by Denoising (RED) algorithms build on top of first order algorithms for minimizing (2.94) involving the gradient of $g = -\log p_X$. Therefore, g needs to be smooth (differentiable with Lipschitz gradient). This can be guaranteed by replacing the prior p_X by its smoothed version with a Gaussian kernel $p_\sigma = p_X * G_\sigma$, where G_σ stands for the Gaussian distribution $\mathcal{N}(0, \sigma^2 \,\text{Id})$. After defining the smoothed posterior using the forward model $p(y|x)$

$$p_\sigma(x|y) = \frac{p(y|x)p_\sigma(x)}{\int_{\hat{x}} p(y|\hat{x})p_\sigma(\hat{x})d\hat{x}}, \tag{2.95}$$

the MAP formulation of the inverse problem derived in Section 2.5 now writes

$$\hat{x}_\sigma^{MAP} = \arg\min_{x \in \mathcal{X}} -\log p_{Y|X}(y|x) - \log p_\sigma(x) \tag{2.96}$$

$$= \arg\min_{x \in \mathcal{X}} \lambda f(x) - \log p_\sigma(x). \tag{2.97}$$

The functions p_σ and $-\log p_\sigma$ are $\mathcal{C}^\infty$ and we showed in Lemma 2 that $-\log p_\sigma$ is $\frac{1}{\sigma^2}$-weakly convex. Moreover, is it shown in Laumont et al. (2021) that p_σ can be made arbitrarily close to the original prior p_X as $\sigma \to 0$. Note also that this smoothing strategy enables to

work with an absolutely continuous measure p_σ even when p_X is not, for example when X is discrete and $p = \sum_i \delta_{x_i}$.

Minimization algorithms for (2.97) involving the gradient of $g_\sigma = -\log p_\sigma(x)$ encompass the full gradient descent (GD) (2.50) and the proximal gradient descent (PGD) (2.48).

$$\textbf{(PGD)} \quad x_{k+1} \in \operatorname{Prox}_{\tau\lambda f} \circ (x_k - \tau \nabla g_\sigma(x_k)) \tag{2.98}$$
$$\textbf{(GD)} \quad x_{k+1} = x_k - \tau(\nabla \lambda f(x_k) + \nabla g_\sigma(x_k)). \tag{2.99}$$

Given a real generic denoiser $D_\sigma : \mathbb{R}^n \to \mathbb{R}^n$ that approximates the MMSE denoiser introduced in Section 2.4.1, we have from Tweedie's identity (2.74),

$$\nabla g_\sigma = -\nabla \log p_\sigma = \frac{1}{\sigma^2}\left(\operatorname{Id} - D_\sigma^{MMSE}\right) \approx \frac{1}{\sigma^2}\left(\operatorname{Id} - D_\sigma\right). \tag{2.100}$$

After multiplying the problem (2.97) by σ^2 and replacing $\lambda \longleftarrow \lambda\sigma^2$, using

$$\nabla g_\sigma \longleftarrow \operatorname{Id} - D_\sigma, \tag{2.101}$$

we get the RED-PGD and RED-GD algorithms

$$\textbf{(RED-PGD)} \quad x_{k+1} \in \operatorname{Prox}_{\tau\lambda f} \circ (\tau D_\sigma(x_k) + (1-\tau)x_k) \tag{2.102}$$
$$\textbf{(RED-GD)} \quad x_{k+1} = \tau D_\sigma(x_k) + (1-\tau)x_k - \tau\lambda \nabla f(x_k). \tag{2.103}$$

Even though differently justified, RED algorithms were originally proposed in Romano et al. (2017) while their score-based interpretation was formulated in Reehorst and Schniter (2018).

PnP algorithms

PnP methods (Venkatakrishnan et al., 2013) build on proximal algorithms for minimizing (2.94) *i.e.* first order minimization algorithms (see Section 2.3) involving the proximity operator of $g = -\log p_X$, such as

$$\textbf{(PGD)} \quad x_{k+1} \in \operatorname{Prox}_{\tau g} \circ \left(x_k - \tau \nabla \lambda f(x_k)\right). \tag{2.104}$$
$$\textbf{(HQS)} \quad x_{k+1} \in \operatorname{Prox}_{\tau g} \circ \operatorname{Prox}_{\tau f}(x_k). \tag{2.105}$$
$$\textbf{(DRS)} \quad x_{k+1} \in \left(\beta \operatorname{Rprox}_{\tau g} \circ \operatorname{Rprox}_{\tau f} + (1-\beta)\operatorname{Id}\right)(x_k). \tag{2.106}$$

Remark that, compared to the PGD version (2.98) used for RED, for PnP, in (2.104), the proximal step needs to be on the prior term g and thus the gradient step on the data-fidelity term f. As explained in the presentation of the algorithms in Section 2.3, we recall that HQS does not exactly optimize (2.94) but a regularized version of this objective.

As before, we assume given a real generic operator $D_\sigma : \mathbb{R}^n \to \mathbb{R}^n$ built to remove Gaussian noise of standard deviation σ, that approximates the MAP denoiser introduced in Section 2.4.1. Then, from the MAP formulation (2.82),

$$\operatorname{Prox}_g = \operatorname{Prox}_{-\sigma^2 \log p_X} = D_\sigma^{MAP} \approx D_\sigma. \tag{2.107}$$

Once again, the σ^2 constant can be removed and accounted via the λ and τ parameters. Taking

$$\operatorname{Prox}_{\tau g} \longleftarrow D_\sigma, \tag{2.108}$$

we get the PnP algorithms

$$\textbf{(PnP-PGD)} \quad x_{k+1} \in D_\sigma \circ (x_k - \tau\lambda\nabla f(x_k)) \,. \tag{2.109}$$

$$\textbf{(PnP-HQS)} \quad x_{k+1} \in D_\sigma \circ \mathrm{Prox}_{\tau f}(x_k). \tag{2.110}$$

$$\textbf{(PnP-DRS)} \quad x_{k+1} \in \left(\beta(2D_\sigma - \mathrm{Id}) \circ \mathrm{Rprox}_{\tau f} + (1-\beta)\,\mathrm{Id}\right)(x_k). \tag{2.111}$$

For simplicity, we do not give here the PnP-ADMM and PnP-PD (Primal-dual) schemes, but they can be easily derived with the same idea.

2.4.3 Practical implementation

Choice of the algorithm Among PnP and RED schemes, the choice of the algorithm is strongly influenced by the regularity of the data-fidelity term f. In our analysis, we will consider forward degradation models with linear A. The regularity of $f(x) = -\log p(y|x)$ depends on A and on the noise distribution. In the most typical case of Gaussian inverse problems, $f(x) = ||Ax - y||^2$. When considering Poisson noise, the data-term falls back to a Kullback-Leibler divergence $f(x) = KL(y, Ax)$.

If f is non-proximable, *i.e.* its proximity operator is not available in closed-form, the use of algorithms involving $\mathrm{Prox}_{\tau f}$ should be avoided in practice. Note that for Gaussian inverse problems,

$$\mathrm{Prox}_{\tau f}(x) = \left(\mathrm{Id} + \tau A^T A\right)^{-1} (\tau A^T y + x) \tag{2.112}$$

and the calculation of the Prox amount to the calculation of the inverse of $\mathrm{Id} + \tau A^T A$. The Prox is then closed-form when A has closed-form singular value decomposition. This is of course the case for *inpainting* where A is diagonal. This is also the case for image *deblurring* when blur is performed with circular boundary conditions. However, for non-circular boundary conditions, the Prox does not have a closed-form solution. Note that when the data-fidelity takes the form $f(x) = \hat{f}(Ax)$ with $\hat{f}$ proximable but f non-proximable when $A \neq \mathrm{Id}$, it is possible to use the primal dual algorithm (or its variants) presented in Section 2.3.4 with $K = A$. In the general case of non-proximable f, a common workaround consists in approximating, at each iteration, $\mathrm{Prox}_{\tau f}$ with an internal optimization algorithm such as gradient descent.

If f is nonsmooth (*i.e.* does not have Lipschitz gradient), we cannot use algorithms requiring Lipschitz ∇f like PnP-PGD or RED-GD. This is for example the case for Poisson image denoising, for which the Kullback-Leibler data-fidelity $f(x) = KL(y, x)$ has non-Lipschitz gradient in 0.

Note also that the case of nonsmooth and non-proximable data-fidelity term arises for Poisson inverse problems with general $A \neq \mathrm{Id}$. This problem is treated in Chapter 6 by introducing a new notion of smoothness, using a Bregman divergence, which is adapted to f.

Choice of the denoiser The choice of the plugged Gaussian denoiser is of uttermost importance as it influences the restoration performance, the overall computational time and the convergence of the algorithm. Originally applied with classical denoisers such as BM3D (Venkatakrishnan et al., 2013; Chan et al., 2016; Kamilov et al., 2017) and pseudo-linear denoisers (Nair et al., 2021; Gavaskar et al., 2021), more recent approaches (Meinhardt et al., 2017; Zhang et al., 2017b; Sun et al., 2019a; Ahmad et al., 2020; Yuan et al., 2020; Sun et al., 2021; Zhang et al., 2021) rely on denoisers parameterized with deep convolutional neural networks (CNN).

The most efficient classical denoisers such as BM3D (Dabov et al., 2007) or EPLL (Zoran and Weiss, 2011) are time-consuming, and limit the acceptable number of iterations of the plug-and-play iterative scheme. Using CNN denoisers allows both state-of-the-art and time efficient algorithms. This will thus be the focus of our analysis.

The CNN most commonly encountered in PnP/RED papers is DnCNN (Zhang et al., 2017a). The advantage of DnCNN is that it provides very good denoising performance while being extremely lightweight and quick to evaluate. From a noisy input $y = x + \xi$, the main idea of DnCNN is to use residual learning to approximate the noise ξ rather than the image x directly.

More recently, state-of-the-art performance for denoising and image translation in general has been achieved with UNet (Ronneberger et al., 2015) architectures. A UNet mainly consists of two paths. An encoding path progressively downsamples an image to create meaningful feature maps at different scales, and a decoding path symmetrically upsamples the feature maps back to an image. Key components of UNets are the skip-connections between encoding and decoding blocks. They are useful to recover spatial information lost during downsampling and to stabilize training by keeping a gradient flow from the first layer to the last layer. UNet are usually fully convolutional and thus can accept input images of any size. A famous example in the PnP literature is the DRUNet (Zhang et al., 2021) architecture, represented in Figure 2.1. It consists in stacking ResNets (He et al., 2016) blocks in a UNet-shaped model. UNet denoisers have proven to be very efficient image prior in the context of image generation with score-based models (Ho et al., 2020; Song and Ermon, 2019; Song and Kingma, 2021). Similar to DnCNN, these models train very large UNets to approximate the noise or a re-scaled version of the noise.

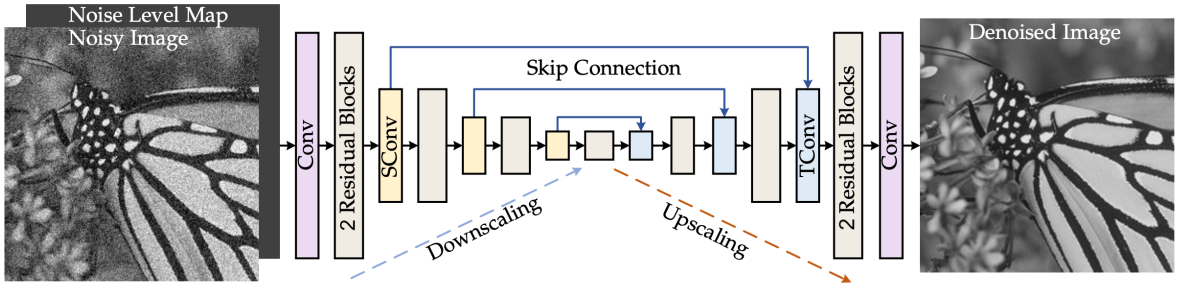


Figure 2.1: Architecture of the DRUNet denoiser (Zhang et al. (2021)).

While DnCNN needs to separately learn a model for each noise level, more recent plug-and-play denoisers $D : \mathbb{R}^n \times \mathbb{R}_{++} \to \mathbb{R}^n$ are built to take both the input image and the noise level σ as inputs. They are trained simultaneously to remove Gaussian noise of standard deviation σ for all σ in a given interval. For the rest of the manuscript, we denote $D(y, \sigma)$ as $D_\sigma(y)$.

It is important to remind that when plugged in a plug-and-play algorithm, the role of the denoiser is to regularize via the MAP/MMSE interpretation rather than to remove Gaussian noise. Indeed, at a certain iteration k, the image fed to the denoiser may even not contain any kind of Gaussian noise. Then, the noise-level parameter σ of the denoiser does not correspond to the noise level present in the input image. Instead, σ can first be interpreted as a regularization parameter. In particular, for RED algorithm (Section 2.4.2) $D_\sigma \approx \mathrm{Id} - \sigma^2 \nabla(-\log p_\sigma)$ and σ plays the role of a smoothing parameter for the regularizer $-\log p_\sigma$ in the objective function (2.97). The σ parameter could also

be interpreted as a stepsize. In particular, for PnP algorithm (Section 2.4.2), where the MAP interpretation gives $D_\sigma \approx \text{Prox}_{-\sigma^2 \log p_X}$. Therefore, for optimal performance, it is important for PnP/RED algorithms to plug denoisers built to take the σ parameter as a tunable parameter.

Deep denoisers are commonly trained to remove Gaussian noise by minimizing the L^2 loss, with noise level σ uniformly sampled in $[0, \sigma_{max}]$,

$$\arg\min_\theta \mathbb{E}_{x\sim p_X, \sigma\sim\mathcal{U}[0,\sigma_{max}], \xi_\sigma\sim\mathcal{N}(0,\sigma^2\,\mathrm{Id})} \left[\left|\left| D_\sigma^\theta(x+\xi_\sigma) - x \right|\right|^2 \right]. \tag{2.113}$$

It has also been observed (Zhang et al., 2021) that the L^1 loss instead of the L^2 loss can allow more stable training. It is common (Song and Ermon, 2019; Ho et al., 2020) to weight each squared L^2 norm in (2.113) by $\frac{1}{\sigma^2}$. [1]

2.4.4 Advantages, limits, and challenges of plug-and-play algorithms

Advantages The main advantage of using as regularization a Gaussian denoising prior is that Gaussian denoising an easy image inverse problems. It has been extensively studied theoretically and practically, and there exists a large variety of algorithms and models that have been developed for this specific task. Training deep Gaussian denoiser is often stable, and it does not require a large amount of training images in order to achieve good denoising performance (only 400 images are used to train DnCNN Zhang et al. (2017a)). Other deep priors such as GANs/VAEs can be much more difficult to train. Therefore, it is convenient to be able to exploit off-the-shelf denoisers for solving more complicated image recovery tasks.

Another advantage of the plug-and-play framework is its *flexibility*. The same model can be used to treat a large variety of degradations encoded in f. A related advantage is that the algorithm is *unsupervised* in the sense that it does not need to be trained on a specific task with ground-truth data. If applied with a deep denoiser, it only requires training a single network to remove artificial Gaussian noise on a dataset of clean images.

Moreover, among unsupervised methods, PnP and RED algorithms achieve *state-of-the-art results* (Meinhardt et al., 2017; Buzzard et al., 2018; Ahmad et al., 2020; Yuan et al., 2020; Zhang et al., 2021) in various image inverse problems. Eventually, even though this is out of the scope of this thesis, in order to reach optimal performance *with ground-truth training data*, a common extension consists in unfolding PnP/RED algorithms in a global trainable architecture (Adler and Öktem, 2018; Zhang et al., 2020). However, in practice unfolded architectures typically use a very limited number of iterations (less than 10) compared to PnP algorithms (hundreds of iterations).

Limitations. A generic denoiser D_σ *e.g.* a deep denoiser is not exactly the MAP nor the MMSE denoiser. In the general case, it cannot be written as a proximity operator Prox_g or as a gradient descent map $\text{Id} - \nabla g$. Therefore, the well established *convergence* of first order optimization algorithm do not generalize to PnP and RED algorithms when replacing a proximity operator or gradient descent map by the denoiser.

Moreover, the regularization is only made implicit via the denoising operation. Therefore, even if they converge, PnP and RED algorithms *do not seek the minimization of an explicit objective* functional in the form (2.94). This strongly limits the interpretation of

[1] $\mathbb{E}_{x\sim p_X, y\sim p_Y}$ signifies that x and y are independent random variables following the laws X and Y.

their output. It also impedes proper numerical control of the algorithm, as we cannot verify the decrease and the convergence of a potential.

The *numerical complexity* of the algorithm during evaluation is also quite high compared to supervised models. Indeed, it requires to evaluate the denoiser for hundreds of iterations.

Additionally, as explained in Section 2.4.1, we recall that with the Denoising Score Matching result (2.80), a deep denoiser trained by minimizing the L^2 loss (2.113) is actually trained to approximate the MMSE denoiser. However, PnP algorithms presented in Section 2.4.2 are originally derived assuming that the plugged denoiser approximates the true MAP estimator and not the MMSE estimator. In spite of this, PnP algorithm with deep denoisers trained with the L^2 loss still provide very good results and perform on par with RED algorithms. In our analysis, we will not need any assumption on the capacity of the denoiser to approximate the true MAP or MMSE estimators. However, it will be useful to keep in mind the Denoising Score Matching interpretation (2.80).

Chapter 3

Preliminary convergence results

Contents

In the following chapter, we delve into a more detailed exploration of the convergence analysis of general first-order optimization algorithms and plug-and-play schemes. We first introduce Section 3.1 a range of useful tools and concepts for addressing both convex and nonconvex convex convergence. We will then able to formulate in Section 3.2 the convergence theorems for each first-order optimization algorithm that was introduced in Section 2.3, addressing both the scenarios of convex and nonconvex settings. Most of the presented results are not from our contribution but were already published in the literature. In Section 3.3, we conduct an in-depth examination of the existing literature concerning the convergence of PnP and RED algorithms.

3.1 Tools and concepts for analyzing convergence

3.1.1 Inequalities for convex and smooth functions

Before going further, we introduce some basic inequalities verified by convex or smooth functions. These inequalities are the basic blocks that will be used in our different proofs of convergence. We consider in this section $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ proper and lower semi-continuous (lsc).

Convex functions First, from the definition of the subdifferential, we can show that a convex function verifies

Proposition 13. *If f is convex, for all $(x, y) \in \operatorname{dom} f \times \operatorname{dom} f$, and for all $z \in \partial f(y)$, we have*

$$f(x) \geqslant f(y) + \langle z, x - y \rangle. \tag{3.1}$$

We can then re-write this result for a weakly convex function and derive a new inequality on the proximity operator of a (weakly) convex function, called *Three-points inequality.*

Corollary 1. *If $x \to f(x) + \frac{\gamma}{2} ||x||^2$ is convex for some $\gamma \in \mathbb{R}$, then*

(i) For all $(x, y) \in \operatorname{dom} f \times \operatorname{dom} f$, and for all $z \in \partial f(y)$, we have

$$f(x) \geq f(y) + \langle z, x - y \rangle - \frac{\gamma}{2} ||x - y||^2 ; \tag{3.2}$$

*(ii) (**Three-points inequality**) For all $(x, y) \in \operatorname{dom} f \times \operatorname{dom} f$, and for all $z \in \operatorname{Prox}_f(y)$, we have*

$$f(x) + \frac{1}{2} ||x - y||^2 \geq f(z) + \frac{1}{2} ||z - y||^2 + \frac{1 - \gamma}{2} ||x - z||^2 . \tag{3.3}$$

Remark 2. *This encompasses the convex ($\gamma = 0$), strongly convex ($\gamma < 0$) and weakly convex ($\gamma > 0$) cases.*

Proof. (i) is direct from Proposition 13 applied to $x \to f(x) + \frac{\gamma}{2} ||x||^2$.

(ii) The optimality conditions of the proximal operator $z \in \operatorname{Prox}_f(y)$ gives

$$y - z \in \partial f(z). \tag{3.4}$$

Hence, by Proposition 13 applied with the convex function $x \to f(x) + \frac{\gamma}{2} ||x||^2$, we have $\forall x, y \in dom(f) \times dom(f)$,

$$\begin{aligned} f(x) + \frac{\gamma}{2} ||x||^2 &\geq f(z) + \frac{\gamma}{2} ||z||^2 + \langle y - z + \frac{\gamma}{2} z, x - z \rangle \\ &\Leftrightarrow f(x) \geq f(z) + \langle y - z, x - z \rangle - \frac{\gamma}{2} ||x - z||^2 \end{aligned} \tag{3.5}$$

and therefore adding $\frac{1}{2} ||x - y||^2$ on both side,

$$\begin{aligned} f(x) + \frac{1}{2} ||x - y||^2 &\geq f(z) + \frac{1}{2} ||x - y||^2 + \langle y - z, x - z \rangle - \frac{\gamma}{2} ||x - z||^2 \\ &= f(z) + \frac{1}{2} ||x - z||^2 + \frac{1}{2} ||y - z||^2 - \frac{\gamma}{2} ||x - z||^2 \\ &= f(z) + \frac{1}{2} ||y - z||^2 + \frac{1 - \gamma}{2} ||x - z||^2 . \end{aligned}$$

□

Smooth functions We now consider the case of f with gradient L-Lipschitz, which we refer to f L-smooth. First, if f is only differentiable (and not necessarily with Lipschitz gradient) then Proposition 13 turns to an equivalence:

Proposition 14 ((Bauschke and Combettes, 2011a, Proposition 17.10)). *Let $C \subseteq \operatorname{int}\operatorname{dom}(f)$ an open and convex subset of $\operatorname{int}\operatorname{dom}(f)$. If f is differentiable on C, then the following statements are equivalent*

(i) $x \to f(x) + \frac{\gamma}{2}\|x\|^2$ *is convex on C for some $\gamma \in \mathbb{R}$*

(ii) for all $(x, y) \in C$,

$$f(x) - f(y) - \langle \nabla f(y), x - y\rangle \geq -\frac{\gamma}{2}\|x - y\|^2 \tag{3.6}$$

If f is twice differentiable, this is also equivalent to

(iii) $\forall x \in C$, $\forall u \in \mathbb{R}^n$, $\langle \nabla^2 f(x)u, u\rangle \geq -\gamma\|u\|^2$

Moreover, if f has Lipschitz gradient on a convex set, without any convexity conditions, it verifies the following descent lemma.

Proposition 15 (Descent Lemma (Bauschke and Combettes, 2011a, Lemma 2.64)). *If f is L-smooth on an open and convex set $C \subseteq \operatorname{int}\operatorname{dom}(f)$, $\forall (x, y) \in C \times C$, we have*

$$|f(x) - f(y) - \langle \nabla f(y), x - y\rangle| \leq \frac{L}{2}\|y - x\|^2. \tag{3.7}$$

Combining Propositions 14 and 15, we can give the following results relating smoothness and convexity.

Corollary 2. *Let $C \subseteq \operatorname{int}\operatorname{dom}(f)$ an open and convex subset of $\operatorname{int}\operatorname{dom}(f)$. f is L-smooth on C , then both $x \to \frac{L}{2}\|x\|^2 + f(x)$ and $x \to \frac{L}{2}\|x\|^2 - f(x)$ are convex on C. If f is also convex, then we have the equivalence.*

Proof. ($\Longrightarrow$) From Propositions 15, f L-smooth implies that for all $x, y \in C$

$$-\left(f(x) - f(y) - \langle \nabla f(y), x - y\rangle\right) \leq \frac{L}{2}\|y - x\|^2 \tag{3.8}$$

$$\Leftrightarrow \quad f(x) - f(y) - \langle \nabla f(y), x - y\rangle \geq -\frac{L}{2}\|y - x\|^2 \tag{3.9}$$

which means, by Proposition 14, that $x \to f(x) + \frac{L}{2}\|x\|^2$ is convex. The convexity of $x \to -f(x) + \frac{L}{2}\|x\|^2$ is direct considering $-f$.

($\Longleftarrow$) (Combettes, 2018, Theorem 18.15) □

Remark 3. *Note that for analyzing the converge results of optimization algorithms, in most of the cases, only one direction of the descent lemma is necessary*

$$f(x) - f(y) - \langle \nabla f(y), x - y\rangle \leq \frac{L}{2}\|y - x\|^2 \tag{3.10}$$

This is equivalent (Proposition 14) to $\frac{L}{2}\|x\|^2 - f(x)$ being convex. While the other direction of the descent lemma

$$-(f(x) - f(y) - \langle \nabla f(y), x - y\rangle) \leq \frac{L}{2}\|y - x\|^2 \tag{3.11}$$

is equivalent to $\frac{L}{2}\|x\|^2 + f(x)$ being convex. Therefore, when f is nonconvex, in most of the cases, it is enough to only assume $\frac{L}{2}\|x\|^2 - f(x)$ convex. This is less restrictive than assuming f with L-Lipschitz gradient.

3.1.2 Kurdyka–Łojasiewicz property

The Kurdyka-Łojasiewicz (KŁ) property (Bolte et al., 2007) is a local property that characterizes the shape of the function f around its critical points $\{x \in int\,\mathrm{dom}(f), 0 \in \partial f(x)\}$. It is a tool widely used in nonconvex optimization (Attouch et al., 2010, 2013; Ochs et al., 2014; Bolte et al., 2018; Zeng et al., 2019). We use the definition from Attouch et al. (2010). In this section and in the rest of this chapter, ∂f stands for the limiting subdifferential (see Section 2.2.1 for more details).

Definition 5 (Kurdyka-Łojasiewicz (KŁ) property (Attouch et al., 2010))**.** *A function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is said to have the Kurdyka-Łojasiewicz property at $x^* \in dom(f)$ if there exists $\eta \in (0, +\infty)$, a neighborhood U of x^* and a continuous concave function $\phi : [0, \eta) \to \mathbb{R}_+$ such that $\phi(0) = 0$, ϕ is $\mathcal{C}^1$ on $(0, \eta)$, $\phi' > 0$ on $(0, \eta)$ and $\forall x \in U \cap [f(x^*) < f < f(x^*) + \eta]$, the* Kurdyka-Łojasiewicz inequality *holds:*

$$\phi'(f(x) - f(x^*))\,dist(0, \partial f(x)) \geq 1. \tag{3.12}$$

Proper lsc functions that satisfy the Kurdyka-Łojasiewicz inequality at each point of $dom(\partial f)$ are called KŁ functions.

Remark 4. *(i) This is a general definition, it is also common to directly define the desingularizing function ϕ as $\phi(s) = cs^{1-\theta}$ for some $c > 0$ and $\theta \in [0, 1)$ (Bolte et al., 2007).*

(ii) For f proper lsc, the KŁ inequality always holds at non-critical points $x^* \in \mathrm{dom}\,\partial f$ (Attouch et al., 2010, Remark 4.(b)).

(iii) For f smooth and $f(x^*) = 0$, the KŁ inequality (3.12) writes $||\nabla(\phi \circ f)(x)|| \geq 1$. The KŁ property can thus be interpreted as the fact that, up to a reparameterization, the function is locally sharp (Attouch et al., 2013).

The KŁ inequality was originally specifically derived by Kurdyka (Kurdyka, 1998) to extend to *tame functions* the "Łojasiewicz inequality" ($||\nabla f(x)|| \geq C\,||f(x) - f(x^*)||^\theta$ with $\theta < 1$) verified by *real analytic functions*. The class of *tame functions* generalizes the class of *real semialgebraic function*. The KŁ inequality was later extended to nonsmooth continuous subanalytic functions by Bolte et al. (2007).

Nonconvex convergence to a critical point with Kurdyka–Łojasiewicz property

We now explain why Kurdyka–Łojasiewicz functions are useful for nonconvex convergence. Given a KŁ function f, we want to show convergence of a sequence $(x_k)_{k\in\mathbb{N}}$, produced by a certain iterative procedure, towards a critical point of f. With the following theorem, it is proposed by Attouch et al. (2013) sufficient conditions on (x_k) such that this convergence is verified. This comprehensive theorem will be the basis of the nonconvex convergence analysis of first-order optimization algorithms in Section 2.3.

Theorem 4 (Attouch et al. (2013, Theorem 2.9))**.** *Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a proper lsc function. Consider a sequence $(x_k)_{k\in\mathbb{N}}$ satisfying the following conditions :*

- ***H1 : Sufficient decrease condition*** *$\forall k \in \mathbb{N}$,*

$$f(x_{k+1}) + a\,||x_{k+1} - x_k||^2 \leq f(x_k). \tag{3.13}$$

- ***H2 : Relative error condition*** $\forall k \in \mathbb{N}$, *there exists* $\omega_{k+1} \in \partial f(x_{k+1})$ *such that*

$$||\omega_{k+1}|| \leq b\,||x_{k+1} - x_k||\,. \tag{3.14}$$

- ***H3 : Continuity condition*** *There exists a subsequence* $(x_{k_i})_{i\in\mathbb{N}}$ *and* $\hat{x} \in \mathbb{R}^n$ *such that*

$$x_{k_i} \to \hat{x} \quad \textit{and} \quad f(x_{k_i}) \to f(\hat{x}) \quad \textit{as} \quad i \to +\infty. \tag{3.15}$$

If f *verifies the Kurdyka–Łojasiewicz property at the cluster point* $\hat{x}$ *specified in H3, then* (x_k) *converges to* $\hat{x}$ *as* $k \to +\infty$ *and* $\hat{x}$ *is a critical point of* f.

Remark 5. *It is proposed in Ochs et al. (2014) similar but slightly different conditions H1 and H2 involving* (x_n, x_{n+1}) *to ensure the same result for a two-step algorithm.*

The first condition H1 models the descent of the objective function f along the iterates. Note that if f is lower-bounded, with H1, as $f(x_k)$ decreases, we have convergence of the function values $f(x_k)$, as well as, by telescopic sum, $\sum ||x_{k+1} - x_k||^2 < +\infty$ *i.e.* the sequence has finite length. If f is also coercive, then $\{f(x) < f(x_0)\}$ is necessarily bounded and the iterates remain bounded. We can then extract from (x_k) a subsequence converging towards $\hat{x}$. With condition H2 and H3, we obtain $0 \in \partial f(\hat{x})$ i.e. the limit point is a critical point of f.

When dealing with a nonconvex function f, a sequence (x_k) satisfying the above conditions is not guaranteed to converge to a single point. It can be the case if f is flat or highly oscillating around its critical points. The KŁ property is a general and flexible condition that prevents the above cases and ensures single-point convergence of any sequence satisfying H1, H2 and H3.

How to verify that a function verifies the Kurdyka–Łojasiewicz property?

In Section 3.2, we will make use of this theorem to study the convergence of iterative algorithms for minimizing the sum of two functions $F = f + g$ with f or g nonconvex. The first requirement for this theorem is to show that F is KŁ. However, the KŁ condition is not stable by sum. Therefore, we will need to introduce conditions on f and g such that $f + g$ is KŁ.

Moreover, in the context of our analysis, g is a regularization term that will be typically parameterized by a neural network. Even if the set of KŁ functions is large in practice, verifying that a function, and more particularly a neural network, is KŁ is not easy. We need to choose a subclass of KŁ functions that, on the one hand, is large enough to encompass all our functions of interest and, on the other hand, has minimal stability properties so that inclusion to that set is easy to verify. In particular, for neural networks, stability by sum and composition are minimal requirements.

Subanalytic functions constitute a large enough set to include all our functions of interest. They include in particular *real analytic* functions and *semialgebraic* functions. We now define these three set of functions and give, in different lemmas, their most important stability properties.

Real analytic functions

Definition 6 (Real analytic)**.** *A function* $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ *with open domain is said to be (real) analytic at* $x \in \mathrm{dom}(f)$ *if* f *may be represented by a convergent power series*

on a neighborhood of x. The function is said to be analytic if it is analytic at each point of its domain. We can extend the definition for $f : \mathbb{R}^n \to \mathbb{R}^m$ which is analytic at $x \in \mathbb{R}^n$ if for $f = (f_1, ...f_m)$, all $f_i : \mathbb{R}^n \to \mathbb{R}$ are analytic at x.

Typical real analytic functions include polynomials, exponential functions, the logarithm, trigonometric and power functions. Moreover, real analytic functions have the following stability properties

Lemma 3 (Krantz and Parks (2002))**.** *(i) The sum, product, and composition of real analytic functions are real analytic.*

(ii) A partial derivative of f real analytic is real analytic.

(iii) ***Real Analytic Inverse Function Theorem*** *Let $f : \mathbb{R}^n \to \mathbb{R}^m$ be real analytic in a neighborhood of $x \in \mathbb{R}^n$ and suppose $J_f(x)$ non-singular, then f^{-1} is defined and is real analytic in a neighborhood of $f(x)$. In particular, if f is real analytic and $\forall x \in \mathbb{R}^n$, $J_f(x)$ is non-singular, then f^{-1} is real analytic on $Im(f)$.*

Semialgebraic functions

Definition 7 (Semialgebraic Attouch et al. (2010))**.**

- *A subset S of $\mathbb{R}^n$ is a real semialgebraic set if there exists a finite number of real polynomial functions $P_{i,j}, Q_{i,j} : \mathbb{R}^n \to \mathbb{R}$ such that*

$$S = \cup_{j=1}^{p} \cap_{i=1}^{q} \{x \in \mathbb{R}^n, P_{i,j} = 0, Q_{i,j} < 0\} \tag{3.16}$$

- *A function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ (resp. $f : \mathbb{R}^n \to \mathbb{R}^m$) is called semialgebraic if its graph $\{(x, y) \in \mathbb{R}^n \times \mathbb{R}, y = f(x)\}$ (resp. $\{(x, y) \in \mathbb{R}^n \times \mathbb{R}^m, y = f(x)\}$) is a semialgebraic subset of $\mathbb{R}^n \times \mathbb{R}$ (resp. $\mathbb{R}^n \times \mathbb{R}^m$).*

From Definition 7, we first easily verify that polynomial functions are semialgebraic functions. Some other typical semialgebraic maps are the indicator function of a semialgebraic set or the Euclidean norm.

The set of semialgebraic functions has strong stability properties. The main ones follow from the Tarski-Seidenberg principle, which states that the image of a semialgebraic subset of $\mathbb{R}^{n+m}$ by projection on the first n coordinates is a semialgebraic subset of $\mathbb{R}^n$. From this theorem, the following holds.

Lemma 4 (Perrin (2020))**.** *(i) The sum, product, and composition of semialgebraic functions are semialgebraic functions.*

(ii) If a partial derivative of f semialgebraic exists, then it is semialgebraic.

(iii) The image and preimage of a semialgebraic set by a semialgebraic function is a semialgebraic set.

Subanalytic functions Similar to semialgebraicity, in order to define subanalytic functions, we need to define first semianalytic and subanalytic sets. A semianalytic set is defined locally by an expression of the form (3.16) but where the polynomials are replaced by general real analytic functions. A subanalytic set is locally the projection of a bounded semianalytic set.

Definition 8 (Subanalytic Bolte et al. (2007))**.**

- *A subset S of $\mathbb{R}^n$ is a semianalytic set if each point of $\mathbb{R}^n$ admits a neighborhood V for which there exists a finite number of real analytic functions $f_{i,j}$, $g_{i,j} : \mathbb{R}^n \to \mathbb{R}$ such that*
$$S \cap V = \cup_{j=1}^{p} \cap_{i=1}^{q} \{x \in \mathbb{R}^n, f_{i,j} = 0, g_{i,j} < 0\} \tag{3.17}$$
- *A subset S of $\mathbb{R}^n$ is a subanalytic set if each point of $\mathbb{R}^n$ admit a neighborhood V for which*
$$S \cap V = \{x \in \mathbb{R}^n, (x, y) \in U\} \tag{3.18}$$
where U is a bounded semianalytic subset of $\mathbb{R}^n \times \mathbb{R}^p$ with $p \geq 1$.
- *A function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ (resp. $f : \mathbb{R}^n \to \mathbb{R}^m$) is called subanalytic if its graph $\{(x, y) \in \mathbb{R}^n \times \mathbb{R}, y = f(x)\}$ (resp. $\{(x, y) \in \mathbb{R}^n \times \mathbb{R}^m, y = f(x)\}$) is a subanalytic subset of $\mathbb{R}^n \times \mathbb{R}$ (resp. $\mathbb{R}^n \times \mathbb{R}^m$).*

Subanalytic functions encompass both analytic functions and semialgebraic functions (Shiota, 2012). However, they do not enjoy the same stability results. In particular, there is not such Tarski-Seidenberg principle for the stability via projection from a general unbounded subanalytic set. However, by definition, the result holds for bounded sets, and we still have the following stability results

Lemma 5 (Shiota (2012))**.** *(i) The sum of two subanalytic functions is subanalytic if at least one of the two functions maps a bounded set to a bounded set or if both functions are non-negative.*

(ii) The product of two subanalytic functions is subanalytic if both functions map a bounded set to a bounded set.

*(iii) The composition $g \circ f$ with g subanalytic is subanalytic if f **or** g^{-1} maps a bounded set to a bounded set.*

Finally, the important point is that continuous subanalytic functions are KŁ:

Lemma 6 (Bolte et al. (2007, Theorem 3.1))**.** *Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a proper, subanalytic function and assume that f is continuous on its domain, then f is a KŁ function.*

We come back to our practical case of interest, i.e. the minimization of $F = f + g$ where f is a data-fidelity term and g a deep regularization. Given the strong stability properties of semialgebraicity, nonconvex optimization papers (Attouch et al., 2013; Bolte et al., 2018) often work directly with semialgebraic functions to verify that an objective verifies the KŁ property on its domain. Typically, some optimization methods directly assume f and g semialgebraic. In our analysis, the regularizer g will be parameterized by a neural network, which basically consists of composition and sums of linear maps and activation functions. A linear map being clearly semialgebraic, the main difficulty is to

show that activation functions are semialgebraic. This is the case of the ReLU which can be expressed with a system of polynomial (in)equalities:

$$\text{for } x, y \in \mathbb{R},\ y = \mathrm{ReLU}(x) = \max(0, x) \text{ if and only if } y(y - x) = 0, y \geq x, y \geq 0. \quad (3.19)$$

However, it is not true anymore for other common activation functions, like softplus

$$s(x) = \frac{1}{\alpha} \log(1 + \exp(\alpha x)). \quad (3.20)$$

Softplus is however clearly analytic. Actually, as shown by Zeng et al. (2019), most of the activation functions are semialgebraic or real analytic. A linear map being also analytic and semialgebraic, by stability of these two notions by sum and composition, it is easy to show that, in general, neural networks are either semialgebraic or real analytic and thus subanalytic.

As f commonly measures a distance to an input, it is non-negative. As g is assumed lower-bounded, we might assume g to be non-negative as well, even if it means adding a constant to the global objective F. Thus, with Lemma 5 it will be enough to assume f subanalytic in order to show that F is KŁ.

3.1.3 Fixed-point theory

In order to study the convergence of first order optimization algorithms and of plug-and-play algorithms, it will be useful to formulate these iterative algorithms as fixed-point algorithms. Convergence will then follow from fixed-point convergence theory.

In the following, we consider a general operator $T : E \to E$ defined on $E \subset \mathbb{R}^n$. **We assume that** $\mathrm{Fix}(T)$**, the set of fixed-point of** T **is non-empty**. In this section, we will review sufficient conditions on T to ensure that the fixed-point iterations

$$x_{k+1} = T(x_k) \quad (3.21)$$

converge towards a fixed-point of T. As soon as T is L-Lipschitz with $L < 1$, it is a contraction and Banach's fixed-point theorem states that there exists a unique fixed-point of T and that the scheme (3.21) converges to this fixed-point. When T is only 1-Lipschitz (that is *nonexpansive*), convergence to a fixed-point can nevertheless be ensured using a *Krasnosel'skii-Mann* relaxation of the iterates. They take the form, for some positive sequence $(\nu_k)_k$

$$x_{k+1} = (\nu_k T + (1 - \nu_k)\,\mathrm{Id})(x_k). \quad (3.22)$$

Convergence of (3.22) is given by the following theorem.

Theorem 5 (Krasnonel'skii-Mann iterations (Bauschke and Combettes, 2011a, Theorem 5.15))**.** *Let T be a nonexpansive operator that admits fixed-points and (ν_k) be a sequence in $[0, 1]$ such that $\sum_k \nu_k(1 - \nu_k) = +\infty$. Then the sequence (3.22) converges to a fixed-point of T and verifies $\sum \nu_k(1 - \nu_k)\,||x_{k+1} - x_k||^2 < +\infty$.*

Given this result, a convenient notion for showing convergence of fixed-point iterations of the form of (3.21) with T non-contractive is the notion of θ**-averaged operators**.

Definition 9 (θ-averaged operator)**.** *T is θ-averaged for $\theta \in (0, 1)$ if $T = \theta R + (1 - \theta)\,\mathrm{Id}$ with R nonexpansive. In particular, an averaged operator is nonexpansive. T is said firmly nonexpansive if it is $\frac{1}{2}$-averaged.*

The following result states convergence of the fixed-point iterations for averaged operators.

Corollary 3. *Let $\theta \in (0,1)$. Let T be a θ-averaged operator that admits fixed-points. Then the sequence $x_{k+1} = T(x_k)$ converges to a fixed-point of T and verifies $\sum ||x_{k+1} - x_k||^2 < +\infty$.*

Proof. Using $T = \theta R + (1-\theta)\,\mathrm{Id}$ with R nonexpansive, the fixed-point of T and R coincide. We get the result by directly applying Theorem 5 with the operator R and fixed $\nu_k = \theta$. □

We now outline some useful properties for establishing the averageness of an operator. First, using Definition 9, we immediately get the following result on the relaxation of θ-averaged operators.

Proposition 16. *Let $\theta > 0$ such that T writes $T = \theta R + (1-\theta)\,\mathrm{Id}$ with R a nonexpansive operator. In particular, if $\theta < 1$, T is θ-averaged. Then for $\alpha \in (0, \frac{1}{\theta})$, $\alpha T + (1-\alpha)\,\mathrm{Id}$ is $\alpha\theta$-averaged.*

We will also make use of the following result showing that the composition and convex combination of averaged operators are averaged.

Proposition 17 ((Bauschke and Combettes, 2011a, Proposition 4.42 and 1.44))**.** *Let $T_1 : E \to E$ and $T_2 : E \to E$, if T_1 and T_2 are respectively θ_1 and θ_2-averaged, then*

(i) $T_1 \circ T_2$ is θ-averaged with $\theta = \frac{\theta_1 + \theta_2 - 2\theta_1\theta_2}{1-\theta_1\theta_2}$.

(ii) For $\alpha \in (0,1)$, $\alpha T_1 + (1-\alpha)T_2$ is $\alpha\theta_1 + (1-\alpha)\theta_2$-averaged.

Application to descent mappings We now make explicit the link between fixed-point operator theory and convex optimization. A step of gradient descent and the proximity operator of a convex function are both averaged operators. Moreover, if the function is strongly convex, then they are contractive.

Proposition 18. *Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a proper lsc convex function,*

(i) for $\tau > 0$, $\mathrm{Prox}_{\tau f}$ is firmly nonexpansive (Bauschke and Combettes, 2011a, Corollary 23.8).

(ii) f is L-smooth $\Leftrightarrow$ f differentiable and $\mathrm{Id} - \frac{2}{L}\nabla f$ is nonexpansive $\Leftrightarrow$ f differentiable and $\forall \tau \in \left(0, \frac{2}{L}\right)$, $\mathrm{Id} - \tau\nabla f$ is $\frac{\tau L}{2}$-averaged.

Remark 6. *This is not true any more for a nonconvex f.*

Proof. (ii) $\forall \tau \in \left(0, \frac{2}{L}\right)$, $T = \mathrm{Id} - \tau\nabla f$ is $\frac{\tau L}{2}$-averaged $\Leftrightarrow \forall \tau \in \left(0, \frac{2}{L}\right)$, $\frac{2}{\tau L}T + (1 - \frac{2}{\tau L})\,\mathrm{Id} = \mathrm{Id} - \frac{2}{L}\nabla f$ is nonexpansive i.e. $\forall x, y$

$$\left|\left|(x-y) - \frac{2}{L}(\nabla f(x) - \nabla f(y))\right|\right|^2 \leq ||x-y||^2 \tag{3.23}$$

or, by expanding the left term, $\forall x, y$

$$||\nabla f(x) - \nabla f(y)||^2 \leq L\langle x - y, \nabla f(x) - \nabla f(y)\rangle \tag{3.24}$$

i.e. ∇f is $1/L$ *co-coercive*. For convex f, this is equivalent to ∇f being L-Lipschitz (Bauschke and Combettes, 2011a, Theorem 18.15). □

Proposition 19. *Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a proper lsc μ-strongly convex function and $\tau > 0$*

(i) $\mathrm{Prox}_{\tau f}$ *is* $\frac{1}{1+\tau\mu}$*- Lipschitz.*

(ii) If f is L-smooth, then $\mathrm{Id} - \tau\nabla f$ *is* $\max(|1-\tau\mu|, |1-\tau L|)$*-Lipschitz (Ryu et al., 2019).*

3.2 Convergence of first order optimization algorithms

In order to analyze and understand the convergence of plug-and-play algorithms, we need to dive more precisely into the convergence theory of the first-order optimization algorithms presented in Section 2.3. Recall that PGD, DRS, ADMM or PD target the minimization of the sum of two functions

$$\min_{x\in\mathbb{R}^n} F(x) = f(x) + g(x) \tag{3.25}$$

with $f, g : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ proper, lower semi-continuous and lower-bounded. For each algorithm, we will prove convergence in the convex (both f and g are convex) and nonconvex (both f and g can be nonconvex) settings. In the nonconvex case, instead of looking for a global or local minimizer, we will rather look for a *critical point* i.e. a point x such that $0 \in \partial F(x)$. In this section, ∂F stands for the limiting subdifferential (see Section 2.2.1) $\partial^{lim} F$ (Equation 2.26). Recall that for F convex, the limiting subdifferential falls back to the standard convex subdifferential (Definition 2).

3.2.1 Proximal Gradient Descent (PGD)

We first analyze the convergence of the PGD algorithm

$$\textbf{(PGD)} \quad x_{k+1} \in \mathrm{Prox}_{\tau g} \circ (x_k - \tau\nabla f(x_k)) \,. \tag{3.26}$$

The convergence of the Gradient Descent (2.50) and Proximal Point (2.49) algorithms, which are particular cases of PGD, will follow immediately.

Convex case

We first assume that both f and g are convex and that f is L_f-smooth (*i.e.* ∇f is L_f-Lipschitz). In this context, convergence of the PGD algorithm (3.26) can be easily derived using the fixed-point convergence of averaged operators derived in Section 3.1.3. From the fact that the gradient descent and proximity mappings are averaged (Proposition 18) and the fact that the composition of averaged operators is averaged (Proposition 17), we get that the PGD fixed-point operator is averaged:

Corollary 4. *For $\tau < \frac{2}{L_f}$, $T_{PGD} = \mathrm{Prox}_{\tau g} \circ (\mathrm{Id} - \tau\nabla f)$ is averaged.*

Using Theorem 3, we can directly conclude on the convergence of the PGD algorithms with convex functions.

Theorem 6 (Convergence of convex PGD)**.** *Assume f and g proper, lsc, convex, lower-bounded and f L_f-smooth. Then for $\tau < \frac{2}{L_f}$, the iterates given by the PGD algorithm* (3.26) *converge towards a global minimizer of F.*

Proof. F being lower-bounded, it admits a global minimum. From Proposition 8, for all $\tau > 0$, $\text{Fix}(T_{PGD}) = \text{zeros}(\partial F) = \arg\min F \neq \emptyset$. Using the fact that for $\tau < \frac{2}{L_f}$, T_{PGD} is averaged (Corollary 4) along with the convergence of fixed-point iterations of averaged operators (Theorem 3), we get the desired result. □

Nonconvex case

When $F = f + g$ is nonconvex, convergence towards a global minimum is out of reach. However, we can still target a critical point of the objective F. Simply assuming that f has Lipschitz-gradient, the descent lemma (Proposition 15) permits to derive a sufficient decrease of the objective function $F = f + g$ along the iterates. If F is also lower-bounded, we get convergence of the function values and of the norm of difference between two iterates towards 0. At this point, we can also show that any converging subsequence does converge towards a stationary point of F.

Proposition 20 (First convergence results for nonconvex PGD.)**.** *Assume f and g proper, lsc, bounded from below, with f L_f-smooth. Then, for $\tau < 1/L_f$, the iterates (x_k) given by the PGD algorithm* (3.26) *verify*

(i) $(F(x_k))$ is non-increasing and converges.

(ii) The sequence has finite length, i.e. $\sum_{k=0}^{+\infty} ||x_{k+1} - x_k||^2 < +\infty$ and $||x_{k+1} - x_k||$ converges to 0 at rate $\min_{k<K} ||x_{k+1} - x_k|| = \mathcal{O}(1/\sqrt{K})$

(iii) All cluster points of the sequence (x_k) are critical points of F.

Proof. Although the arguments of this result are standard and can be found in various works (Beck and Teboulle, 2009b; Attouch et al., 2013; Ochs et al., 2014; Bolte et al., 2018), we here give the full proof for the sake of completeness.

(i) We first reformulate the PGD iterates as

$$x_{k+1} \in \text{Prox}_{\tau g} \circ (x_k - \tau \nabla f(x_k)) \tag{3.27}$$

$$\Leftrightarrow \quad x_{k+1} \in \arg\min_x g(x) + \frac{1}{2\tau} ||x - (x_k - \tau \nabla f(x_k))||^2 \tag{3.28}$$

$$\Leftrightarrow \quad x_{k+1} \in \arg\min_x g(x) + \langle x - x_k, \nabla f(x_k) \rangle + \frac{1}{2\tau} ||x - x_k||^2 . \tag{3.29}$$

Hence by evaluating the right-hand side at x_{k+1} and x_k, and adding $f(x_k)$ on both sides, we get

$$f(x_k) + g(x_{k+1}) + \langle x_{k+1} - x_k, \nabla f(x_k) \rangle + \frac{1}{2\tau} ||x_{k+1} - x_k||^2 \leq f(x_k) + g(x_k) = F(x_k). \tag{3.30}$$

Then, using the descent lemma (Proposition 15)

$$f(x_k) + \langle x_{k+1} - x_k, \nabla f(x_k) \rangle + \frac{1}{2\tau} ||x_{k+1} - x_k||^2 \tag{3.31}$$

$$= f(x_k) + \langle x_{k+1} - x_k, \nabla f(x_k) \rangle + \frac{L_f}{2} ||x_{k+1} - x_k||^2 + \left(\frac{1}{2\tau} - \frac{L_f}{2} \right) ||x_{k+1} - x_k||^2 \tag{3.32}$$

$$\geq f(x_{k+1}) + \left(\frac{1}{2\tau} - \frac{L_f}{2} \right) ||x_{k+1} - x_k||^2 . \tag{3.33}$$

Leading to the sufficient decrease equation

$$F(x_k) \geq F(x_{k+1}) + \left(\frac{1}{2\tau} - \frac{L_f}{2}\right) ||x_{k+1} - x_k||^2 . \tag{3.34}$$

For $\tau < \frac{1}{L_f}$, (3.34) together with the fact that F is assumed lower bounded proves (i).

(ii) Denoting by F^* the limit of $F(x_k)$, summing (3.34) between $k = 0$ and $k = K - 1$ gives

$$\sum_{k=0}^{k-1} ||x_{k+1} - x_k||^2 \leq \frac{1}{\left(\frac{1}{2\tau} - \frac{L_f}{2}\right)} (F(x_0) - F(x_K)) \leq \frac{1}{\left(\frac{1}{2\tau} - \frac{L_f}{2}\right)} (F(x_0) - F^*) \tag{3.35}$$

which proves that $||x_{k+1} - x_k||^2$ is summable and thus converges to 0. Additionally,

$$\min_{k<K} ||x_{k+1} - x_k||^2 \leq \frac{1}{K} \sum_{k=0}^{k-1} ||x_{k+1} - x_k||^2 = \mathcal{O}(1/K). \tag{3.36}$$

(iii) Suppose that a subsequence $(x_{k_i})_i$ is converging towards x. Let us show that x is a critical point of F. By optimality of the proximal operator of g, for all $k \geq 0$

$$\frac{x_{k_i+1} - x_{k_i}}{\tau} - \nabla f(x_{k+1}) \in \partial g(x_{k_i+1}). \tag{3.37}$$

where ∂g stands for the limiting subdifferential (2.26). In other words, for $\omega_{k_i} = \frac{x_{k_i} - x_{k_i-1}}{\tau} - \nabla f(x_{k_i})$, $(\omega_{k_i}, x_{k_i}) \in \operatorname{Graph} \partial g$. From the continuity of ∇f, we have $\nabla f(x_{k_i}) \to \nabla f(x)$. As $||x_{k_i+1} - x_{k_i}|| \to 0$, we get

$$\frac{x_{k_i} - x_{k_i-1}}{\tau} - \nabla f(x_{k_i}) \to -\nabla f(x). \tag{3.38}$$

By closeness of ∂g (Proposition 2), if we can show that $g(x_{k_i}) \to g(x)$, we get $-\nabla f(x) \in \partial g(x)$ i.e. x is a critical point of F (with Proposition 1). We now show that

$$g(x_{k_i}) \to g(x). \tag{3.39}$$

Using the fact that g is lsc we first have

$$\liminf_{i \to \infty} g(x_{k_i}) \geq g(x). \tag{3.40}$$

On the other hand, by optimality of (3.29),

$$\begin{aligned} g(x_{k_i+1}) + \langle x_{k_i+1} - x_{k_i}, \nabla f(x_{k_i}) \rangle + \frac{1}{2\tau} ||x_{k_i+1} - x_{k_i}||^2 \\ \leq g(x) + \langle x - x_{k_i}, \nabla f(x_{k_i}) \rangle + \frac{1}{2\tau} ||x - x_{k_i}||^2 . \end{aligned} \tag{3.41}$$

Using that $x_{k_i} \to x$ and $x_{k_i+1} - x_{k_i} \to 0$ when $i \to +\infty$, we get

$$\limsup_{i \to \infty} g(x_{k_i}) \leq g(x), \tag{3.42}$$

and

$$\lim_{i \to \infty} g(x_{k_i}) = g(x). \tag{3.43}$$

□

To go further, we can use the abstract nonconvex convergence result of Theorem 4, with the Kurdyka–Łojasiewicz (KŁ) property (Refer to Section 3.1.2 for more details on the KŁ property). Using this result, we now show that if $F = f + g$ is KŁ and the sequence generated by the PGD algorithm is bounded, the sequence converges towards a critical point of the objective function.

Theorem 7 (Single point convergence of nonconvex PGD.)**.** *Assume f and g proper, lsc, lower-bounded, with f L_f-smooth and $F = f + g$ KŁ. Then, for $\tau < 1/L_f$, if the iterates (x_k) given by the PGD algorithm* (3.26) *are bounded, then they converge towards a critical point of $F = f + g$.*

Proof. This result is a direct application of the general nonconvex convergence result from Theorem 4. We need to verify its assumptions **H1**, **H2** and **H3**. All three assumptions were verified in the proof of the previous Proposition 20:

- **H1** was verified with (3.34).
- **H2** was verified with (3.37). Indeed, we showed that $\omega_k = \frac{x_{k+1}-x_k}{\tau} \in \partial g(x_{k+1}) + \nabla f(x_{k+1}) = \partial F(x_{k+1})$.
- **H3**: The iterates (x_k) are assumed bounded. Thus, there exists a subsequence $(x_{k_i})_{i\in\mathbb{N}}$ converging towards $\hat{x} \in \mathbb{R}^n$ as $i \to +\infty$. In the proof of Proposition 20 (iii) it was shown that for such a subsequence $\lim_{i\to\infty} g(x_{k_i}) = g(\hat{x})$. By continuity of f, we get $\lim_{i\to\infty} F(x_{k_i}) = F(\hat{x})$.

□

Remark 7. *(i) The boundedness of the iterates is verified as soon as the objective F is **coercive**. Indeed, it ensures that $\{F(x) \leq F(x_0)\}$ is bounded and, since $F(x_k)$ is non-increasing (Proposition 20(i)), that the iterates remain bounded.*

(ii) We explained in Section 3.1.2 that, in practice, the Kurdyka–Łojasiewicz (KŁ) property is verified by a very large class of functions and can be checked using the notions of semialgebraicity or subanalyticity.

Remark 8. *In both Proposition 20 and Theorem 7, instead of assuming f with L_f-Lipschitz gradient, we get the exact same results by assuming the weaker condition*

$$f \text{ of class } \mathcal{C}^1 \text{ and } x \to \frac{1}{2}||x||^2 - f(x) \text{ convex.} \tag{3.44}$$

As explained in Remark 3, this is a sufficient condition for the descent Lemma (applied Equation (3.33)*) to be true. The res of the proof follows identically.*

Weakly convex - Nonconvex case

When the function under the Prox (here g) is assumed M-weakly convex, one can relax the condition on the stepsize from $\tau < 1/L_f$ to $\tau < 2/(L_f + M)$. This result will be useful in Chapter 5.

Theorem 8 (Convergence of PGD for weakly convex g)**.** *Assume f and g proper, lsc, lower-bounded, with f L_f-smooth and g M-weakly convex. Then, for $\tau < \max(\frac{2}{L_f+M}, \frac{1}{L_f})$, the properties (i)-(iii) from Proposition 20 hold and if F is KŁ and the iterates are bounded, then they converge towards a critical point of $F = f + g$.*

Remark 9. *In particular, if g is convex, the condition on the stepsize becomes $\tau < 2/L_f$.*

Proof. Compared to the proofs of Proposition 20 and Theorem 7, we only need to modify the sufficient decrease condition (3.34) so that the condition on the stepsize changes. The optimality condition of (3.29) gives

$$\frac{x_k - x_{k+1}}{\tau} - \nabla f(x_k) \in \partial g(x_{k+1}). \tag{3.45}$$

As g is M-weakly convex, Corollary 1 (i) gives that for $z \in \partial g(x_{k+1})$

$$g(x_k) \geq g(x_{k+1}) + \langle z, x_k - x_{k+1}\rangle - \frac{\gamma}{2}\left|\left|x_k - x_{k+1}\right|\right|^2; \tag{3.46}$$

Using (3.45), we get

$$g(x_k) \geq g(x_{k+1}) + \frac{\left|\left|x_k - x_{k+1}\right|\right|^2}{\tau} + \langle \nabla f(x_k), x_{k+1} - x_k\rangle - \frac{M}{2}\left|\left|x_k - x_{k+1}\right|\right|^2. \tag{3.47}$$

Using the descent lemma (Proposition 15)

$$f(x_{k+1}) \leq f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k\rangle + \frac{L_f}{2}\left|\left|x_k - x_{k+1}\right|\right|^2. \tag{3.48}$$

Combining both inequalities, for $F = f + g$, we obtain

$$F(x_k) \geq F(x_{k+1}) + \left(\frac{1}{\tau} - \frac{L_f + M}{2}\right)\left|\left|x_k - x_{k+1}\right|\right|^2. \tag{3.49}$$

Using $\tau < \frac{2}{L_f + M}$, the rest of the proofs from Proposition 20 and Theorem 7 hold. Note that when $M \geq L$, the stepsize condition from the fully nonconvex convergence results of Proposition 20 and Theorem 7 $\tau < 1/L_f$ is less restrictive. We thus keep the condition $\tau < \max(\frac{2}{L_f+M}, \frac{1}{L_f})$. □

Remark 10. *Once again, the smoothness condition on f can be replaced by*

$$f \text{ of class } \mathcal{C}^1 \text{ and } x \to \frac{1}{2}\left|\left|x\right|\right|^2 - f(x) \text{ convex.} \tag{3.50}$$

3.2.2 Douglas Rachford Splitting (DRS) / ADMM

We remind the form of the DRS and ADMM algorithms already introduced in Section 2.3.3.

$$\textbf{(DRS)} \quad \begin{cases} y_{k+1} & \in \operatorname{Prox}_{\tau f}(x_k) \\ z_{k+1} & \in \operatorname{Prox}_{\tau g}(2y_{k+1} - x_k) \\ x_{k+1} & = x_k + 2\beta(z_{k+1} - y_{k+1}) \end{cases} \tag{3.51}$$

$$\textbf{(ADMM)} \quad \begin{cases} z_{k+1} & \in \operatorname{Prox}_{\tau g}(y_k - u_k) \\ y_{k+1} & \in \operatorname{Prox}_{\tau f}(u_k + z_{k+1}) \\ u_{k+1} & = u_k + z_{k+1} - y_{k+1} \end{cases} \tag{3.52}$$

With Lemma 1 we showed the equivalence between DRS (with $\beta = 1/2$) and ADMM. In this section, we study the convergence of the DRS algorithm (3.51) for $\beta \in (0, 1]$. The convergence results of ADMM (2.63) will therefore follow immediately. We follow the same kind of analysis as the one done in the previous section for PGD. Remind that the DRS iterates (3.51) also write as

$$x_{k+1} \in \left(\beta \operatorname{Rprox}_{\tau g} \circ \operatorname{Rprox}_{\tau f} + (1-\beta)\operatorname{Id}\right)(x_k) \tag{3.53}$$

Convex case

We first assume that both f and g are convex. Once again, in this context, we can show convergence of DRS using the fixed-point convergence results derived in Section 3.1.3. From Proposition 9, with the notations of (3.51), if we show that the DRS fixed-point iterations converge, then (y_k) converges towards a critical point of F.

Theorem 9 (Convergence of convex DRS)**.** *Assume f and g proper, lsc, lower-bounded, convex. Then for $\tau > 0$ and $\beta \in (0, 1]$, the iterates $(y_k)_k$ given by the DRS algorithm* (3.51) *converge towards a global minimizer of F.*

Proof. For $\tau > 0$, we denote $T_\tau = \text{Rprox}_{\tau g} \circ \text{Rprox}_{\tau f}$. As $F = f + g$ is convex and lower bounded, $\text{zeros}(\partial F) \neq \emptyset$ and by Proposition 8, $\text{Fix}(T_\tau) \neq \emptyset$. Moreover, the Prox of a convex function being firmly nonexpansive (*i.e.* $\frac{1}{2}$-averaged), by definition, its Rprox is nonexpansive. T_τ is then nonexpansive by composition of nonexpansive operators. Using Theorem 5 with fixed $\mu_k = \beta$, we get that the fixed-point iterations given by (3.53) converge towards a fixed-point of T_τ. By Proposition 8, the iterates y_k from (2.62) then converge towards a critical point of F i.e. a global minimum. □

Remark 11. *Using Theorem 5 in its most general form, DRS can be extended with iteration-dependent relaxation parameters β_k in $(0, 1]$ such that $\sum_k \beta_k(1 - \beta_k) = +\infty$.*

Remark 12. *By fixed-point convergence, the iterates $(x_k)_k$ in* (2.61) *verify $||x_{k+1} - x_k|| \to 0$. However, via the form* (2.62) *of DRS, $||x_{k+1} - x_k|| = \beta\, ||z_{k+1} - y_{k+1}||$. Thus, the iterates $(z_k)_k$ in* (2.62) *also converge towards the same minimizer of F.*

Nonconvex case

Li and Pong (2016) and Themelis and Patrinos (2020) propose convergence proofs of the DRS algorithm for the minimization of the sum of two nonconvex functions $F = f + g$, one of the two functions (here f) being differentiable with Lipschitz gradient L_f. Themelis and Patrinos (2020) generalize the result from Li and Pong (2016) with a less restrictive stepsize condition. Both consider as Lyapunov function the *Douglas-Rachford envelope* (Themelis and Patrinos, 2020) (or *Douglas-Rachford merit function* (Li and Pong, 2016))

$$F_\tau^{DR}(x, y, z) = f(y) + g(z) + \frac{1}{\tau}\langle y - x, y - z\rangle + \frac{1}{2\tau}\,||y - z||^2 \tag{3.54}$$

Similar to the PGD convergence analysis realized in Section 2.3.1, two convergence results can be derived. First, under no additional assumption, one can show convergence of the Douglas-Rachford envelope $F_\tau^{DR}(x_k, y_k, z_k)$ along the iterates and convergence to 0 of $||x_{k+1} - x_k|| = \beta\, ||y_{k+1} - z_{k+1}||$. Second, invoking the Kurdyka–Łojasiewicz (KŁ) property (Section 3.1.2), we get convergence towards a critical point of the objective function. Both results are encompassed in the following theorem.

Theorem 10 (Li and Pong (2016); Themelis and Patrinos (2020))**.** *Assume that f and g are proper, lsc, lower-bounded and that f is L_f-smooth and M_f-weakly convex. Then, for a stepsize*

$$0 < \tau < \min\left(\frac{1 - \beta}{M_f}, \frac{1}{L_f}\right), \tag{3.55}$$

the sequence (x_k, y_k, z_k) generated by the DRS algorithm (2.62) *verifies*

(i) $F_\tau^{DR}(x_{k-1}, y_k, z_k)$ *is non-increasing and converges (Themelis and Patrinos, 2020, Theorem 4.1)*

(ii) $x_k - x_{k-1} = \beta(y_k - z_k)$ *tends to* 0 *with rate* $\min_{k\leq K} ||y_k - z_k|| = \mathcal{O}(\frac{1}{\sqrt{K}})$. *(Themelis and Patrinos, 2020, Theorem 4.2)*

(iii) (y_k) *and* (z_k) *have the same cluster points, which are critical points of* F. *(Themelis and Patrinos, 2020, Theorem 4.2)*

(iv) If the sequence (x_k, y_k, z_k) *is bounded, and if* F_τ^{DR} *is KŁ, then the sequences* (y_k) *and* (z_k) *converge to the same critical point of* F. *(Li and Pong, 2016, Theorem 2)*

Remark 13. *(i) We can assume* f *weakly convex without loss of generality. Indeed, as* f *is assumed* L_f*-smooth, by Proposition 2,* f *is at least* L_f*-weakly convex and we have necessarily* $M_f \leq L_f$.

(ii) Point (iv) follows from (Li and Pong, 2016, Theorem 2). In this theorem, instead of assuming that F_τ^{DR} *is KŁ, it is directly assumed that* f *and* g *are semialgebraic, which implies that* F_τ^{DR} *is KŁ. We prefer to keep a more general result, as this theorem will be used in Chapter 5 with non semialgebraic functions.*

(iii) Contrary to what we had with PGD, for DRS here the decreasing function is not F *itself but the Liapunov* F^{DR}. *As is shown in (Li and Pong, 2016, Theorem 4), if* f *or* g *are coercive,* F^{DR} *is coercive and the iterates remain bounded.*

3.2.3 Primal-Dual

We now analyze the convergence of the Primal-Dual algorithms introduced in Section 2.3.4.

Convex case

For convex f and g, we recall the form of the Primal-Dual Chambolle-Pock algorithm

$$\left\{\begin{array}{lcl} y_{k+1} & = & \mathrm{Prox}_{\sigma f^*}(y_k + \sigma K \bar{x}_k) \\ x_{k+1} & = & \mathrm{Prox}_{\tau g}(x_k - \tau K^* y_{k+1}) \\ \bar{x}_{k+1} & = & x_{k+1} + \beta(x_{k+1} - x_k) \end{array}\right. \tag{3.56}$$

For convex f and g, it is shown in (Chambolle and Pock, 2011, Theorem 1) that, for $\tau\sigma\,||K^*K|| < 1$, the iterates (x_k, y_k) given by the algorithm (3.56) converge towards (x^*, y^*) solution of the primal-dual problem (2.69).

The Primal-Dual algorithm is generalized in Chambolle and Pock (2016) with Bregman proximity functions as,

$$\left\{\begin{array}{ll} y_{k+1} & = \arg\min_y \frac{1}{\sigma} D_{h^Y}(y, y_k) + f^*(y) - \langle K\bar{x}_k, y\rangle \\ x_{k+1} & = \arg\min_x \frac{1}{\tau} D_{h^X}(x, x_k) + g(x) + \langle x, K^* y_{k+1}\rangle \\ \bar{x}_{k+1} & = x_{k+1} + \beta(x_{k+1} - x_k) \end{array}\right. \tag{3.57}$$

where D_h here denotes the Bregman divergence associated to a smooth and convex potential h:

$$D_h(x, y) = h(x) - h(y) - \langle \nabla h(y), x - y\rangle. \tag{3.58}$$

(3.57) falls back to (3.56) for $h^X(x) = h^Y(x) = \frac{1}{2}||x||^2$. For convex f and g, provided that the potentials h^X and h^Y are 1-convex with respect to the norm $||.||^2$ (*i.e.* $D_h(x, y) \geq \frac{1}{2}||x - y||^2$), convergence of (3.57) towards a saddle point of (2.68) is ensured as long as $\tau\sigma\,||K^*K|| < 1$ (Chambolle and Pock, 2016, Remark 3).

Nonconvex case

For nonconvex f, we consider the full primal version of the algorithm presented in (2.71)

$$\begin{cases} z_{k+1} & \in \text{Prox}_{\frac{1}{\sigma}f}(\frac{1}{\sigma}y_k + K\bar{x}_k) \\ y_{k+1} & = y_k + \sigma(K\bar{x}_k - z_{k+1}) \\ x_{k+1} & \in \text{Prox}_{\tau g}(x_k - \tau K^* y_{k+1}) \\ \bar{x}_{k+1} & = x_{k+1} + \beta(x_{k+1} - x_k). \end{cases} \tag{3.59}$$

The authors of Möllenhoff et al. (2015) study the convergence of this algorithm for weakly convex f and strongly convex g, also assuming that the strong convexity of g dominates the weak convexity of f in order to ensure that the overall objective $F = f(K.) + g$ is convex. This is too restrictive for our study, as neither the data-fidelity nor the regularization functions is strongly convex in the general case. More relevant for us, the authors of Sun et al. (2018) study the convergence of the algorithm (3.59) with $\beta = 0$, for nonconvex f and g functions, and prove a result, reminiscent of the previous convergence result of nonconvex DRS Theorem 10.

We define $w_k = (z_k, y_k, x_k, x_{k-1})$, $d_k = (x_k, y_k)$ and the Liapunov function

$$F^{PD}(w_k) = g(x_k) + f(z_k) + \langle Kx_k - z_k, y_k \rangle + \sigma \left|\left|z_k - Kx_{k-1}\right|\right|^2 + \sigma \left|\left|K\right|\right|^2 \left|\left|x_{k-1} - x_k\right|\right|^2 \tag{3.60}$$

Theorem 11 (Sun et al. (2018))**.** *Assume that f and g are proper, lsc. Let (z_k, y_k, x_k) produced by Algorithm (3.59) with $\beta = 0$, $2\tau\sigma \left|\left|K^2\right|\right| \leq 1$. Then,*

(i) $(F^{PD}(w_k))$ is non-increasing, with the sufficient decrease condition

$$F^{PD}(w_k) - F^{PD}(w_{k+1}) \geq \min\left(\frac{1}{2\tau} - \sigma \left|\left|K\right|\right|^2, \frac{1}{2\sigma}\right) \left|\left|d_k - d_{k+1}\right|\right|^2 \tag{3.61}$$

(ii) If (w_k) is bounded, $||d_{k+1} - d_k|| \to 0$ with finite length and for any cluster point (y^*, x^*) of (d_k), x^* is a critical point of $F : x \to f(Kx) + g(x)$.

(iii) If the sequence (w_k) is bounded and F^{PD} is KŁ, (d_k) converges towards (x^*, y^*) where x^* is a stationary point of $x \to f(Kx) + g(x)$.

Compared to the convergence of DRS Theorem 10, at this point, we do not need f or g to be smooth and bounded from below. However, the second and third points of the theorem require the boundedness of the iterates. The reason is that, with this choice of Liapunov function F^{PD}, they cannot show that F^{PD} is bounded from below when f and g are. Moreover, a sufficient condition for the boundedness of the iterates (w_k) is the coercivity of the decreasing Liapunov function F^{PD}. However, contrary to DRS, the coercivity of F is not enough (see (Sun et al., 2018, Lemma 3.10)) to guarantee the coercivity of F^{PD}. We now prove that this gap can be addressed by assuming, similar to Theorem 10 for DRS convergence, f differentiable with L_f-Lipschitz gradient. This is a new result previously not seen in the literature.

Theorem 12. *Assume f and g proper, lsc, lower-bounded. Suppose that f is L_f-smooth. Let (z_k, y_k, x_k) produced by Algorithm (3.59) with $\beta = 0$, $2\tau\sigma \left|\left|K\right|\right|^2 \leq 1$ and $\sigma \geq L_f$. Then, if F is coercive, the sequence (w_k) remains bounded.*

Proof. Departing from

$$F^{PD}(w_k) = g(x_k) + f(z_k) + \langle Kx_k - z_k, y_k \rangle + \sigma \left|\left|z_k - Kx_{k-1}\right|\right|^2 + \sigma \left|\left|K\right|\right|^2 \left|\left|x_{k-1} - x_k\right|\right|^2, \tag{3.62}$$

and using that

$$\begin{aligned} \left|\left|z_k - Kx_k\right|\right|^2 &= \left|\left|z_k - Kx_{k-1} + Kx_{k-1} - Kx_k\right|\right|^2 && (3.63)\\ &\leq 2\left|\left|z_k - Kx_{k-1}\right|\right|^2 + 2\left|\left|Kx_{k-1} - Kx_k\right|\right|^2 && (3.64)\\ &\leq 2\left|\left|z_k - Kx_{k-1}\right|\right|^2 + 2\left|\left|K\right|\right|^2 \left|\left|x_{k-1} - x_k\right|\right|^2 && (3.65) \end{aligned}$$

we get

$$F^{PD}(w_k) \geq g(x_k) + f(z_k) + \langle Kx_k - z_k, y_k \rangle + \frac{\sigma}{2}\left|\left|z_k - Kx_k\right|\right|^2 \tag{3.66}$$

that is to say

$$F^{PD}(w_k) \geq L_\sigma(x_k, y_k, z_k) \tag{3.67}$$

where $L_\sigma(x, y, z) := g(x) + f(z) + \langle Kx - z, y \rangle + \frac{\sigma}{2}\left|\left|z - Kx\right|\right|$ is the augmented Lagrangian, with dual variable y for the minimization problem

$$\begin{cases} f(z) + g(x) \\ Kx = z. \end{cases} \tag{3.68}$$

The optimality condition for the update of z_{k+1} gives ($\bar{x}_k = x_k$ since $\beta = 0$)

$$Kx_k + \frac{1}{\sigma} y_k - z_{k+1} \in \frac{1}{\sigma} \partial f(z_{k+1}). \tag{3.69}$$

Using the update $y_{k+1} = y_k + \sigma(Kx_k - z_{k+1})$ we get

$$y_{k+1} \in \partial f(z_{k+1}). \tag{3.70}$$

Since f is differentiable, $\forall k > 0$, $y_k = \nabla f(z_k)$ and

$$F^{PD}(w_k) \geq g(x_k) + f(z_k) + \langle Kx_k - z_k, \nabla f(z_k) \rangle + \frac{\sigma}{2}\left|\left|z_k - Kx_k\right|\right|^2 \tag{3.71}$$

Since we also suppose that f has L_f-Lipschitz gradient, we get by the descent lemma (Proposition 15), for $\sigma > L_f$,

$$F^{PD}(w_k) \geq g(x_k) + f(Kx_k) = F(x_k) \tag{3.72}$$

First we get that the sequence $(F^{PD}(w_k))$ is bounded from below and thus converges. Also, since $F(x_k) \leq F^{PD}(w_k) \leq F^{PD}(w_0)$, if F is coercive, we get that (x^k) remains bounded. From the second update of (3.59), $\left|\left|z^{k+1} - Kx^k\right|\right| = \frac{1}{\mu}\left|\left|y^{k+1} - y^k\right|\right|$ which tends to 0 by (ii) and thus (z_k) is also bounded. Finally, by relation $y^k = \nabla f(z^k)$, as f has Lipschitz gradient, (y_k) is also bounded. All in all, $w_k = (z_k, y_k, x_k, x_{k-1})$ remains bounded. □

3.3 Existing results on plug-and-play convergence

Designing convergence proofs for the PnP and RED algorithms presented in Section 2.4.2 is an active research topic. The goal is to derive sufficient and minimal conditions on the denoiser such that the algorithms converge. We separate the existing convergence studies into two main classes: *fixed-point convergence* and *convergence via minimization*. The former views plug-and-play algorithms as fixed-point operators and analyze their convergence using the fixed-point convergence theory developed in Section 3.1.3. The latter considers that the algorithms minimize a global functional and studies the convergence towards a minimizer (for a convex objective) or a stationary point (for a nonconvex objective).

3.3.1 Fixed-Point convergence

In the following, we view PnP and RED algorithms as fixed-point iterations

$$x_{k+1} = T(x_k). \tag{3.73}$$

For instance, the RED algorithms (2.102) and (2.103) give to the following fixed-point operators

$$\begin{cases} T_{RED-PGD} &= \mathrm{Prox}_{\lambda\tau f} \circ (\tau D_\sigma + (1-\tau)\,\mathrm{Id}) \\ T_{RED-GD} &= (\tau D_\sigma + (1-\tau)\,\mathrm{Id}) - \tau\lambda\nabla f \end{cases} \tag{3.74}$$

while for the PnP algorithms (2.109), (2.110) and (2.111)

$$\begin{cases} T_{PnP-PGD} &= D_\sigma \circ (\mathrm{Id} - \tau\lambda\nabla f) \\ T_{PnP-HQS} &= D_\sigma \circ \mathrm{Prox}_{\lambda\tau f} \\ T_{PnP-DRS} &= \beta(2D_\sigma - \mathrm{Id}) \circ \mathrm{Rprox}_{\tau\lambda f} + (1-\beta)\,\mathrm{Id} \end{cases} \tag{3.75}$$

In Section 3.1.3, we gave sufficient conditions on T (e.g. T averaged) to ensure the convergence of such iterations towards a fixed-point of T. Assuming a **convex data-fidelity term** f, in the context of plug-and-play, we look for sufficient conditions on the denoiser such that these assumptions are met.

Convergence theory with averaged denoisers

A first common assumption is to take the denoiser θ-averaged for $\theta \in (0,1)$ (see Definition 9). From the fixed-point convergence theory developed in Section 3.1.3, we derive convergence results for the PnP algorithms in Theorem 13 and RED algorithms in Theorem 14. Some of these results or their extensions can be found in the literature (Sun et al., 2019b,a, 2021; Pesquet et al., 2021; Ryu et al., 2019) but others, in particular for RED convergence, have not been given in the literature so far. Therefore, for the sake of completeness, we give the full proofs of both theorems.

Theorem 13 (Fixed-Point convergence of PnP algorithms)**.** *Assume D_σ θ-averaged for $\theta \in (0,1)$ and f proper, lsc, convex. If the PnP fixed-points operators* (3.75) *have fixed-points, then*

(i) If f is differentiable with L_f-Lipschitz gradient, for $0 < \tau\lambda\nabla f < 2$, PnP-PGD converges towards a fixed-point of $T_{PnP-PGD}$.

(ii) For $\tau > 0$, PnP-HQS converges towards a fixed-point of $T_{PnP-HQS}$.

(iii) If D_σ is firmly nonexpansive (i.e. $\theta = \frac{1}{2}$), for $\tau > 0$ and $\beta \in (0,1)$, PnP-DRS converges towards a fixed-point of $T_{PnP-DRS}$.

And all the fixed-point sequences verify $\sum ||x_{k+1} - x_k||^2 < +\infty$.

Proof. Points (i) and (ii) consist in showing that the PnP operator T is averaged using the fact that the composition of averaged operators is averaged (Proposition 17(i)). As f is convex, Proposition 18 indicates that $\text{Prox}_{\lambda\tau f}$ is firmly nonexpansive and that $\text{Id} - \lambda\tau f$ is averaged for $\tau\lambda L < 2$. As D_σ is also averaged, by composition, we respectively get that $T_{PnP-HQS}$ and $T_{PnP-PGD}$ (for $\tau\lambda L < 2$) are averaged. Eventually, we conclude on the convergence result applying Corollary 3.

For point (iii), we show that the operator $T = (2D_\sigma - \text{Id}) \circ \text{Rprox}_{\tau\lambda f}$ is nonexpansive, and then we can conclude on the convergence applying Theorem 5. $\text{Prox}_{\lambda\tau f}$ and D_σ are both firmly nonexpansive, thus, by definition, $\text{Rprox}_{\lambda\tau f} = 2\,\text{Prox}_{\lambda\tau f} - \text{Id}$ and $2D_\sigma - \text{Id}$ are nonexpansive, and T is nonexpansive. □

Theorem 14 (Fixed-Point convergence of RED algorithms)**.** *Assume D_σ θ-averaged for $\theta \in (0,1)$ and f proper, lsc, convex. If the RED fixed-points operators* (3.74) *have fixed-points, then*

(i) For $0 < \tau\theta < 1$, RED-PGD converges towards a fixed-point of $T_{RED-PGD}$.

(ii) If f is differentiable with L_f-Lipschitz gradient, for $0 < \tau\left(\frac{\lambda L_f}{2} + \theta\right) < 1$, RED-GD converges towards a fixed-point of T_{RED-GD}.

And all the fixed-point sequences verify $\sum ||x_{k+1} - x_k||^2 < +\infty$.

Remark 14. *The proof below remaining true for the case $\theta = 1$, the convergence results of RED hold when D_σ is only nonexpansive but not necessarily averaged (at the expense of stronger conditions on the stepsize).*

Proof. For (i), we use Proposition 16 to get that $\tau D_\sigma + (1-\tau)\,\text{Id}$ is $\tau\theta$-averaged, and we conclude using again that the composition of averaged operators is averaged (Proposition 17(i)) along with Corollary 3 for convergence. For (ii), we re-write T_{RED-GD} as

$$T_{RED-GD} = \tau\left(D_\sigma - \lambda\nabla f\right) + (1-\tau)\,\text{Id} \tag{3.76}$$

We denote $\hat{T} = D_\sigma - \lambda\nabla f$ and $\hat{\theta} = \frac{\lambda L_f}{2} + \theta$. If we show that $\hat{T} = \hat{\theta}R + (1-\hat{\theta})\,\text{Id}$ for some nonexpansive operator R ($\hat{\theta}$ is possibly bigger than 1), then using Proposition 16, we get that $T_{RED-GD} = \tau\hat{T} + (1-\tau)\,\text{Id}$ is $\tau\hat{\theta}$-averaged, and we can conclude on the convergence using Corollary 3. We have

$$\hat{T} = D_\sigma - \lambda\nabla f = \frac{1}{2}\left((2D_\sigma - \text{Id}) + (\text{Id} - 2\lambda\nabla f)\right). \tag{3.77}$$

Using that D_σ is θ-averaged and $\text{Id} - \frac{1}{L}\nabla f$ is $\frac{1}{2}$-averaged (Proposition 18), we write, with R_1 and R_2 two nonexpansive operators,

$$D_\sigma = \theta R_1 + (1-\theta)\,\text{Id} \tag{3.78}$$

$$\text{Id} - \frac{1}{L}\nabla f = \frac{1}{2}R_2 + \frac{1}{2}\,\text{Id} \tag{3.79}$$

giving respectively

$$2D_\sigma - \mathrm{Id} = 2\theta R_1 + (1-2\theta)\,\mathrm{Id} \tag{3.80}$$

$$\mathrm{Id} - 2\lambda\nabla f = 2\lambda L(\mathrm{Id} - \frac{1}{L}\nabla f) + (1-2\lambda L)\,\mathrm{Id} = \lambda L R_2 + (1-\lambda L)\,\mathrm{Id}\,. \tag{3.81}$$

Thus we can write $\hat{T}$ as

$$\hat{T} = \frac{1}{2}\left(2\theta R_1 + (1-2\theta)\,\mathrm{Id}\right) + \frac{1}{2}\left(\lambda L R_2 + (1-\lambda L)\,\mathrm{Id}\right) \tag{3.82}$$

$$= \theta R_1 + \frac{\lambda L}{2} R_2 + \left(1 - (\theta + \frac{\lambda L}{2})\right)\mathrm{Id} \tag{3.83}$$

$$= \hat{\theta} R + (1-\hat{\theta})\,\mathrm{Id} \tag{3.84}$$

with

$$R := \frac{\theta}{\theta + \frac{\lambda L}{2}} R_1 + \frac{\frac{\lambda L}{2}}{\theta + \frac{\lambda L}{2}} R_2 \tag{3.85}$$

which is nonexpansive as a weighted average of nonexpansive operators. □

For both previous convergence theorems, the fact that the sequences verify

$$\sum ||x_{k+1} - x_k||^2 < +\infty \tag{3.86}$$

implies that they converge with rate

$$\min_{0\leq k\leq K} ||x_{k+1} - x_k|| = \mathcal{O}(\frac{1}{\sqrt{K}}). \tag{3.87}$$

As we will see now, it is as hard to train averaged deep denoiser as to train nonexpansive denoisers. However, for a practitioner who does not want to train any denoiser but has access to an *off-the-shelf nonexpansive denoiser, but not averaged*, it may still be possible to get convergent algorithms. First, for RED algorithms, as noticed in Remark 14, the convergence result holds for a nonexpansive denoiser. For the case of PnP-PGD and PnP-HQS algorithm, it is easy to show that the PnP fixed-point operators T_{PnP} with nonexpansive denoisers are not averaged but nonexpansive. The solution to ensure convergence is thus to modify the algorithms $x_{k+1} = T_{PnP}(x_k)$ with Krasnonel'skii-Mann iterations

$$x_{k+1} = \nu_k T_{PnP}(x_k) + (1-\nu_k)x_k \tag{3.88}$$

with a sequence $\nu_k \in [0,1]$ such that $\sum \nu_k(1-\nu_k) = \infty$. Theorem 5 then gives the convergence of (3.88) towards a fixed-point of T_{PnP}. Note that this trick does not extend to PnP-DRS which already writes as Krasnonel'skii-Mann iterations.

Training averaged and nonexpansive deep denoisers

A generic deep denoiser trained without specific constraint is not averaged nor nonexpansive in practice. In this section, we review different methods in the literature that have been proposed to train nonexpansive denoisers, or nonexpansive neural networks in general. Controlling the Lipschitz constant of deep neural networks is a topic of interest in machine learning, as it has been observed that it helps generalization and adversarial robustness. However, training Lipschitz-constrained networks is difficult, as the computation of the

Lipschitz constant of multi-layer models is NP-hard. Note that if we can train a network to be nonexpansive, we can train a denoiser to be θ-averaged simply by parameterizing

$$D = \theta R + (1 - \theta)\,\mathrm{Id} \tag{3.89}$$

with R the constrained nonexpansive network.

Hard constraints consist in hard coding the Lipschitz condition by modifying the architecture of the network. See Neumayer et al. (2023) for a thorough discussion on existing methods. The most common strategy is *Spectral Normalization Miyato et al. (2018)* which normalizes, during training, each layer of a feed-forward neural network by its spectral norm. More precisely, consider a feed-forward network with $L + 1$ layers

$$R = a_L \circ W^L \circ a_{L-1} \circ W^{L-1} \circ .... \circ a_0 \circ W^0 x \tag{3.90}$$

where the bias terms are omitted for simplicity and (a_k) are 1-Lipschitz activation functions (e.g. ReLU). Spectral normalization normalizes the spectral norm of each weight matrix

$$W_k \longleftarrow \frac{W_k}{||W_k||_S} \tag{3.91}$$

where the spectral norm $||W_k||_S = \sup_{||x||\leq 1} ||W_k x||$ corresponds to the Lipschitz constant of the linear operator $x \to W_k x$. The Lipschitz constant of the full network being upper-bounded by $\prod_{k=0}^{L} \mathrm{Lip}(a_k)\,\mathrm{Lip}(W_k) \leq 1$, we get that the full network is nonexpansive. In practice, for large networks with numerous layers, this upper-bound is quite pessimistic. Spectral normalization has the tendency to over-constrain the overall Lipschitz constant of R and severely reduces the expressivity of the model, or to create vanishing gradients (Anil et al., 2019). Neumayer et al. (2023) mitigates this effect by replacing the suboptimal ReLU activation by a more expressive learnable 1-Lipschitz linear spline.

Instead of constraining the spectral norm of W_k, it has also been proposed (Cisse et al., 2017; Huang et al., 2018; Hasannasab et al., 2020; Hertrich et al., 2021) to project each weight matrix W_k (or W_k^T) on the *Stiefel manifold i.e.* to impose $W_k^T W_k = \mathrm{Id}$ (or $W_k {W_k}^T = \mathrm{Id}$).

While all these methods have been applied to denoisers parameterized with feed-forward architectures for PnP (Ryu et al., 2019; Hertrich et al., 2021; Bohra et al., 2021), they are not applicable to state-of-the-art UNet architectures because of the presence of skip-connections. Moreover, for spectral normalization, the spectral norm is approximated with a few power iterations Golub and Van der Vorst (2000) on the current training batch of images. Therefore, the nonexpansivity is not rigorously hard constrained and may fail when evaluated on singular input images which do not resemble training images.

Soft constraint Instead of imposing a hard constraint on the architecture, another strategy consists in regularizing the training loss of the denoiser with an additional term. This is common practice in GAN training to impose smoothness on the discriminator with a gradient penalty (Gulrajani et al., 2017). In our case, we want to impose a precise upper bound on the Lipschitz constant of the denoising network D. A differentiable map $R : \mathbb{R}^n \to \mathbb{R}^n$ is nonexpansive *if and only if* the spectral norm of its Jacobian is everywhere smaller than 1 i.e.

$$\forall x \in \mathbb{R}^n, ||J_R(x)||_S \leq 1. \tag{3.92}$$

Pesquet et al. (2021) then propose to regularize the standard training loss $l(x, D(y))$ via

$$\mathbb{E}_{x,y\sim p(y|x)}\Big[l(x, D(y)) + \mu \max(||J_D(\tilde{x})||_S^2, 1 - \epsilon)\Big] \tag{3.93}$$

where $\tilde{x}$ randomly interpolates on the segment $[x, D(y)]$. The strength of the penalty is controlled by the parameters $\mu > 0$ and $\epsilon \in (0, 1)$. The spectral norm of the Jacobian is computed via power iterations. This algorithm requires only Jacobian-vector products (and not the full Jacobian) which can be efficiently computed via automatic differentiation. Note that in order to constrain the denoiser to be θ-averaged, one should apply the spectral regularization on $R = \frac{1}{\theta}D + (1 - \frac{1}{\theta})\,\mathrm{Id}$. The main disadvantage of this method is again that the nonexpansivity is not guaranteed and is likely not to hold for specific images not seen during training.

Nonexpansive denoisers are suboptimal It has been observed that constraining a denoising neural network to be nonexpansive can severely degrade its denoising performance compared to the same unconstrained network, see for example (Hertrich et al., 2021, Table 1), (Bohra et al., 2021, Figure 1), (Nair and Chaudhury, 2022, Table 3). Nonexpansivity appears to be a non-realistic assumption for a Gaussian denoiser. A semi-theoretical reason is given by Denoising Score Matching (DSM) (Equation (2.80)). DRS shows that training a neural network to denoise (with the L^2 loss) comes back to approximating the optimal MMSE denoiser. However, *the MMSE denoiser is not nonexpansive if p_σ is not log-concave.* To see this, we recall the Tweedie formula

$$D_\sigma^{MMSE}(x) = x + \sigma^2 \nabla \log p_\sigma(x) = \nabla \left(\frac{1}{2} ||x||^2 + \sigma^2 \log p_\sigma(x) \right) \tag{3.94}$$

and from Corollary 2,

$$D_\sigma^{MMSE} \text{ is nonexpansive} \Leftrightarrow x \to \frac{1}{2} ||x||^2 + \sigma^2 \log p_\sigma(x) \text{ 1-smooth} \Longrightarrow -\log p_\sigma \text{ convex..} \tag{3.95}$$

As p_σ is not log-concave (at least for reasonable small σ values), it is unrealistic to assume that a denoiser trained with L^2 loss is nonexpansive.

Strongly convex data-fidelity term

The absence of nonexpansivity of the denoising operator D_σ can be compensated when the data fidelity term f is strongly convex. Indeed, in that case, as presented in Proposition 19, the gradient descent and proximal maps are contractive. Strongly convex data-fidelity terms exclude numerous image restoration tasks, in particular all data-fidelity of the form $f(x) = \frac{1}{2} ||Ax - y||^2$ with singular A, such as inpainting, deblurring, or super-resolution with Gaussian noise assumption. However, it does encompass a few interesting inverse problems, such as Poisson denoising or single photon imaging. With strongly-convex f, Ryu et al. (2019) show fixed-point convergence of PnP-PGD and PnP-DRS without requiring nonexpansivity of D_σ but under some constraint on the Lipschitz constant of $\mathrm{Id} - D_\sigma$.

Theorem 15 (PGD convergence Ryu and Boyd (2016), Theorem 1)**.** *For f L-smooth and γ-strongly convex, assume $\mathrm{Id} - D_\sigma$ ϵ-Lipschitz with $\epsilon < \frac{2\gamma}{L-\gamma}$. Then for a stepsize $\frac{\epsilon}{\gamma(1+\epsilon)} < \tau < \frac{2}{L} - \frac{\epsilon}{L(1+\epsilon)}$, the operator $T_{PnP-PGD}$ is contractive and the PnP-PGD fixed-point iterations converge geometrically towards its unique fixed point.*

The proof simply combines the Lipschitz constant of the gradient descent map for a strongly convex function (Proposition 19) with the fact that D_σ is $(1 + \epsilon)$-Lipschitz.

Note that one could instead assume that D_σ is $\sqrt{1+\hat{\epsilon}}$-Lipschitz, which is called in the literature Russell Luke et al. (2018) *almost-nonexpansive with violation* $\hat{\epsilon}$.

Theorem 16 (DRS convergence Ryu and Boyd (2016), Theorem 2)**.** *For f differentiable and γ-strongly convex, assume* $\mathrm{Id} - D_\sigma$ *ϵ-Lipschitz with $\epsilon < 1$. Then for a stepsize $\tau > \frac{\epsilon}{\gamma(2\epsilon+1)(1-\epsilon)}$, the operator $T_{PnP-DRS}$ with $\beta = \frac{1}{2}$ is contractive and the PnP-DRS fixed-point iterations converge geometrically towards its unique fixed point.*

The proof of this result uses the theory of averaged operators. The result is however quite limiting, as it requires ϵ the Lipschitz constant of $\mathrm{Id} - D_\sigma$ to be quite small for the constraint on the stepsize to be acceptable.

3.3.2 Convergence via minimization

The previous section consisted in proving convergence of the PnP and RED algorithms towards the fixed-points of a certain operator. Besides the required nonexpansivity of the denoiser, these fixed-point convergence results have different limitations. First, it is necessary to assume existence of such fixed-points. When the operator is not contractive, this is a condition which is difficult to verify in practice. Second, contrary to fixed-point of real optimization algorithms, fixed-point of PnP and RED algorithms do not correspond to critical points of a given functional. Therefore, the interpretation of the output of the algorithm is limited.

Instead, in this manuscript, our primary approach revolves around transforming the PnP and RED algorithms into genuine optimization algorithms designed to address the critical points of a global objective function. This transformation will be accomplished through the utilization of specific parameterizations of the plugged denoisers. Prior to our work, several studies have pursued a related strategy for RED or PnP convergence.

RED convex convergence RED original paper Romano et al. (2017) introduced RED algorithms as minimizers of an explicit objective. Given an off-the-shelf denoiser D_σ, they construct an explicit regularizer $g_\sigma : \mathbb{R}^n \to \mathbb{R}$ given by

$$g_\sigma(x) = \frac{1}{2}\langle x, x - D_\sigma(x)\rangle \tag{3.96}$$

They propose sufficient conditions on the denoiser for the following identity to be verified

$$\nabla g_\sigma = \mathrm{Id} - D_\sigma. \tag{3.97}$$

This result was completed in Reehorst and Schniter (2018) which states the following conditions on D_σ for the identity (3.97) to hold:

(i) Local homogeneity : There is $\delta > 0$, $\forall x \in \mathbb{R}^n, D_\sigma\left((1+\epsilon)x\right) = (1+\epsilon)D_\sigma(x)$ for $||\epsilon|| \leq \delta$.

(ii) Jacobian Symmetry (JS) : $\forall x \in \mathbb{R}^n, J_{D_\sigma}(x) = J_{D_\sigma}(x)^T$.

The Jacobian symmetry is a necessary and sufficient condition for a differential mapping to be a *conservative vector field*, i.e. to write $D_\sigma = \nabla h_\sigma$ for some differential potential $h_\sigma : \mathbb{R}^n \to \mathbb{R}$. Romano et al. (2017) added the following assumption

(iii) Strong passivity : $\forall x \in \mathbb{R}^n, ||J_{D_\sigma}(x)||_S \leq 1$ where $||.||_S$ denotes the spectral norm,

which implies nonexpansivity of $D_\sigma = \mathrm{Id} - \nabla g_\sigma$. From Corollary 2, this implies that g_σ is convex.

As presented in Section 2.4.2, RED algorithms were originally built using the identity (3.97) to replace the gradient of $-\log p_\sigma$. Therefore, if a denoiser D_σ verifies (3.97), then the RED algorithm (e.g. RED-PGD (2.102)) takes its original form of a first order optimization algorithm (e.g. a real PGD (2.98)) for minimizing $\lambda f + g_\sigma$. g_σ being convex and assuming f also convex, we get convergence of RED-GD and RED-PGD algorithms using the convex convergence properties of GD and PGD derived Section 3.2. We summarize this analysis in the following Proposition.

Proposition 21 (Convergence of RED algorithms via convex minimization)**.** *Assume D_σ verifying local homogeneity, Jacobian symmetry and strong passivity. Then D_σ verifies $D_\sigma = \mathrm{Id} - \nabla g_\sigma$ for $g_\sigma : x \to \frac{1}{2}\langle x, x - D_\sigma(x)\rangle$. Assume that f proper, lsc, convex. Then,*

(i) For $0 < \tau < \frac{2}{\mathrm{Lip}(\mathrm{Id} - D_\sigma)}$, RED-PGD (2.102) converges towards a global minimum of $\lambda f + g_\sigma$.

(ii) If f is differentiable with L_f-Lipschitz gradient, for $0 < \tau < \frac{2}{\lambda L_f + \mathrm{Lip}(\mathrm{Id} - D_\sigma)}$, RED-GD (2.103) converges towards a global minimum of $\lambda f + g_\sigma$.

The Jacobian Symmetry (JS) property is verified when D_σ is a conservative vector field, i.e. $D_\sigma = \nabla h_\sigma$ for some differential potential $h_\sigma : \mathbb{R}^n \to \mathbb{R}$. It is thus verified by both the theoretical MMSE and MAP denoisers (see Section 2.4.1). However, for practical generic denoisers, such as BM3D or deep denoiser, Reehorst and Schniter (2018) provides evidences that the JS property is not true. It is moreover difficult to constrain a deep denoiser to verify this hypothesis. Additionally, we explained in Section 3.3.1 that nonexpansivity is a suboptimal assumption for a denoiser and that enforcing this condition inevitably worsens restoration performance.

Instead, in Chapter 4, we propose to train a deep denoiser which, by parametrization, inherently satisfies relation (3.97) for some **nonconvex deep potential** g_σ. Although, the proposed denoiser will not be nonexpansive, we will show that RED algorithms still converge towards the critical points of an explicit functional while achieving performant restoration.

PnP convex convergence The authors of Sreehari et al. (2016) make use of Moreau's characterization of the proximity operator (Theorem 2) to give sufficient conditions for the denoiser D_σ to be an explicit proximal map. We recall that according to Theorem 2, if D_σ is nonexpansive and $D_\sigma = \nabla h_\sigma$ for some convex potential h_σ, then there exists $\phi_\sigma : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ convex such that $D_\sigma = \mathrm{Prox}_{\phi_\sigma}$.

As in the previous section on RED convergence, the authors assume Jacobian Symmetry (JS) but also double stochasticity of the Jacobian of the denoiser. Indeed, from JS, $D_\sigma = \nabla h_\sigma$ for some potential $h_\sigma : \mathbb{R}^n \to \mathbb{R}$. Moreover, double stochasticity implies that the eigenvalues of D_σ belong to $(0, 1]$. From this fact, we get that h_σ is convex (as its Hessian is positive) and that D_σ is nonexpansive. All in all, under these assumptions, there is $\phi_\sigma : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ convex, such that

$$D_\sigma = \mathrm{Prox}_{\phi_\sigma} \tag{3.98}$$

As presented in Section 2.4.2, PnP algorithm were originally built using precisely $D_\sigma = \mathrm{Prox}_{-\tau\sigma^2 \log p}$ in proximal optimization algorithms. Then, with a plugged denoiser verifying

relation (3.98), a PnP algorithm (with stepsize $\tau = 1$) becomes again a real optimization algorithm for minimizing $\lambda f + \phi_\sigma$. ϕ_σ being convex, assuming f convex, we get convergence of PnP algorithms using the convex convergence properties of PGD, HQS or DRS derived Section 3.2. We summarize this analysis in the following Proposition.

Proposition 22 (Convergence of PnP algorithms via convex minimization)**.** *Assume D_σ verifying Jacobian symmetry and double stochasticity. Then there is $\phi_\sigma : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ convex, such that $D_\sigma = \mathrm{Prox}_{\phi_\sigma}$. Assume f proper, lsc, convex. Then,*

(i) If f is differentiable with L_f-Lipschitz gradient, for $0 < \lambda \nabla f < 2$, PnP-PGD (2.109) *with $\tau = 1$ converges towards a global minimum of $\lambda f + \phi_\sigma$.*

(ii) PnP-DRS (2.111) *with $\tau = 1$ converges towards a global minimum of $\lambda f + \phi_\sigma$.*

(iii) The PnP version of the Primal-Dual algorithm (2.71) *(PnP-PD) with $\tau = 1$ and $\sigma ||K||^2 < 1$ converges towards a global minimum of $\lambda f(K\cdot) + \phi_\sigma$.*

The stepsize $\tau = 1$ is due to the fact that the denoiser writes as $D_\sigma = \mathrm{Prox}_{\phi_\sigma}$ and not $D_\sigma = \mathrm{Prox}_{\tau\phi_\sigma}$ as it is normally the case in PGD and DRS algorithms.

However, these convergence results have the same limitations as the ones presented above for RED: Jacobian Symmetry and nonexpansivity are not a satisfactory assumption for a generic denoiser. In Chapter 5, we train a deep denoiser which inherently verifies relation (3.98) for some **nonconvex potential** ϕ_σ. We then exploit the convergence theory of proximal algorithms in the nonconvex setting to obtain convergence results for PnP algorithms with this denoiser.

Chapter 4

Gradient-Step Denoiser for Convergent RED algorihms

Contents

In this chapter, we propose novel theoretical convergence guarantees for the Regularization by Denoising (RED) algorithms introduced in Section 2.4.2. As detailed in the "RED convergence" paragraph in Section 3.3.2, convergence of RED was proven by Romano et al. (2017) under nonexpansivity and Jacobian symmetry of the plugged denoiser. We explained that these assumptions are suboptimal or non-realistic for existing deep denoisers. Instead of taking such general assumptions, in this work, we introduce a new denoiser that satisfies, by construction, sufficient conditions for RED convergence.

To be more precise, we incorporate a denoiser that performs a deliberate gradient descent step on a nonconvex potential, which is parameterized by a deep neural network. This denoiser is inherently designed as a conservative vector field and inherently fulfills the requisite Jacobian Symmetry condition outlined in (Romano et al., 2017). This condition is enforced without compromising the denoising performance. The outcome is that the RED-PGD (2.102) and RED-GD (2.103) algorithms take the form of precise PGD and GD schemes. By leveraging the nonconvex convergence analysis of these algorithms, detailed in Section 3.2, we can demonstrate that our RED algorithms are assured to converge toward a stationary point of an explicit functional. These convergence guarantees avoid the need for unrealistic assumptions regarding the denoiser, such as its nonexpansivity. The convergence also doesn't hinge on the strong convexity of the data-fidelity term, thus

encompassing challenging image restoration tasks like deblurring, super-resolution, or inpainting. Furthermore, as a supplementary outcome from the Denoising Score Matching interpretation, our denoiser provides an explicit regularization term that approximates the true smoothed image prior.

In Section 4.1, we present the Gradient-Step (GS) denoiser. In Section 4.2, we delve into an analysis of the convergence for the derived RED algorithms utilizing the GS denoiser, thereby termed as GSRED algorithms. Lastly, in Section 4.3, we present the training and denoising capabilities of the GS denoiser, followed by an experimental validation of both the convergence and the performance of our proposed restoration algorithms.

Most of the results presented in this chapter were published in (Hurault et al., 2022a). The code implementing the proposed framework is available at `https://github.com/samuro95/GSPnP`.

4.1 Gradient-Step denoiser

4.1.1 Parameterization and training

In order to define a convergent scheme, we first set up a Gradient-Step (GS) denoising operator D_σ that takes the form of a gradient descent step

$$D_\sigma = \mathrm{Id} - \nabla g_\sigma, \tag{4.1}$$

with $g_\sigma : \mathbb{R}^n \to \mathbb{R}$ parameterized by a neural network. The gradient ∇g_σ can be computed exactly using automatic differentiation. Contrary to the original RED regularizer (Romano et al., 2017) detailed in Section 3.3.2, by construction, our denoiser exactly represents a conservative vector field.

The choice of the parameterization of g_σ is fundamental for the denoising performance. As already noticed by Salimans and Ho (2021), we experimentally found that directly modeling g_σ as a neural network (*e.g.* a standard network used for classification) leads to poor denoising performance. In order to keep the strength of state-of-the-art unconstrained denoisers, we rather use

$$g_\sigma(x) = \frac{1}{2}\left|\left|x - N_\sigma(x)\right|\right|^2, \tag{4.2}$$

$$\text{which leads to } D_\sigma(x) = x - \nabla g_\sigma(x) = N_\sigma(x) + J_{N_\sigma}(x)^T(x - N_\sigma(x)), \tag{4.3}$$

where $N_\sigma : \mathbb{R}^n \to \mathbb{R}^n$ is parameterized by a neural network and $J_{N_\sigma}(x)$ is the Jacobian of N_σ at point x. Thanks to our definition (4.3) for D_σ, **we can parameterize N_σ with any differentiable neural network architecture** $\mathbb{R}^n \to \mathbb{R}^n$ that has proven efficient for image denoising. With (4.3), our denoiser corresponds to applying the neural network N_σ with an additive correction that makes the denoiser a conservative field.

Although the representation power of the denoiser is limited by the particular form (4.3), we show in the experiment section that a such parameterization still yields state-of-the-art denoising results. We train the denoiser D_σ for denoising Gaussian noise of standard deviation σ by minimizing the MSE loss function

$$\mathcal{L}(D_\sigma) = \mathbb{E}_{X\sim p_X, \xi_\sigma \sim \mathcal{N}(0,\sigma^2 I)}\left[\left|\left|D_\sigma(x+\xi_\sigma) - x\right|\right|^2\right], \tag{4.4}$$

$$\text{or } \mathcal{L}(g_\sigma) = \mathbb{E}_{x\sim p_X, \xi_\sigma \sim \mathcal{N}(0,\sigma^2 I)|}\left[\left|\left|\nabla g_\sigma(x+\xi_\sigma) - \xi_\sigma\right|\right|^2\right], \tag{4.5}$$

when written in terms of g_σ using equation (4.1). From the *Denoising Score Matching* (Vincent, 2011) result, detailed in Section 2.4.1, we have

$$\mathcal{L}(D_\sigma) = \mathbb{E}_{y\sim p_\sigma}\left[\left|\left|D_\sigma(y) - (y + \sigma^2\nabla \log p_\sigma(Y))\right|\right|^2\right] + \text{const} \tag{4.6}$$

$$= \mathbb{E}_{y\sim p_\sigma}\left[\left|\left|\nabla g_\sigma(y) - \sigma^2\nabla(-\log p_\sigma)(Y))\right|\right|^2\right] + \text{const} \tag{4.7}$$

where $p_\sigma = p_X * \mathcal{N}(0, \sigma^2 I)$ is the marginal distribution of the noisy observation. Hence, training the potential g_σ by minimizing the denoising loss (4.4) is related to the approximation of the gradient of the log-smoothed image prior. Instead of directly approximating this gradient with a neural network like *score-based models* (see Section 2.1.1), we realize the approximation with another gradient field. Our potential g_σ is an *energy-based* model, parameterizing a probability density

$$\hat{p}_\sigma(x) = \frac{\exp(-g_\sigma(x))}{Z} \tag{4.8}$$

and trained to minimize the divergence (4.7) between $\hat{p}_\sigma$ and p_σ.

4.1.2 More details on the regularization potential g_σ.

We first underline that the main point of our method is to define the denoiser as $D_\sigma = \mathrm{Id} - \nabla g_\sigma$. The choice for g_σ is important for the denoising performance. With respect to the RED convergence properties, this is nevertheless a secondary issue, as our method would converge for other differentiable regularizers g_σ.

In practice, as detailed in Section 4.3, we parameterize g_σ as (4.2) with N_σ the DRUNet architecture (Zhang et al., 2021) (represented Figure 2.1) where the non-differentiable RELU activation function is replaced with the smooth ELU activation defined by

$$ELU(x) = \begin{cases} x & \text{if } x \geq 0 \\ e^x - 1 & \text{if } x < 0\ . \end{cases} \tag{4.9}$$

The proposed regularization $g_\sigma(x) = \frac{1}{2}||x - N_\sigma(x)||^2$ was previously mentioned in the RED original paper (Romano et al., 2017) (but explicitly left aside) and used in the DAEP paper (Bigdeli and Zwicker, 2017). The main difference between our regularizer and the one proposed in RED and DAEP is the following:

- RED and DAEP both consider a generic given pretrained denoiser $D_\sigma : \mathbb{R}^n \to \mathbb{R}^n$, which is then associated with the regularizer $g_\sigma(x) = \frac{1}{2}||x - D_\sigma(x)||^2$ and used as such in IR problems.
- In our method, we set $g_\sigma(x) = \frac{1}{2}||x - N_\sigma(x)||^2$ (with $N_\sigma : \mathbb{R}^n \to \mathbb{R}^n$ differentiable) and then we train the denoiser as $D_\sigma = \mathrm{Id} - \nabla g_\sigma$ with the loss function $||D_\sigma(x + \xi) - x||^2$ for clean images x and additive white Gaussian noise ξ.

With this new formulation, we are ensured that $D_\sigma = \mathrm{Id} - \nabla g_\sigma$ is inherently a conservative vector field, without further assumptions on N_σ. In contrast to the original RED paper, we aim at finding one setting of plug-and-play image restoration that allows for a convergence proof with sufficiently general hypotheses. For this purpose, we have to consider this very particular form of regularization.

Nonconvexity of g_σ With the above parameterization, the deep potential g_σ has no reason to be convex. If properly trained, it should be nonconvex. Indeed, with the Denoising Score Matching result (4.7), g_σ approximates $-\log p_\sigma$ which is likely to be highly nonconvex. We now show that g_σ has Lipschitz gradient by construction and is thus weakly convex (Corollary 2).

Lipschitz continuity of ∇g_σ We now give a result which ensures that a large class of neural networks trained with differentiable activation functions have Lipschitz gradients with respect to the input image.

Proposition 23. *Let $H = h_p \circ \ldots \circ h_1$ be a composition of differentiable functions $h_i : \mathbb{R}^{d_{i-1}} \to \mathbb{R}^{d_i}$. Let us assume that for any i the differential map h_i' is bounded and Lipschitz. Then H' is Lipschitz.*

Proof. Let us denote $H_i = h_i \circ \ldots \circ h_1$ (and by convention, $H_0 = \mathrm{Id}$). Let $\|h_i'\|_\infty$ be the best uniform bound on the operator norms $\|h_i'(x)\|, x \in \mathbb{R}^{d_{i-1}}$ (which is also the best Lipschitz constant of h_i). Let us also denote $\|h_i'\|_{\mathrm{Lip}}$ the Lipschitz constant of h_i'. The chain rule gives that for any x, $H'(x)$ can be expressed as a composition of linear maps

$$H'(x) = h_p'(H_{p-1}(x))h_{p-1}'(H_{p-2}(x)) \ldots h_1'(x). \tag{4.10}$$

Therefore, for any x, y,

$$H'(x) - H'(y) = \sum_{i=0}^{p-1} h_p'(H_{p-1}(x)) \ldots h_{i+1}'(H_i(x))h_i'(H_{i-1}(y)) \ldots h_1'(y) \tag{4.11}$$

$$- h_p'(H_{p-1}(x)) \ldots h_{i+1}'(H_i(y))h_i'(H_{i-1}(y)) \ldots h_1'(y). \tag{4.12}$$

We can thus bound the operator norms

$$\|H'(x) - H'(y)\| \leq \sum_{i=0}^{p-1} \Big(\|h_p'(H_{p-1}(x)) \ldots h_{i+2}'(H_{i+1}(x))\| \\ \|h_{i+1}'(H_i(x)) - h_{i+1}'(H_i(y))\| \|h_i'(H_{i-1}(y)) \ldots h_1'(y)\| \Big). \tag{4.13}$$

$$\tag{4.14}$$

and thus

$$\|H'(x) - H'(y)\| \leq \sum_{i=0}^{p-1} \Big(\prod_{j \neq i+1} \|h_j'\|_\infty \Big) \|h_{i+1}'\|_{\mathrm{Lip}} \|H_i'\|_\infty \|x - y\| \tag{4.15}$$

which concludes the proof because the chain-rule ensures that $\|H_i'\|_\infty \leq \|h_i'\|_\infty \ldots \|h_1'\|_\infty$. □

Proposition 23 applies for a neural network obtained as a composition of convolutional layers with ELU activation functions, that is, by composing functions of the form

$$h(x) = ELU(Ax + b). \tag{4.16}$$

It is easy to see that ELU defined in (4.9) is differentiable and that E' is 1-Lipschitz with $\|E'\|_\infty \leq 1$. Therefore,

$$h'(x) = A^T ELU'(Ax + b) \tag{4.17}$$

is also bounded and Lipschitz.

Let us also mention that this proposition encompasses the case of U-nets which, in addition to composing fully-connected layers, also integrate skip-connections. For example, taking a skip-connection on a composition $h_3 \circ h_2 \circ h_1$ amounts to defining

$$H(x) = h_3\big(\ h_2(h_1(x))\ ,\ h_1(x)\ \big). \tag{4.18}$$

This can be simply rewritten $H = h_3 \circ \tilde{h}_2 \circ h_1$ where

$$\tilde{h}_2(x) = \big(h_2(h_1(x)), h_1(x)\big). \tag{4.19}$$

It is then clear that $\tilde{h}_2$ has bounded Lipschitz differential as soon as h_1 and h_2 do.

Therefore, we can ensure the Lipschitz continuity of the Jacobian of N_σ and thus of the Jacobian of $T_\sigma(x) = x - N_\sigma(x)$. However, $x \to \frac{1}{2}||x||^2$ does not have bounded gradient, thus we cannot directly show that $g_\sigma = \frac{1}{2}||. - N_\sigma||^2$ has everywhere Lipschitz gradient. Instead, we can replace, in the definition of g_σ, the L^2 norm by a transformation which flattens away from a large ball. For instance, $g_\sigma^R(x) := \frac{1}{2} T_\sigma^R\,(x - N_\sigma(x))$, where

$$T_\sigma^R(x) := \begin{cases} \frac{1}{2}||x||^2 \text{ if } ||x|| \leq R \\ \exp\Big(-\Big(\frac{2}{R^2}||x||^2 - \frac{4}{R}||x|| - \log(\frac{R^2}{2}) + 3\Big)\Big) \text{ otherwise} \end{cases} \tag{4.20}$$

T_σ^R is of class $\mathcal{C}^2$ and has bounded gradient. Thus g_σ^R is ensured to have Lipschitz gradient. In practice, for R sufficiently large, the iterates produced by the algorithms presented in the next section never reach $||x|| \geq R$ and g_σ^R behaves as g_σ in practice.

Subanalycity of g_σ The ELU activation function is subanalytic as its graph can be described with a finite number of analytic functions. Moreover, ELU and its inverse map bounded sets to bounded sets. Therefore, by composition and sum (Lemma 5), the deep neural network N_σ is subanalytic. As N_σ is also Lipschitz (by composition of Lipschitz layers), it maps bounded sets to bounded sets. Thus, using that $x \to \frac{1}{2}||x||^2$ (or the above T_σ^R) is subanalytic, again by composition (Lemma 5(iii)), $g_\sigma(x) = \frac{1}{2}||x - N_\sigma(x)||^2$ (or the above $g_\sigma^R(x) = \frac{1}{2} T_\sigma^R\,(x - N_\sigma(x))$) is subanalytic. Moreover, g_σ is non-negative, thus when added to any positive subanalytic data-fidelity term f, $F = \lambda f + g_\sigma$ verifies the Kurdyka–Łojasiewicz property (Lemma 5 (i)). Let us precise that almost all data-fidelity terms are subanalytic. For instance, L^p norms (for $p \geq 0$ rational) are semialgebraic (Bolte et al., 2014) and thus subanalytic, or the Kullback-Leiber divergence is analytic and thus subanalytic.

4.2 Gradient-Step Regularization by Denoising (GSRED)

4.2.1 GSRED algorithm

Regularization by Denoising algorithms for image inverse problems are introduced in Section 2.4.2. They write, for a data-fidelity f and a denoiser D_σ

$$\text{(RED-PGD)}\quad x_{k+1} \in \mathrm{Prox}_{\tau\lambda f} \circ (\tau D_\sigma(x_k) + (1-\tau)x_k)\,. \tag{4.21}$$

$$\text{(RED-GD)}\quad x_{k+1} = \tau D_\sigma(x_k) + (1-\tau)x_k - \tau\lambda\nabla f(x_k) \tag{4.22}$$

Using our Gradient-Step denoiser $D_\sigma = \mathrm{Id} - \nabla g_\sigma$, we refer to them as Gradient-Step RED (GSRED) algorithms:

$$\textbf{(GSRED-PGD)} \quad x_{k+1} \in \mathrm{Prox}_{\tau\lambda f} \circ (x_k - \tau \nabla g_\sigma(x_k)) \,. \tag{4.23}$$

$$\textbf{(GSRED-GD)} \quad x_{k+1} = x_k - \tau(\nabla \lambda f(x_k) + \nabla g_\sigma(x_k)) \tag{4.24}$$

The Gradient-Step denoiser makes RED algorithms write as **real** first order optimization algorithm, respectively Proximal Gradient Descent (PGD) and Gradient Descent (GD), for minimizing

$$F(x) = \lambda f(x) + g_\sigma(x). \tag{4.25}$$

Therefore, using the Gradient-Step denoiser in RED algorithms is equivalent to incorporating the explicit regularization g_σ to solve an inverse problem formulated by the variational approach (4.25), with a first-order optimization algorithm.

Remark 15. *Our paper (Hurault et al., 2022a) originally presented the Gradient-Step denoiser for convergence of the PnP-HQS method. PnP-HQS, presented Section 2.4.2 writes*

$$\textit{(PnP-HQS)} \quad x_{k+1} \in D_\sigma \circ \mathrm{Prox}_{\tau\lambda f}(x_k) \tag{4.26}$$

By switching the proximal and gradient steps and with a convex relaxation of the denoising step with τ, PnP-HQS algorithm with Gradient-Step denoiser then writes

$$\begin{aligned} x_{k+1} &\in \mathrm{Prox}_{\tau\lambda f} \circ (\tau D_\sigma + (1-\tau)\,\mathrm{Id})(x_k) && (4.27)\\ &\in \mathrm{Prox}_{\tau\lambda f} \circ (\mathrm{Id} - \tau \nabla g_\sigma)(x_k) && (4.28) \end{aligned}$$

i.e. we retrieve exactly the GSRED-PGD algorithm (4.23)*.*

4.2.2 Convergence analysis

In this section, we introduce conditions on f that ensure the convergence of the GSRED iterations (4.23) and (4.24) towards a solution of (4.25). For that purpose, g_σ being nonconvex, we make use of our convergence analysis given in Theorem 6 on the PGD algorithm in the nonconvex setting. Note that, even if it means replacing g_σ by g_σ^R with large R, as explained in Section 4.1.2, we consider ∇g_σ globally L-Lipschitz.

Convergence results

Applying Proposition 20 and Theorem 6, we get the following convergence theorems for GSRED-PGD and GSRED-GD

Theorem 17 (Convergence of GSRED-PGD)**.** *Assume $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ proper, lsc, bounded from below. Let L be the Lipschitz constant of ∇g_σ. Then, for $\tau L < 1$, the iterates (x_k) given by the GSRED-PGD algorithm* (4.23) *verify*

(i) $(F(x_k))$ is non-increasing and converges.

(ii) The sequence as finite length i.e. $\sum_{k=0}^{+\infty} ||x_{k+1} - x_k||^2 < +\infty$ and $||x_{k+1} - x_k||$ converges to 0 at rate $\min_{k<K} ||x_{k+1} - x_k|| = \mathcal{O}(1/\sqrt{K})$

(iii) If f is non-negative and subanalytic, and g_σ is coercive, then the iterates (x_k) converge towards a critical point of F.

Proof. (i) and (ii) come from Proposition 20 and (iii) from Theorem 6 with g_σ in the role of the smooth function and f for the other one. We showed in Section 4.1 that g_σ is smooth, non-negative and subanalytic. Theorem 6 required F to verify the Kurdyka–Łojasiewicz (KŁ) property and the iterates to be bounded. The KŁ property of F is induced by the non-negativeness and subanalyticity of f and g_σ (see Section 3.1.2). A sufficient condition for the boundedness of the iterates is the coercivity of the objective function, that is, $\lim_{|x|\to\infty} F(x) = +\infty$ (because the non-increasing property gives $F(x_k) \leq F(x_0)$). The coercivity of F follows from the coercivity of g_σ and the fact that f is lower-bounded. □

Theorem 18 (Convergence of GSRED-GD)**.** *Assume $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ L_f-smooth, bounded from below. Let L be the Lipschitz constant of ∇g_σ. Then, for $\tau(\lambda L_f + L) < 1$, the iterates (x_k) given by the GSRED-GD algorithm* (4.24) *verify*

(i) $(F(x_k))$ *is non-increasing and converges.*

(ii) The sequence as finite length i.e. $\sum_{k=0}^{+\infty} ||x_{k+1} - x_k|| < +\infty$ *and* $||x_{k+1} - x_k||$ *converges to* 0 *at rate* $\min_{k<K} ||x_{k+1} - x_k|| = \mathcal{O}(1/\sqrt{K})$

(iii) If f is non-negative and subanalytic, and g_σ is coercive, then the iterates (x_k) converge towards a critical point of F.

Proof. This is again the application of Proposition 20 and Theorem 6 with the whole objective function $F = \lambda f + g_\sigma$ in the role of the smooth function and 0 for the other one. The rest of the arguments are identical. □

Remark 16. *Using Theorem 8, if we assume that the data-fidelity term f is M-weakly convex, the condition on the stepsize for GSRED-PGD can be relaxed to $\tau < \max(\frac{2}{L+M}, \frac{1}{L})$. In particular, for convex f, it becomes $\tau L < 2$.*

A common setting for image restoration is the convex smooth L^2 data-fidelity $f(x) = \frac{1}{2}||Ax - y||^2$ which satisfies all assumptions on f. Note however that our theory allows to deal with a broader range of degradations with nonconvex (e.g. phase retrieval) and nonsmooth (e.g. Poisson denoising) data-fidelity terms. See Section 2.1.2 for more details. In practice, it is helpful to have f proximable, *i.e.* Prox_f with closed-form formula. Otherwise, Prox_f needs to be calculated at each iteration with an internal optimization procedure.

Coercivity of g_σ Similar to Laumont et al. (2021), we can constrain F to be coercive by choosing a convex compact set $C \subset \mathbb{R}^n$ where the iterates should stay and by adding an extra term to the regularization g_σ:

$$\hat{g}_\sigma(x) = g_\sigma(x) + \frac{1}{2}||x - \Pi_C(x)||^2 = \frac{1}{2}||x - N_\sigma(x)||^2 + \frac{1}{2}||x - \Pi_C(x)||^2 \tag{4.29}$$

with Π_C the Euclidean projection on C. As g_σ is differentiable, the gradient step becomes

$$(\text{Id} - \tau\lambda\nabla_x \hat{g}_\sigma)(x) = (\text{Id} - \tau\lambda\nabla_x g_\sigma) + \tau\lambda(x - \Pi_C(x)). \tag{4.30}$$

In our experiments, we choose the compact set C as $C = [-1, 2]^n$. In practice, we observe that all the iterates always remain in C and that the extra regularization term $||x - \Pi_C(x)||^2$ is never activated.

Backtracking to handle the Lipschitz constant of ∇g_σ

The convergence of GSRED algorithms actually requires controlling the Lipschitz constant of ∇g_σ only on a small subset of images related to $\{x_k\}$. Therefore, estimating L for all images and setting the maximum stepsize τ accordingly will lead to suboptimal convergence speed. In order to avoid small stepsizes, we use a backtracking line search strategy, already used for nonconvex optimization in (Beck, 2017, Chapter 10) or (Ochs et al., 2014).

The core of the proof from Proposition 20 was to establish the following sufficient decrease property of F

$$F(x_k) \geq F\left(T_{\text{GSRED}}(x_k)\right) + \left(\frac{1}{2\tau} - \frac{L}{2}\right) ||T_{\text{GSRED}}(x_k) - x_k||^2 \tag{4.31}$$

where T_{GSRED} is the fixed-point operator iterated by the GSRED-PGD (4.23) or GSRED-GD (4.24) algorithm.

Without knowing the exact Lipschitz constant L, backtracking aims at finding the maximal stepsize τ yielding a sufficient decrease property. Given $\gamma \in (0, 1/2)$, $\eta \in [0, 1)$ and an initial stepsize $\tau_0 > 0$, the following update rule on τ is applied at each iteration k:

$$\text{while} \quad F(x_k) - F(T_{\text{GSRED}}(x_k)) < \frac{\gamma}{\tau} ||T_{\text{GSRED}}(x_k) - x_k||^2, \quad \tau \longleftarrow \eta\tau. \tag{4.32}$$

Other procedures could be investigated in future work. For instance, Li and Lin (2015) use a Barzilai-Borwein rule to initialize the backtracking line search. Scheinberg et al. (2014) and Calatroni and Chambolle (2019) have also proposed a backtracking strategy that allows for both decreasing and increasing stepsize.

Proposition 24. *Under the assumptions of Theorem 17 (respectively Theorem 18), at each iteration of the algorithm, the backtracking procedure (4.32) is finite (*i.e. *a stepsize satisfying $F(x_k) - F(T_{GSRED}(x_k)) \geq \frac{\gamma}{\tau} ||T_{GSRED}(x_k) - x_k||^2$ is found in a finite number if iterations), and with backtracking, the convergence results of Theorem 17 (respectively Theorem 18) still hold.*

Proof. From the sufficient decrease property (4.31), taking $\tau < \frac{1-2\gamma}{L}$, we get $\frac{1}{2}(\frac{1}{\tau} - L) > \frac{\gamma}{\tau}$ so that

$$F(x_k) - F(T_\tau(x_k)) > \frac{\gamma}{\tau} ||T_\tau(x_k) - x_k||^2. \tag{4.33}$$

Hence, when $\tau < \frac{1-2\gamma}{L}$, the sufficient decrease condition (4.33) is satisfied and the backtracking procedure ($\tau \longleftarrow \eta\tau$) must end. With this sufficient decrease condition, the rest of the proofs from Proposition 20 and Theorem 6 follow with the same arguments. □

4.3 Experiments

In this section, we first study the performance of the GS denoiser. Next, we empirically confirm that our GSRED algorithms are convergent while providing state-of-the art results for different IR tasks.

4.3.1 Gradient-Step denoiser

Denoising Network Architecture We choose to parameterize N_σ with the architecture DRUNet (Zhang et al., 2021) (represented in Figure 2.1), a U-Net in which residual

σ(./255)	5	15	25	50	Time (ms)
FFDNet	39.95	33.53	30.84	27.54	1.9
DnCNN	39.80	33.55	30.87	27.52	2.3
DRUNet	**40.31**	**33.97**	**31.32**	**28.08**	69.8
DRUNet *light*	40.19	33.89	31.25	28.00	6.3
GS-DRUNet	<u>40.26</u>	<u>33.90</u>	<u>31.26</u>	<u>28.01</u>	10.4

Table 4.1: Average PSNR denoising performance and runtime of our GS denoiser on 256×256 center-cropped images from the CBSD68 dataset, for various noise levels σ. While keeping small runtime, GS-DRUNet slightly outperforms its unconstrained counterpart DRUNet *light* and outdistances the deep denoisers FFDNet and DnCNN.

blocks are integrated. One first benefit of DRUNet is that it is built to take the noise level σ as input, which is consistent with our formulation. Also, the U-Net models have previously offered good results in the context of prior approximation via Denoising Score Matching (Ho et al., 2020). Furthermore, Zhang et al. (2021) show that DRUNet yields state-of-the-art results for denoising but also for PnP image restoration. In order to ensure differentiability w.r.t the input, we change RELU activations to ELU. We also limit the number of residual blocks to 2 at each scale to lower the computational burden.

Training details We use the color image training dataset proposed by Zhang et al. (2021) *i.e.* a combination of the Berkeley segmentation dataset (CBSD) (Martin et al., 2001), Waterloo Exploration Database (Ma et al., 2017), DIV2K dataset (Agustsson and Timofte, 2017) and Flick2K dataset (Lim et al., 2017). During training, the input images are corrupted with a white Gaussian noise ξ_σ with standard deviation σ randomly chosen in $[0, 50/255]$. With our parameterization (4.3) of D_σ, the network is trained to minimize the L^2 loss (4.4). For each input batch, ∇g_σ is computed with automatic differentiation. Note that for each batch, there are two backward passes through the network. The first one for computing ∇g_σ w.r.t. the input and the second one for computing the derivatives w.r.t. the parameters. While the original DRUNet is trained with L^1 loss, we stick to the L^2 loss to keep the Denoising Score Matching interpretation of g_σ.

We train the model on 128×128 patches randomly sampled from the training images, with batch size 16, during 1500 epochs. We use the ADAM optimizer with learning rate 10^{-4}, divided by 2 every 300 epochs. It takes around one week to train the model on a single Tesla P100 GPU.

Denoising results We evaluate the PSNR performance of the proposed GS denoiser (GS-DRUNet) on 256×256 color images center-cropped from the original CBSD68 dataset images (Martin et al., 2001). In Table 4.1, we compare, for various noise levels σ, our model with the DRUNet (called "DRUNet *light*") that has the same architecture as our GS-DRUNet (but with 2 residual blocks instead of 4) and that is trained (with L^2 loss) without the conservative field constraint. We also provide comparisons with the original DRUNet (Zhang et al., 2021) (with 4 residual blocks at each scale and trained with L^1 loss) and two state-of-the-art denoisers encountered in the PnP literature: FFDNet (Zhang et al., 2018) and DnCNN (Zhang et al., 2017a). For each network, we indicate in Table 4.1 the average runtime while processing a 256×256 color image on one Tesla P100 GPU.

Our GS-DRUNet denoiser, despite being constrained to be an exact conservative field,

reaches the performance of (and even slightly outperforms) its unconstrained counterpart DRUNet *light*. Second, departing from the latter, we are able to reduce the processing time by a large margin (÷7) while keeping PSNR close to the original DRUNet (around -0.05dB) and maintaining a significant PSNR gap (around +0.5dB) with other deep denoisers like DnCNN and FFDNet. Note that the time difference between GS-DRUNet and DRUNet *light* is due to the computation of ∇g_σ via backpropagation. These results indicate that GS-DRUNet is likely to yield a competitive and fast PnP algorithm.

Lipschitz constant of ∇g_σ The bound obtained in the proof of Proposition 23 is exponential in the depth of the neural network. We now provide some experiments showing that, in practice, the Lipschitz constant of ∇g_σ does not explode. We show in Figure 4.1, for various noise levels σ, the distribution of the spectral norms $||\nabla^2 g_\sigma(x)||_S$ on the training image set X, estimated with power iterations. The computed value varies a lot across images. Hence, approximating the Lipschitz constant of ∇g_σ with $L = \max_{x \in X} ||\nabla^2 g_\sigma(x)||_S$ would lead to under-estimated stepsizes and slow convergence on most images. Backtracking solves this issue by finding at each iteration the optimal stepsize allowing sufficient decrease of the objective function.

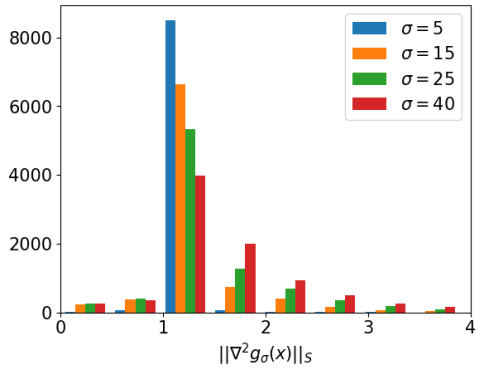


Figure 4.1: Histogram of the values of the spectral norm $||\nabla^2 g_\sigma(x)||_S$ evaluated on 128×128 images from the training dataset, degraded with white Gaussian noise with various standard deviations σ (./255). Figure best seen in color.

4.3.2 Plug-and-play image restoration

We show in this section that, in addition to being convergent, our GSRED algorithms reach state-of-the-art performance, among plug-and-play methods, in deblurring and super-resolution (Sections 4.3.2 and 4.3.2) and realize relevant inpainting (Section 4.3.2). We seek an estimate of a clean image $x \in \mathbb{R}^n$, from an observation obtained as $y = Ax + \xi_\nu \in \mathbb{R}^m$, with A a $m \times n$ degradation matrix and ξ_ν a white Gaussian noise with zero mean and standard deviation ν. The minimized objective function is

$$F(x) = \lambda f + g_\sigma = \frac{\lambda}{2} ||Ax - y||^2 + \frac{1}{2} ||N_\sigma(x) - x||^2 . \tag{4.34}$$

The convergence of the iterates given by GSRED-PGD and GSRED-GD algorithms is guaranteed by Theorems 17 and 18. We also demonstrate in Section 4.3.2 that our framework can be extended to other kinds of objective functions. For example, inpainting noise-free input images leads to a non-differentiable data-fidelity term f.

Due to the large computational time of some compared methods, we use for evaluation and comparison a subset of 10 color images taken from the CBSD68 dataset (CBSD10) together with the 3 famous set3C images (butterfly, leaves, and starfish). All images are center-cropped to the size 256×256. For each IR problem, we provide default values for the parameters σ and λ that can be used to treat successfully a large class of images and degradations. The influence of both parameters is then analyzed in Section 4.3.3. Performance can be marginally improved by automatic hyperparameter tuning strategies. For example, Vidal et al. (2020) propose a Bayesian method for setting the regularization parameter and Wei et al. (2020) employ deep reinforcement learning to train a policy network yielding well-suited parameters for PnP.

In our experiments, backtracking is performed with $\eta = 0.9$ and $\gamma = 0.1$. We set the initial stepsize of GSRED-PGD (resp. GSRED-PG) to $\tau = 1$ (resp. $\tau = \frac{2}{\lambda+1}$) . At the first iteration, the Gradient-Step in GSRED-PGD is thus exactly D_σ, and GSRED-PGD exactly corresponds to PnP-HQS (see Remark 15). The algorithm is initialized with a proximal step and terminates when the norm between consecutive values of the objective function is less than 10^{-5} or the number of iterations exceeds $K = 1000$. After convergence of both algorithms, we found it useful to apply an extra denoising step in order to discard the residual noise.

Deblurring

For image deblurring, the degradation operator $A = H$ is a convolution performed with circular boundary conditions. Therefore, $H = \mathcal{F}^*\Lambda\mathcal{F}$, where $\mathcal{F}$ is the orthogonal matrix of the discrete Fourier transform (and $\mathcal{F}^*$ its inverse), and Λ is a diagonal matrix. The proximal operator of the data-fidelity term $f(x) = \frac{1}{2}\,||Hx - y||^2$ involves only element-wise inversion, and writes

$$\mathrm{Prox}_{\tau f}(z) = \mathcal{F}^*(I_n + \tau\Lambda^*\Lambda)^{-1}\mathcal{F}(\tau H^T y + z). \tag{4.35}$$

We demonstrate the effectiveness of our method on a large variety of blur kernels (represented in Table 4.2) and noise levels. As in Zhang et al. (2017b); Pesquet et al. (2021); Zhang et al. (2021), we use the 8 real-world camera-shake kernels proposed in Levin et al. (2009) as well as the 9×9 uniform kernel and the 25×25 Gaussian kernel with standard deviation 1.6 (as in Romano et al. (2017)). We consider Gaussian noise with 3 noise levels $\nu \in \{2.55, 7.65, 12.75\}/255$ *i.e.* $\nu \in \{0.01, 0.03, 0.05\}$.

For both GSRED algorithms and for all noise levels, we set $\sigma = 1.8\nu$, $1/\lambda = 0.1$ for motion blur (kernels (a) to (h)) and $1/\lambda = 0.075$ for static blur (kernels (i) and (j)). Initialization is done with $x_0 = y$ and we study in Section 4.3.3 the robustness to initialization.

We compare in Table 4.2 our algorithms (GSRED-PGD and GSRED-GD) against the patch-based method EPLL (Zoran and Weiss, 2011; Hurault et al., 2018), and the deep PnP methods IRCNN (Zhang et al., 2017b), DPIR (Zhang et al., 2021) and MMO (Pesquet et al., 2021).Both IRCNN and DPIR use PnP-HQS with a fast decrease of τ and σ in a few iterations (8 iterations for DPIR) without guarantee of convergence. DPIR uses the DRUNet denoiser from Table 4.1. MMO is the only compared method that guarantees convergence by plugging a DnCNN denoiser trained with Lipschitz constraints (but the network provided by the authors was trained for very low noise level).

GSRED-PGD and GSRED-GD perform almost equally well in terms of PSNR. However, we observe that GSRED-PGD converges faster in practice than GSRED-GD (See for

example Figure 4.2). Among all methods, they closely follow the state-of-the-art DPIR in terms of PSNR for low noise level but perform equally or better for higher noise levels. Other comparisons are also conducted on the full CBSD68 dataset in Table 4.3. These results exhibit that both GSRED algorithms reach state-of-the-art in deblurring for a variety of kernels and noise levels. We underline that the convergence of GSRED algorithms are guaranteed, whereas DPIR can asymptotically diverge (see Section 4.3.3).

For qualitative comparison, we show in Figure 4.2 and 4.3 the deblurring obtained with various methods on two different images. Note that our algorithms, compared to competing methods, can recover the sharpest edges and the finest textures. We also give convergence curves that empirically confirm the convergence of the values $F(x_k)$ and of the residual $\min_{0\leq i\leq k} ||x_{i+1} - x_i||^2 / ||x_0||^2$. The plotted evolution of the PSNR illustrates that the minimization of F coincides with the maximization of the PSNR, which supports the choice of the optimized functional $F = f + \lambda g_\sigma$. Note also that the empirical convergence rate is faster than the $\mathcal{O}(\frac{1}{k})$ theoretical worst case rate established in the convergence theorems.

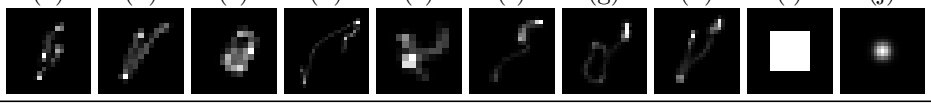
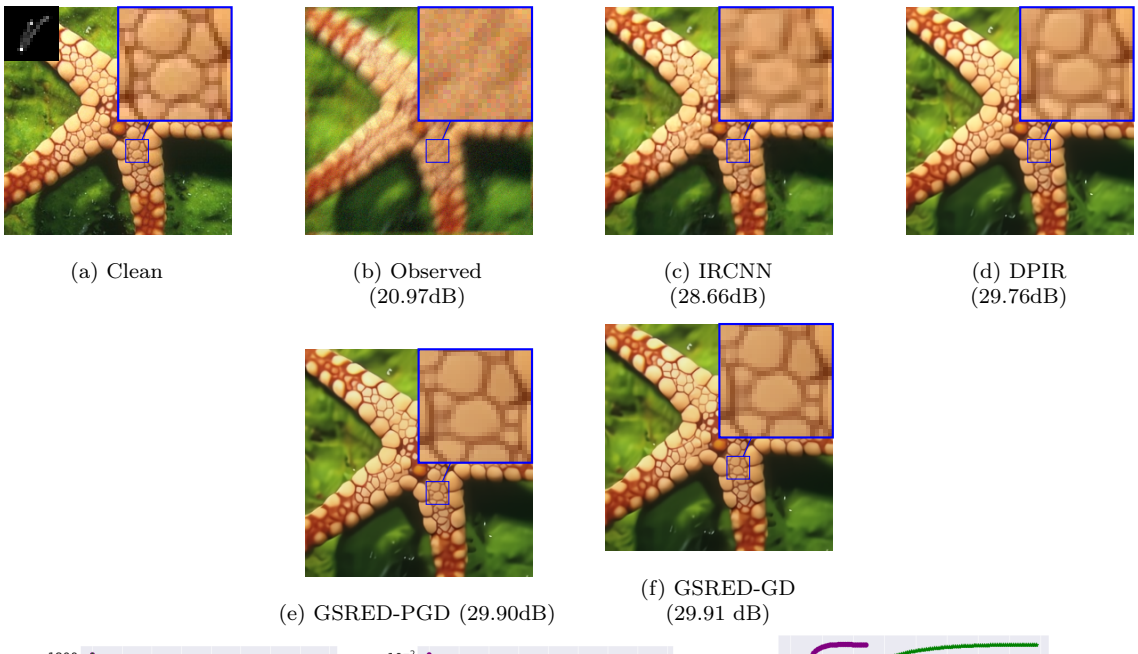
		(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	
ν	Method											*Avg*
0.01	EPLL	28.32	28.24	28.36	25.80	29.61	27.15	26.90	26.69	25.84	26.49	*27.34*
	IRCNN	32.96	32.62	32.53	32.44	33.51	33.62	32.54	32.20	28.11	29.19	*31.97*
	MMO	32.35	32.06	32.24	31.67	31.77	33.17	32.30	31.80	27.81	**29.26**	*31.44*
	DPIR	**33.76**	**33.30**	**33.04**	**33.09**	**34.10**	**34.34**	**33.06**	**32.77**	**28.34**	29.16	***32.50***
	GSRED-PGD	33.52	33.07	32.91	32.83	34.07	34.25	32.96	32.54	28.11	29.03	*32.33*
	GSRED-GD	33.51	33.01	32.86	32.74	33.99	34.21	32.88	32.49	27.84	28.76	*32.23*
0.03	EPLL	25.31	25.12	25.82	23.75	26.99	25.23	25.00	24.59	24.34	25.43	*25.16*
	IRCNN	28.96	28.65	28.90	28.38	30.03	29.87	28.92	28.52	25.92	27.64	*28.58*
	DPIR	**29.38**	**29.06**	**29.21**	**28.77**	30.22	**30.23**	**29.34**	28.90	26.19	27.81	***28.91***
	GSRED-PGD	29.22	28.89	29.20	28.60	**30.32**	30.21	29.32	**28.92**	**26.38**	**27.89**	*28.90*
	GSRED-GD	29.26	28.92	29.16	28.65	30.29	30.20	29.22	28.81	26.16	27.68	28.84
0.05	EPLL	24.08	23.91	24.78	22.57	25.68	23.98	23.70	23.19	23.75	24.78	*24.04*
	IRCNN	27.00	26.74	27.25	26.37	28.29	28.06	27.22	26.81	24.85	26.83	*26.94*
	DPIR	**27.52**	**27.35**	**27.73**	**27.02**	28.63	**28.46**	27.79	27.30	25.25	27.11	*27.42*
	GSRED-PGD	27.45	27.28	27.70	26.98	**28.68**	28.44	**27.81**	**27.38**	**25.49**	**27.15**	***27.44***
	GSRED-GD	27.47	27.28	27.63	26.98	28.66	28.44	27.71	27.27	25.35	26.98	*27.38*

Table 4.2: PSNR(dB) comparison of image deblurring methods on CBSD10 with various blur kernels k and noise levels ν. Best and second-best results are bold and underlined.

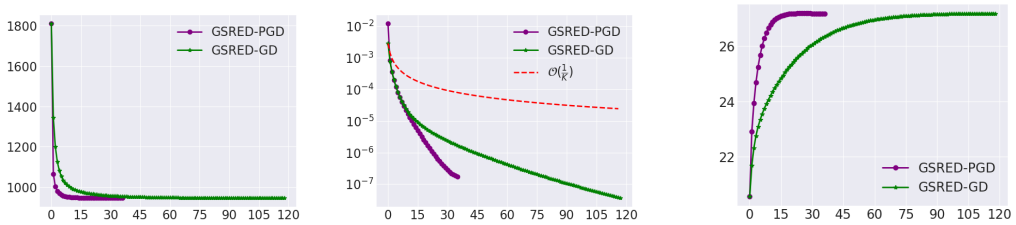
		(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	
ν	Method											*Avg*
0.01	IRCNN	32.47	32.14	31.94	31.97	32.94	33.13	31.92	31.62	27.57	**28.45**	*31.42*
	DPIR	**33.26**	**32.82**	**32.48**	**32.65**	**33.57**	**33.85**	**32.49**	**32.22**	**27.65**	28.26	***31.93***
	GSRED-PGD	32.95	32.54	32.26	32.31	33.41	33.71	32.29	31.92	27.43	28.17	*31.70*
0.03	IRCNN	28.43	28.11	28.28	27.87	29.42	29.21	28.37	27.97	25.52	26.96	*28.01*
	DPIR	**28.88**	**28.53**	**28.55**	**28.30**	29.58	**29.62**	**28.69**	28.28	25.60	26.96	***28.30***
	GSRED-PGD	28.64	28.32	**28.55**	28.06	**29.71**	29.60	**28.69**	**28.31**	**25.79**	**27.10**	*28.28*
0.05	IRCNN	26.73	26.42	26.73	26.13	27.69	27.39	26.69	26.33	24.68	26.18	*26.40*
	DPIR	**27.04**	**26.80**	**27.07**	**26.53**	28.00	27.85	27.17	26.72	24.75	26.32	*26.82*
	GSRED-PGD	26.93	26.72	**27.07**	26.45	**28.09**	**27.87**	**27.21**	**26.82**	**25.02**	**26.45**	***26.86***

Table 4.3: PSNR(dB) performance of the fastest methods for image deblurring on the full CBSD68 dataset, in the same conditions as Table 4.2. On CBSD10 (Table 4.2) or on CBSD68, we observed very similar PSNR gaps between the compared methods, which confirms that CBSD10 is large enough to compare accurately the methods.



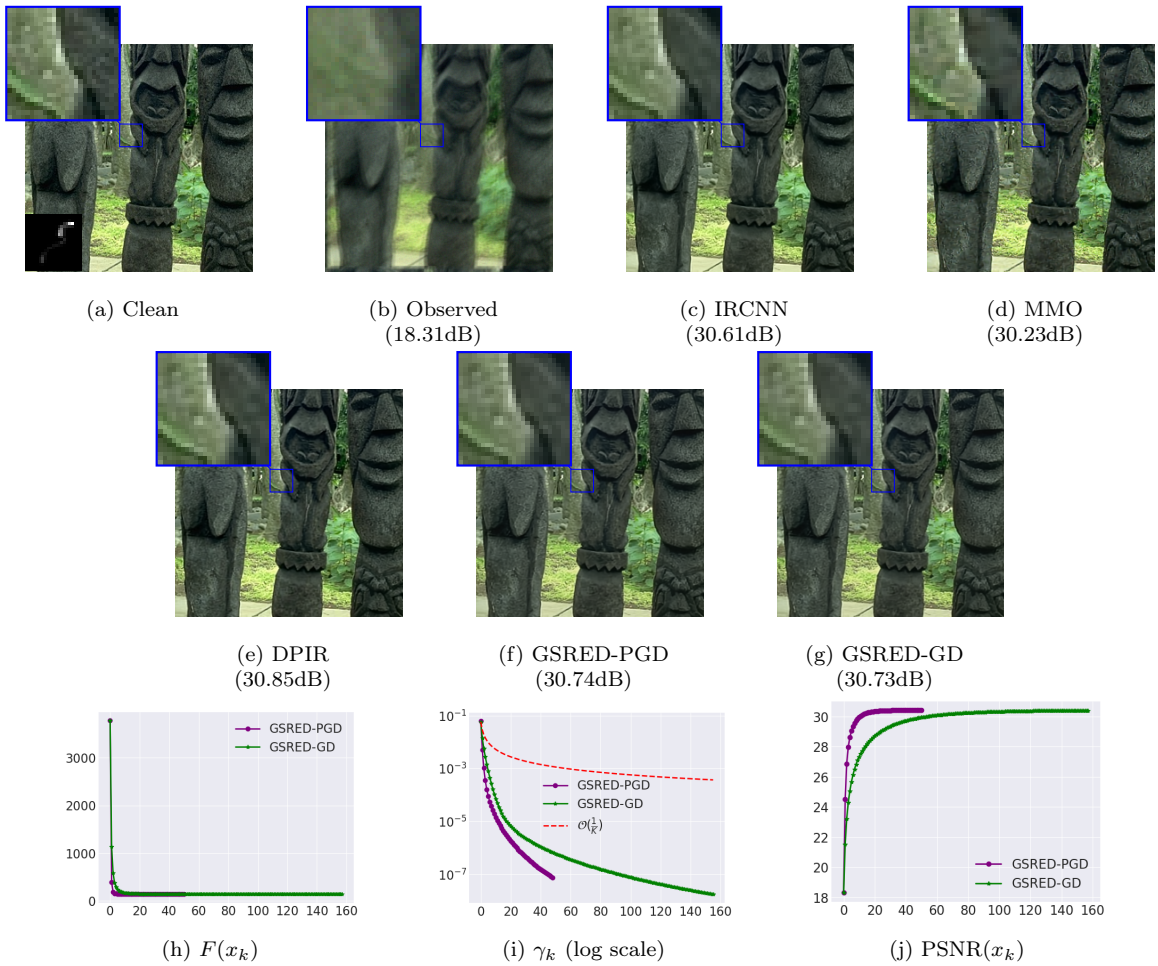
(a) Clean (b) Observed (20.97dB) (c) IRCNN (28.66dB) (d) DPIR (29.76dB)

(e) GSRED-PGD (29.90dB) (f) GSRED-GD (29.91 dB)



(g) $F(x_k)$ (h) γ_k (log scale) (i) PSNR(x_k)

Figure 4.2: Deblurring with various methods of "starfish" degraded with the indicated blur kernel and input noise level $\nu = 0.03$. Note that our algorithms better recover the structures. In (h), (i) and (j), we plot the evolution of the objective value $F(x_k)$, the residual $\gamma_k = \min_{0 \le i \le k} ||x_{i+1} - x_i||^2 / ||x_0||^2$ and the PSNR.



(a) Clean (b) Observed (18.31dB) (c) IRCNN (30.61dB) (d) MMO (30.23dB)

(e) DPIR (30.85dB) (f) GSRED-PGD (30.74dB) (g) GSRED-GD (30.73dB)

(h) $F(x_k)$ (i) γ_k (log scale) (j) PSNR(x_k)

Figure 4.3: Deblurring with various methods of an image from CSBD10 degraded with the indicated blur kernel and input noise level $\nu = 0.01$.

Super-resolution

For single image super-resolution, the low-resolution image $y \in \mathbb{R}^m$ is obtained from the high-resolution one $x \in \mathbb{R}^n$ via $y = SHx + \xi_\nu$ where $H \in \mathbb{R}^{n \times n}$ is the convolution with an antialiasing kernel. The matrix S is the standard s-fold downsampling matrix of size $m \times n$ and $n = s^2 \times m$. In this context, we make use of the closed-form calculation of the proximal map for the data-fidelity term $f(x) = \frac{1}{2}\,||SHx - y||^2$, given by Zhao et al. (2016):

$$\text{Prox}_{\tau f}(z) = \hat{z}_\tau - \frac{1}{s^2}\mathcal{F}^*\underline{\Lambda}^*\left(I_m + \frac{\tau}{s^2}\underline{\Lambda}\underline{\Lambda}^*\right)^{-1}\underline{\Lambda}\mathcal{F}\hat{z}_\tau, \tag{4.36}$$

where $\hat{z}_\tau = \tau H^T S^T y + z$ and $\underline{\Lambda} = [\Lambda_1, \ldots, \Lambda_{s^2}] \in \mathbb{R}^{m \times n}$, with $\Lambda = \text{diag}(\Lambda_1, \ldots, \Lambda_{s^2})$ a block-diagonal decomposition according to a $s \times s$ paving of the Fourier domain. Note that $I_m + \frac{\tau}{d}\underline{\Lambda}\underline{\Lambda}^*$ is a $m \times m$ diagonal matrix, and its inverse is computed element-wise. As expected, with $s = 1$, equation (4.36) comes down to equation (4.35).

As in (Zhang et al., 2021), we evaluate super-resolution performance on 8 Gaussian blur kernels represented in Table 4.4: 4 isotropic kernels with different standard deviations (0.7, 1.2, 1.6 and 2.0) and 4 anisotropic kernels. Results are averaged between isotropic and anisotropic. We consider downsampled images at scale $s = 2$ and $s = 3$ and Gaussian noise with 3 different noise levels $\nu \in \{0.01, 0.03, 0.05\}$. Our methods are compared

against bicubic upsampling, IRCNN ("IRCNN+" from Zhang et al. (2021)) and DPIR. GSRED algorithms are the only compared PnP methods with convergence guarantees. For both GSRED algorithms, all our results are obtained with $1/\lambda = 0.065$ and $\sigma = 2\nu$. Initialization of z_0 is done with bicubic upsampling of y (with a shift correction (Zhang et al., 2021)).

GSRED-PGD outperforms in PSNR all other PnP algorithms and GSRED-GD performs on par with DPIR, over the considered range of blur kernels, noise levels and scale factors.

We show in Figures 4.4 and 4.5 the super-resolution of two images downsampled respectively by factor 2 and 3, with isotropic kernels and noise levels respectively $\nu = 0.03$ and $\nu = 0.01$. GSRED algorithms recover accurately structures and color, while converging, as attested by the plotted residual and functional convergence curves.

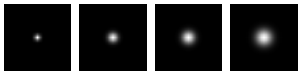
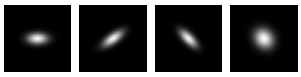
Kernels	Method	$s=2$			$s=3$			*Avg*
		$\nu=0.01$	$\nu=0.03$	$\nu=0.05$	$\nu=0.01$	$\nu=0.03$	$\nu=0.05$	
	Bicubic	24.85	23.96	22.79	23.14	22.52	21.62	*23.15*
	IRCNN	27.43	26.22	25.86	26.12	25.11	24.79	*25.92*
	DPIR	28.62	27.30	26.47	**26.88**	25.96	25.22	*26.74*
	GSRED-PGD	**28.77**	**27.54**	**26.63**	26.85	**26.05**	**25.29**	***26.86***
	GSRED-GD	28.64	27.46	26.59	26.66	25.96	25.22	*26.76*
	Bicubic	23.38	22.71	21.78	22.65	22.08	21.25	*22.31*
	IRCNN	25.83	24.89	24.59	25.36	24.36	23.95	*24.83*
	DPIR	**26.84**	25.59	24.89	**26.24**	24.98	**24.32**	*25.48*
	GSRED-PGD	26.80	**25.73**	**25.03**	26.18	**25.08**	24.31	***25.52***
	GSRED-GD	26.48	25.56	24.94	25.86	24.91	24.23	*25.33*

Table 4.4: PSNR(dB) comparison of image super-resolution methods on CBSD10 with various scales s, blur kernels k and noise levels ν. PNSR results are averaged over kernels at each row. The best and second-best results are respectively in bold and underlined.

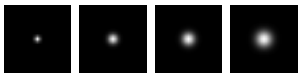
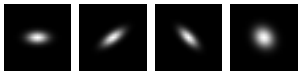
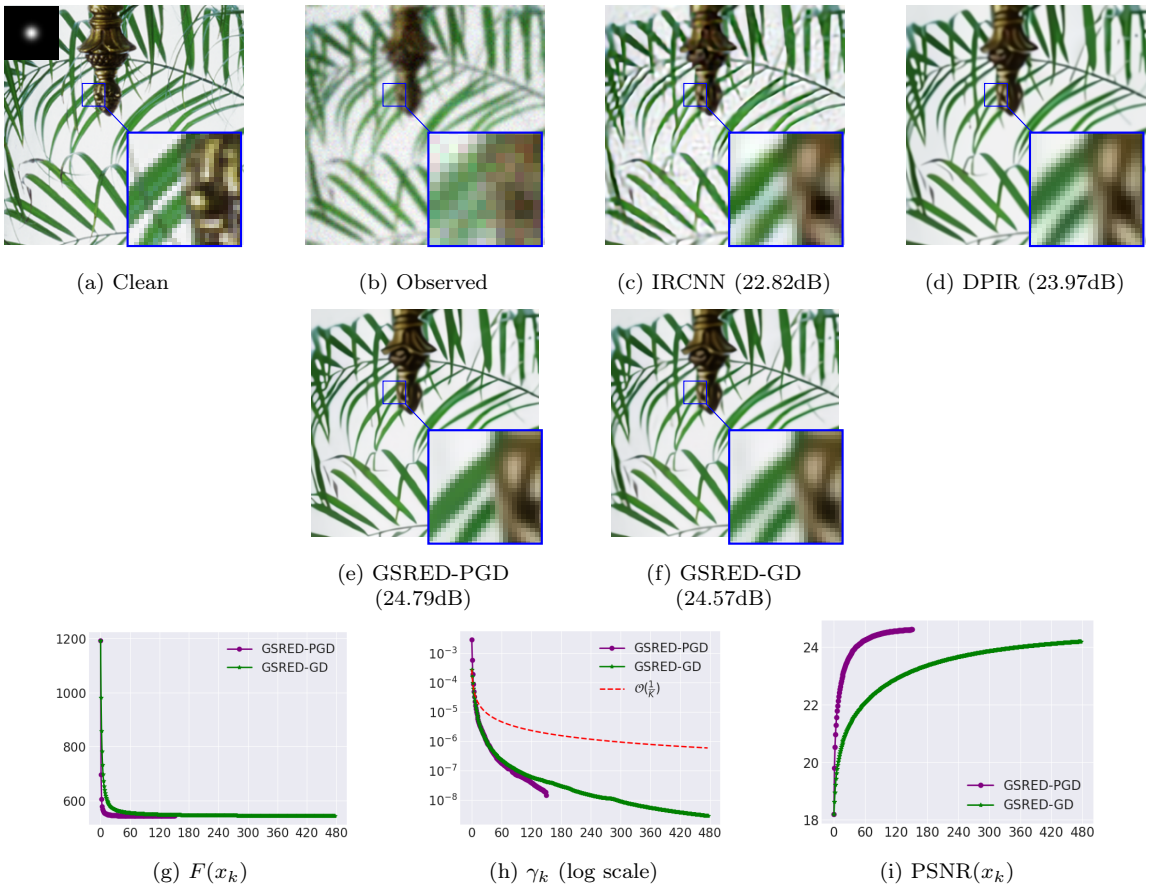
Kernels	Method	$s=2$			$s=3$			*Avg*
		$\nu=0.01$	$\nu=0.03$	$\nu=0.05$	$\nu=0.01$	$\nu=0.03$	$\nu=0.05$	
	IRCNN	26.97	25.86	25.45	25.60	24.72	24.38	*25.50*
	DPIR	27.79	26.58	25.83	**26.05**	25.27	24.66	*26.03*
	GSRED-PGD	**27.88**	**26.81**	**26.01**	25.97	**25.35**	**24.74**	***26.13***
	IRCNN	25.41	24.52	24.18	24.94	24.04	23.61	*24.45*
	DPIR	**26.08**	24.99	24.39	**25.53**	24.46	23.80	*24.88*
	GSRED-PGD	25.98	**25.07**	**24.53**	25.47	**24.56**	**23.92**	***24.92***

Table 4.5: PSNR(dB) performance of the fastest method (IRCNN/DPIR/GS-PnP) for image super-resolution on the full CBSD68 dataset with various blur kernels k and noise levels ν, in the same conditions as Table 4.4. Once again, on CBSD10 (Table 4.4) or on CBSD68, we observed very similar performance gaps between the compared methods, which again confirms that CBSD10 is large enough to compare accurately the PnP methods.

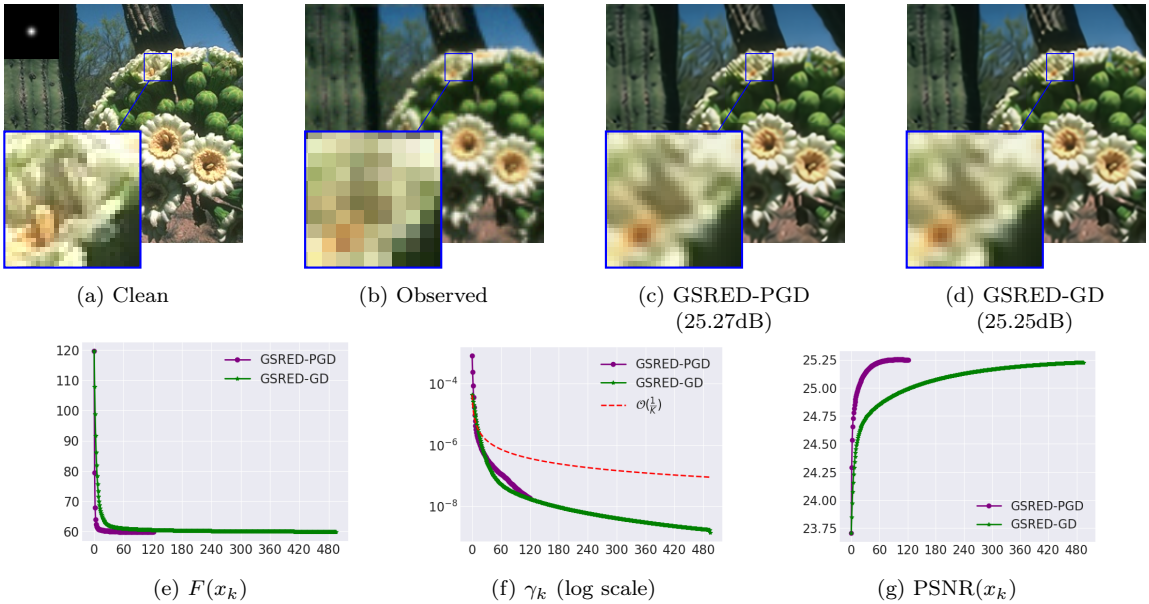


(a) Clean (b) Observed (c) IRCNN (22.82dB) (d) DPIR (23.97dB)

(e) GSRED-PGD (24.79dB) (f) GSRED-GD (24.57dB)

(g) $F(x_k)$ (h) γ_k (log scale) (i) PSNR(x_k)

Figure 4.4: Super-resolution with various methods on "leaves" downsampled by 2, with the indicated blur kernel and input noise level $\nu = 0.03$. Note that our algorithms recover the sharpest leaves. In (h), (i) and (j), we plot the evolution of the objective value $F(x_k)$, the residual $\gamma_k = \min_{0 \leq i \leq k} ||x_{i+1} - x_i||^2 / ||x_0||^2$ and the PSNR.



(a) Clean (b) Observed (c) GSRED-PGD (25.27dB) (d) GSRED-GD (25.25dB)

(e) $F(x_k)$ (f) γ_k (log scale) (g) PSNR(x_k)

Figure 4.5: Super-resolution with our proposed algorithms on an image from CBSD10 downsampled by 3, with the indicated blur kernel and input noise level $\nu = 0.01$.

Inpainting (with non-differentiable data-fidelity term)

We now propose to apply our GSRED-PGD algorithm to image inpainting with the degradation model

$$y = Ax \tag{4.37}$$

where A is a diagonal matrix with values in $\{0, 1\}$. For inpainting, no noise is added to the degraded image. In this context, the data-fidelity term is the indicator function of $A^{-1}(\{y\}) = \{x \mid Ax = y\}$: $f(y) = \imath_{A^{-1}(\{y\})}$ (which, by definition, equals 0 on $A^{-1}(\{y\})$ and $+\infty$ elsewhere). Despite being non-differentiable, f still verifies the assumptions of Theorems 17 and convergence of GSRED-PGD is theoretically ensured. The proximal map becomes the orthogonal projection $\Pi_{A^{-1}(\{y\})}$

$$\mathrm{Prox}_{\tau f}(x) = \Pi_{A^{-1}(\{y\})}(x) = Ay - Ax + x \tag{4.38}$$

In our experiments, the diagonal of A is filled with Bernoulli random variables with parameter $p = 0.5$. We run our PnP algorithm with $\sigma = 10/255$. Given the form of f, we do not use the backtracking strategy and keep a fixed stepsize. Even if we do not exactly know the Lipschitz constant of ∇g_σ, we observed in Figure 4.1 that, for small noise, it was almost always estimated as slightly larger than 1. We thus choose $\tau = 1$ and empirically confirm convergence with this choice in follow-up experiments (see Figure 4.6). The algorithm is initialized with $x_0 = y + 0.5(\mathrm{Id} - A)y$ (masked pixels with value 0.5) and terminates when the number of iterations exceeds $K = 100$. We found it useful to run the first 10 iterations of the algorithm at a larger noise level $\sigma = 50/255$. The reason is that the algorithm first deals with high-scales structures that helps for creating more precise details in the second run.

We show Figure 4.6 inpainting results on set3C images. Our algorithm restores the input images with high accuracy, including its small details. Furthermore, convergence of the residual at rate $\mathcal{O}(\frac{1}{k})$ is empirically confirmed.

4.3.3 Additional experiments

Visualization of g_σ In order to understand better the regularization g_σ, we first propose to visualize $g_\sigma(y)$ for y which has been degraded from a clean image x with different levels of degradations. In Figures 4.7, we plot the average value of $g_\sigma(y)$ averaged over the CBSD68 dataset, for respectively,

(i) Gaussian noise: $y \sim \mathcal{N}(x, \nu^2 \,\mathrm{Id})$.

(ii) Poisson noise: $y \sim \frac{1}{\alpha}\mathcal{P}(\alpha x)$ with $\mathcal{P}$ Poisson distribution.

(ii) Blur: $y = k_\nu * x$ with k_ν Gaussian kernel of std ν and size $(15, 15)$.

(iii) Missing pixels: $y = M_p x$ where M_p diagonal with diagonal coefficients following Bernoulli distributions with probability $1 - p$ (*i.e.* each pixel is masked with probability p).

For Gaussian and Poisson noise, we observe the expected evolution: the more noisy the image, the higher the value of g_σ. The plots are more surprising when g_σ is evaluated on blurry images or images with missing pixels. The regularization seems to favor smooth and piecewise constant regions. Indeed, on Figures 4.7 (c), g_σ decreases when the input gets

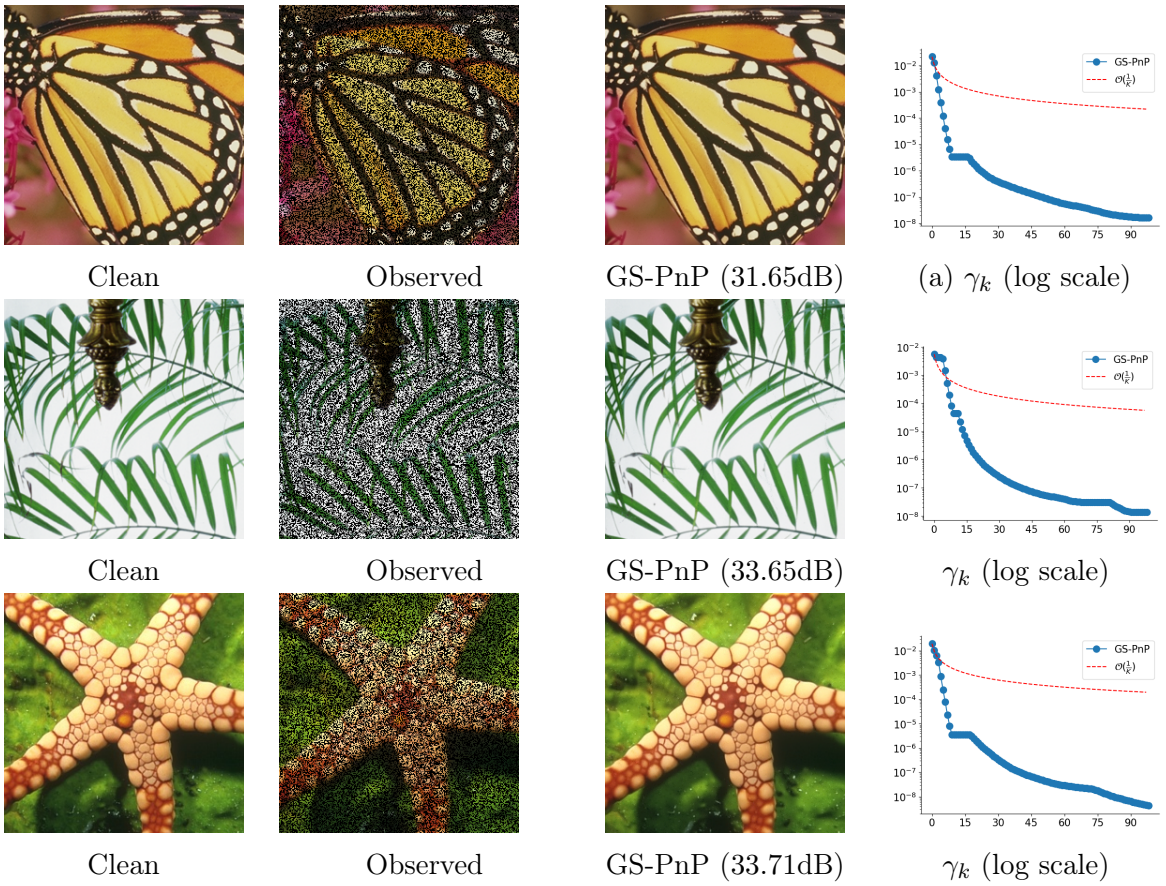


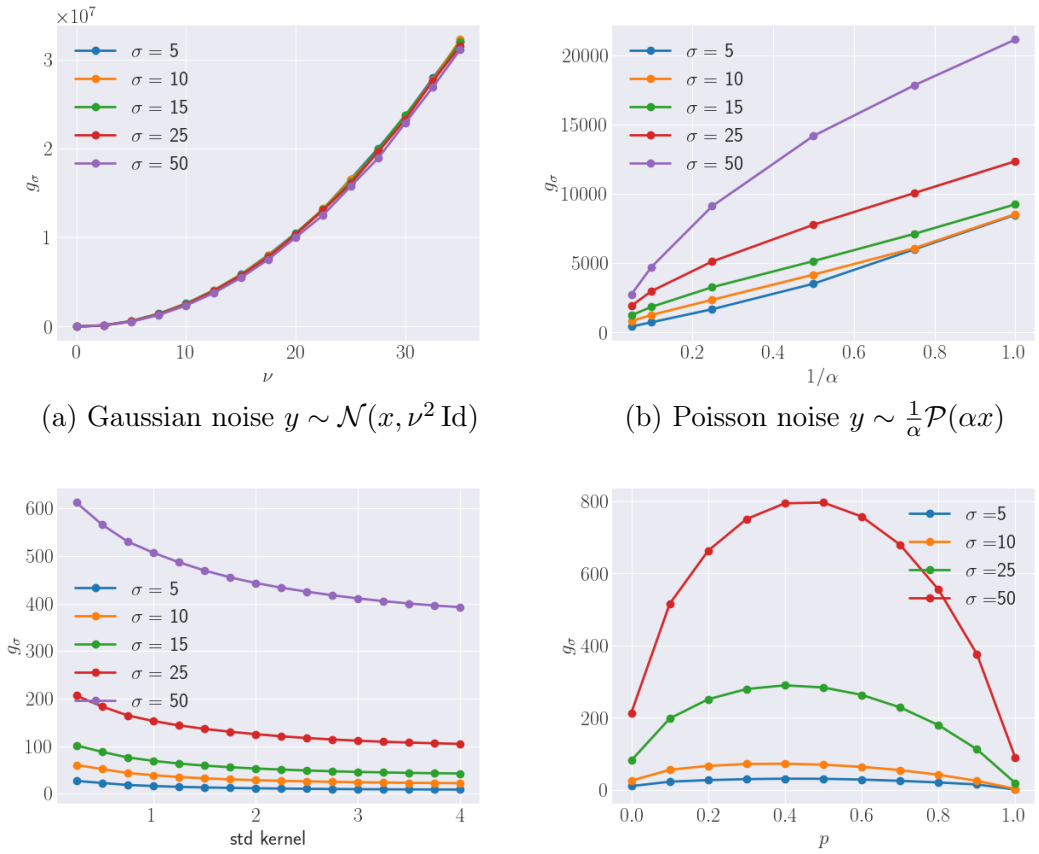
Figure 4.6: Inpainting results on set3C with pixels randomly masked with probability $p = 0.5$. In the last column, we show the evolution of $\gamma_k = \min_{0 \leq i \leq k} ||x_{i+1} - x_i||^2 / ||x_0||^2$ along the iterations.

more and more blurry. For missing pixels, in Figures 4.7 (d), g_σ first increases when the proportion of masked pixels increases, until approximately half of the pixels are masked ($p < 0.5$). Then, for p from 0.5 to 1, g_σ re-decreases with p. For $p > 0.5$, there are more and more piecewise constant black regions on the image. g_σ seems to favor such images.

Expansiveness of the denoiser As g_σ is not necessarily convex, our GS-DRUNet denoiser $D_\sigma = \mathrm{Id} - \nabla g_\sigma$ is not necessarily nonexpansive and neither is the gradient-descent step $\mathrm{Id} - \lambda\tau\nabla g_\sigma$. This is not an issue as, unlike previous theoretical PnP studies Terris et al. (2020); Reehorst and Schniter (2018), our convergence results do not require a nonexpansive denoising step. To advocate that our method converges without this assumption, we show in Figure 4.8 the evolution of $\frac{||D_\sigma(x_{k+1}) - D_\sigma(x_k)||}{||x_{k+1} - x_k||}$ along the algorithm that was run to obtain the super-resolution results of Figure 4.4. Note that the Lipschitz constant of D_σ goes above 1 but convergence is still observed as shown by the convergence curves in Figure 4.4.

Influence of the parameters We study more deeply the influence of the parameters involved in the GSRED-PGD algorithm. Three parameters are involved: the stepsize τ, the denoiser level σ and the regularization parameter λ.

- The stepsize τ is automatically tuned with backtracking and is not tweaked heuristically, contrary to other competing PnP methods like DPIR.
- The first regularization parameter σ is linked to the used denoiser.



(a) Gaussian noise $y \sim \mathcal{N}(x, \nu^2 \,\mathrm{Id})$

(b) Poisson noise $y \sim \frac{1}{\alpha}\mathcal{P}(\alpha x)$

(c) Gaussian blur $y = k_\nu * x$

(d) Missing pixels $y = M_p x$

Figure 4.7: Evolution, for various σ, of $g_\sigma(y)$ where y has been obtained from x with different degradations. (a) Gaussian noise with increasing std ν. (b) Poisson noise with decreasing parameter α (c) Blur with Gaussian kernel of increasing std ν (d) Each pixel is masked with increasing probability $1 - p$.

- The second regularization parameter λ is introduced in order to target the objective function $\lambda f + g_\sigma$. It is a classical formulation of inverse problems, and the trade-off parameter λ is usually tuned manually.

Thus, we have two parameters that we are free to tune manually. One additional motivation for keeping both λ and σ as regularization parameters is to be able to use our PnP algorithm with noise-blind denoisers like DnCNN that are independent on σ. In practice, in our experiments, we first roughly estimated σ proportionally to the input noise level ν and tweaked λ more precisely. Note that, for each inverse problem, our parameters λ and σ are fixed for a large variety of kernels, images, and noise levels ν. The parameters are not optimized for each image.

Figure 4.9 and Figure 4.10 respectively plot the average PSNR when deblurring the CBSD10 images with different values of $1/\lambda$ and σ, at fixed $\nu = 0.03$. Both parameters control the strength of the regularization. We observe that λ and σ have a similar influence on the output: for large λ or small σ, the regularization involved by the denoising pass is not sufficient to counteract the noise amplification done by the proximal steps with large τ (when $\tau \to \infty$, $\mathrm{Prox}_{\tau f}$ for $f(x) = \frac{1}{2}\,||Ax - y||^2$ tends to the pseudo-inverse of A). On the contrary, as expected, decreasing λ or increasing σ tends to over-smooth the output result.

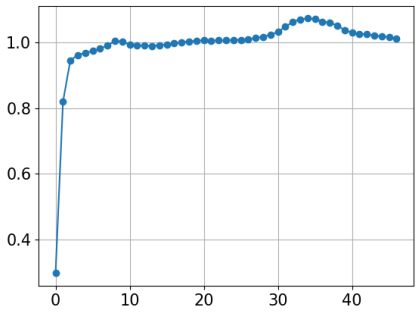


Figure 4.8: Lipschitz constant of D_σ along the iterates of the algorithm when performing the super-resolution experiments presented in Figure 4.4. Note that the Lipschitz constant goes above 1 *i.e.* D_σ is not nonexpansive, but we still empirically verified convergence (see the convergence curves in Figure 4.4).

Influence of the initialization In Figure 4.11, we examine the robustness of GSRED-PGD to the initialization. As can be seen on this experiment, the output image does not change much even for relatively large perturbation of the initialization. We thus observe a robustness to the initialization, both in terms of visual aspect and PSNR. We also observed that initializing with a uniform image does not change the output of the algorithm. We suggest that this robustness comes from the first proximal steps on the data-fidelity term (with a large τ), which prevent the algorithm to be stuck in a poor local minimum. Note that the use of large τ in the beginning of the algorithm is possible thanks to the backtracking procedure.

Divergence of DPIR (Zhang et al., 2021) We illustrate here that, contrary to our method, the DPIR algorithm is not guaranteed to converge and can even easily diverge. In Figure 4.12, we plot the convergence curves of both DPIR and GSRED-PGD when deblurring the "starfish" image degraded with a motion kernel and $\nu = 0.01$. In the original DPIR paper (Zhang et al., 2021), only 8 iterations are used with decreasing τ and σ. More precisely, σ decreases uniformly in log-scale from 49 to the input noise level ν, and τ is set proportional to σ^2. In order to study the asymptotic behavior of the method, we propose two strategies to run DPIR with 1000 iterations:

(i) Decreasing σ from 49 to ν over 1000 iterations instead of 8 (Figure 4.12, row 1).

(ii) Decreasing σ from 49 to ν in 8 iterations, and then keep the last values of σ and τ unchanged for the remaining iterations (Figure 4.12, row 2).

As illustrated by the plot of $\sum_{i \leq k} ||x_{i+1} - x_i||^2$ (third column), DPIR fails to converge with both strategies, even if the residual $||x_{k+1} - x_k||^2$ tends to decrease with the second strategy. This divergence also involves a loss of restoration performance in terms of PSNR (first column). On the other hand, as theoretically shown in this paper, the residual $||x_{k+1} - x_k||^2$ with GSRED-PGD tends to 0 (reaches $\sim 10^{-13}$ before the activation of backtracking, versus $\sim 10^{-4}$ for DPIR) and its series converges.

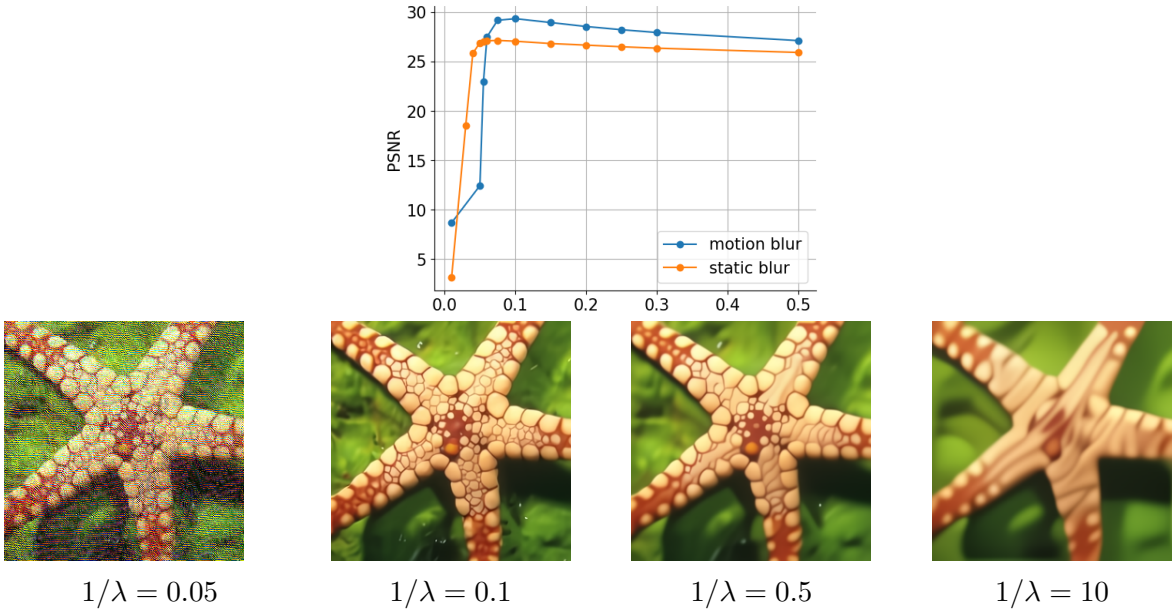


Figure 4.9: Influence of the choice of the parameter λ for deblurring. Top: average PSNR when deblurring the images of CBSD10, blurred with motion blurs or static blurs, w.r.t $1/\lambda$. The other parameters remain unchanged. Bottom: visual results when deblurring "starfish" with various λ (in the same conditions as Figure 4.2).

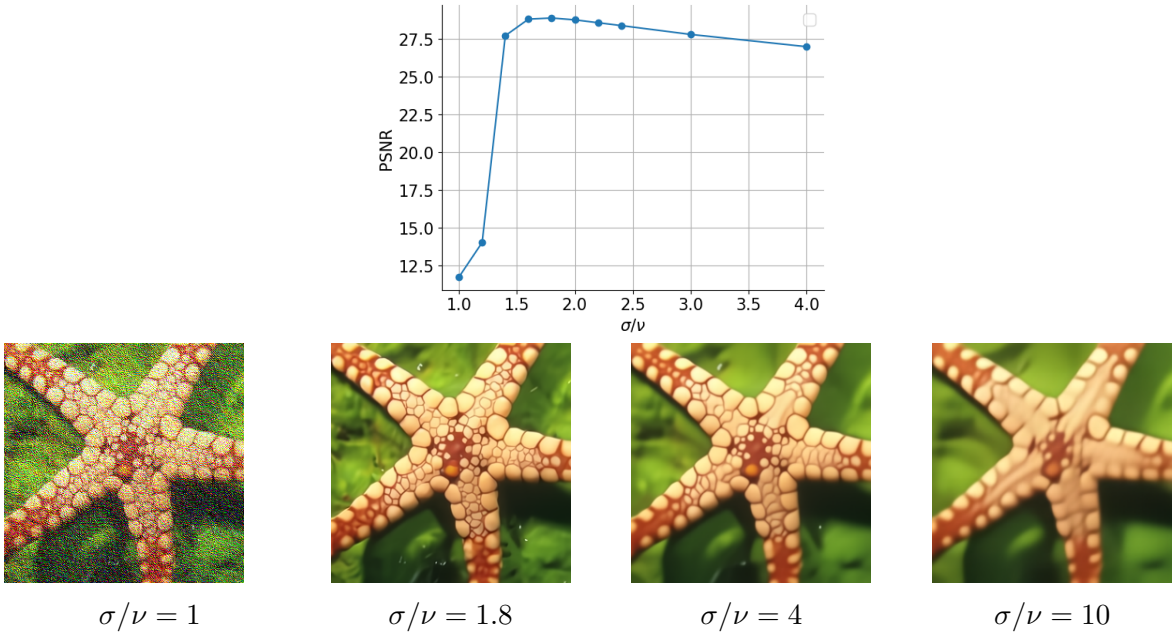


Figure 4.10: Influence of the choice of the parameter σ for deblurring. Top: average PSNR when deblurring the images of CBSD10, blurred with the 10 kernels, for different values of σ, with $\nu = 0.03$. The other parameters remain unchanged. Bottom: visual results when deblurring "starfish" with various σ, with $\nu = 0.03$ (in the same conditions as Figure 4.2).

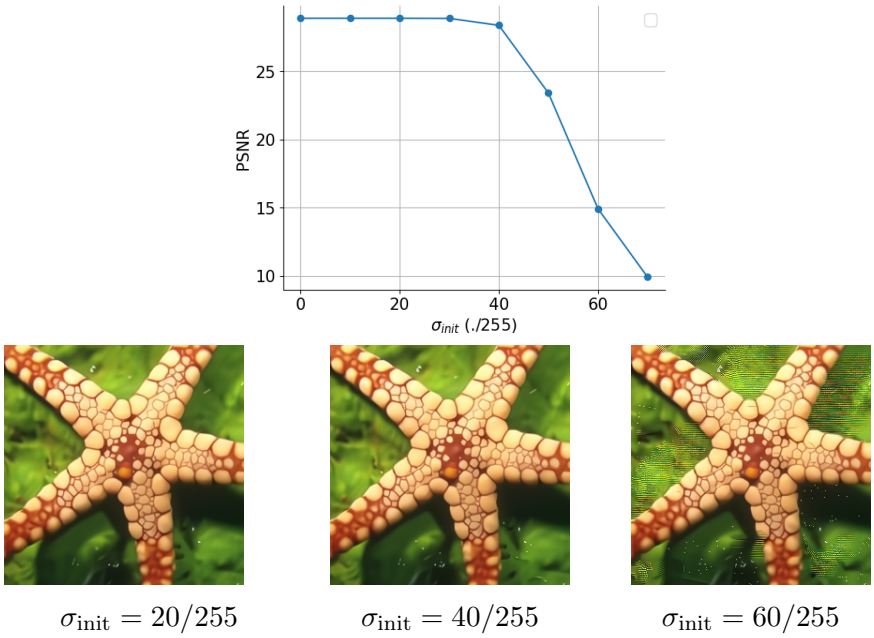


Figure 4.11: Influence of the initialization z_0 on the deblurring result. Instead of initializing with the blurred image $z_0 = y$, as done in Section 4.3.2, we set $z_0 = y + \xi_{\sigma_{\text{init}}}$ with $\xi_{\sigma_{\text{init}}}$ an AWGN with standard deviation σ_{init}. By increasing the noise level σ_{init}, we investigate the robustness of the result to changes in the initialization of the algorithm. Top: PSNR values, along with values of σ_{init}. Bottom: corresponding visual results on "starfish" with various σ_{init} (in the same conditions as Figure 4.2). The algorithm is robust to noisy initialization up to a relatively large value of σ_{init}.

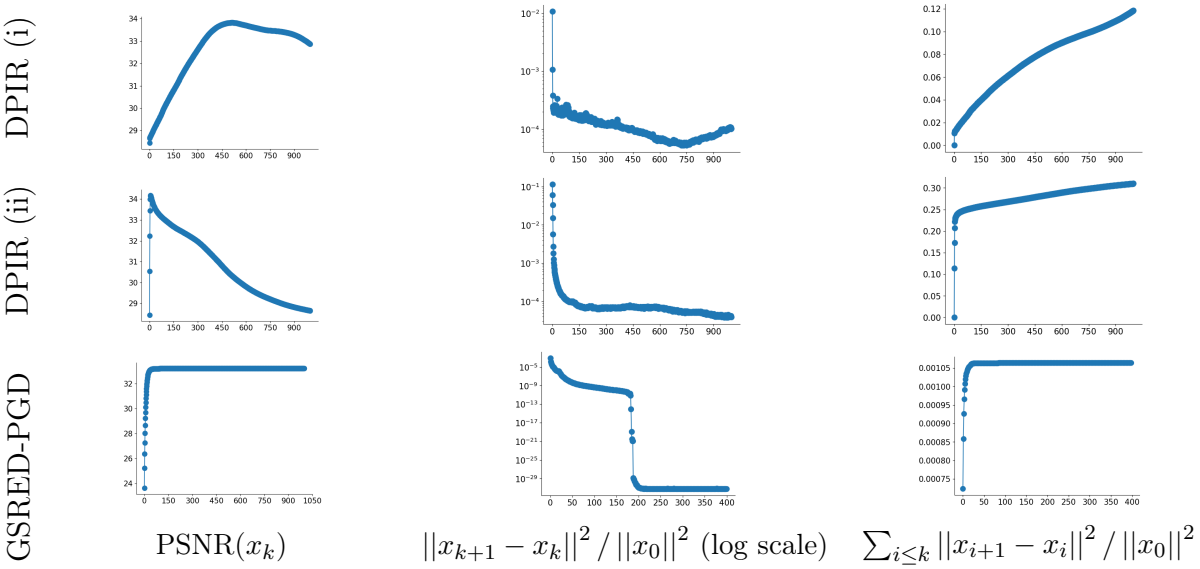


Figure 4.12: Divergence of the DPIR algorithm versus convergence of GSRED-PGD when deblurring the "starfish" image. The two first rows display results obtained with DPIR with two different strategies used for decreasing σ: in the first row, σ is decreased from 49 to ν over 1000 iterations; in the second row, σ is decreased in 8 iterations and then kept fixed for the remaining iterations.

4.4 Conclusion

In this chapter, we introduce a new denoiser for convergence of RED algorithms. The denoiser is trained to realize an exact gradient descent step on a regularization function that is formulated through a neural network. Once plugged in RED algorithms, the iterative schemes are proved to converge towards a stationary point of an explicit functional. One strength of this approach is to simultaneously allow for a denoiser that is not nonexpansive and a non strongly convex (and non-smooth) data-fidelity term. Experiments conducted on ill-posed imaging problems (deblurring, super-resolution, inpainting) confirm the convergence results and show that the proposed algorithms reach state-of-the-art image restoration performance.

A similar denoiser was recently proposed by Cohen et al. (2021). However, they choose to directly parameterize g_σ by a scalar-valued deep network. In practice, this network is built by adding a global average pooling to the output of the DnCNN architecture (Zhang et al., 2017a). Their denoiser is then plugged in the RED-GD algorithm. Their restoration algorithm is thus the same as GSRED-GD (4.24) but with a different regularizer g_σ. The authors do not show denoising performance but argue that the resulting algorithm outperforms RED and PnP with DnCNN denoiser for deblurring and super-resolution. We tried to apply the same average pooling strategy on the DRUNet architecture, but we did not reach the same performance as GSDRUNet.

While the Gradient-Step Denoiser proves beneficial in achieving convergence for RED, it does not ensure convergence for the PnP algorithms (presented in Section 2.4.2) when used in its generic form. In the upcoming chapter, we will introduce a constraint on the denoiser to address the issue of PnP convergence.

Chapter 5

Proximal Denoiser for Convergent PnP algorithms

Contents

In this chapter, we shed a new light on the Gradient-Step denoiser proposed in Chapter 4:

$$D_\sigma = \mathrm{Id} - \nabla g_\sigma \tag{5.1}$$

where $g_\sigma : \mathbb{R}^n \to \mathbb{R}$ is a scalar function parameterized by a differentiable neural network. We have seen in Chapter 4 that, if this denoiser is used in RED algorithms, we can ensure convergence of the iterates towards a stationary point of an explicit functional. However, this strategy does not extend to the PnP algorithms presented in Section 2.4.2. Indeed, while RED methods were built using a gradient descent step by a denoiser, PnP methods were built by replacing a proximal operation by a denoiser.

It is therefore interesting to investigate under which conditions the Gradient-Step denoiser (5.1) can actually be exactly a proximal map. We have seen with Moreau's Theorem 2 that a differentiable operator is a proximity operator of some *convex* potential if and only if it is nonexpansive and the gradient of a convex function. We described Section 3.3.2 how Sreehari et al. (2016) make use of this property to reach convergence of PnP algorithms assuming that the plugged denoiser has symmetric and doubly stochastic Jacobian. The Gradient-Step denoiser (5.1) has by construction symmetric Jacobian

(because it writes as a gradient) and double stochasticity of the Jacobian is equivalent to assume that the operators D_σ and $\mathrm{Id} - D_\sigma$ are both nonexpansive. However, we explained in Section 3.3.1 that nonexpansivity is not a satisfactory assumption for a denoiser.

In this chapter, we rather exploit Theorem 3 *i.e.* the extension of Moreau's theorem proposed in (Gribonval and Nikolova, 2020). It characterizes proximity operators of nonconvex potentials as gradient of convex functions, thus precisely avoiding the nonexpansivity requirement of the denoiser. In Section 5.1, we prove that if $\mathrm{Id} - D_\sigma$ is contractive, then Gradient-Step denoiser (5.1) is the proximity operator of a smooth and weakly convex explicit potential. As described in Section 5.3.1, this condition is softly enforced while training, with little consequence on the denoising efficacy.

With this denoiser, which we refer to "Proximal Gradient-Step denoiser", the PnP algorithms presented in Section 2.4.2 take the form of precise first-order optimization algorithm. Leveraging the nonconvex convergence analysis of these algorithms realized in Section 2.3, we provide in Section 5.2 new proofs of convergence of the iterates of PnP-PGD (2.109) and PnP-DRS (2.111) towards stationary points of an explicit functional.

We will see in Section 5.2.1 that a limitation of this approach is that the proximal denoiser prevents from controlling the stepsize of the overall PnP algorithm, which has to be fixed to 1. Consequently, the stepsize condition for the convergence of the PnP-PGD algorithm will translate to a restrictive condition on the regularization parameter λ. In Section 5.2.3, we propose a relaxation of the PGD algorithm, called αPGD, such that when used with the Proximal Gradient-Step denoiser, the corresponding PnP scheme can converge for any regularization parameter $\lambda > 0$.

In Section 5.3, we provide experiments for both image deblurring and image super-resolution applications. We demonstrate the convergence and the effectiveness of our PnP algorithms with Proximal Gradient-Step denoiser.

Most of the results presented in this Chapter were published in (Hurault et al., 2022b) and (Hurault et al., 2023a). The code implementing the proposed framework is available at `https://github.com/samuro95/ProxPnP`.

5.1 Proximal Gradient-Step denoiser

5.1.1 The Gradient-Step denoiser as a proximity operator

The Gradient Step (GS) Denoiser (5.1) writes

$$D_\sigma = \mathrm{Id} - \nabla g_\sigma = \nabla h_\sigma, \tag{5.2}$$

with a potential

$$h_\sigma : x \to \frac{1}{2}\left|\left|x\right|\right|^2 - g_\sigma(x), \tag{5.3}$$

obtained from a scalar function $g_\sigma : \mathbb{R}^n \to \mathbb{R}$ parameterized by a differentiable neural network. This denoiser D_σ is trained to denoise images degraded with Gaussian noise of standard deviation σ. In Chapter 4, it is shown that, although constrained to be an exact conservative field (5.2), it can realize state-of-the-art denoising.

In this chapter, we make use of the recent characterization of nonconvex proximity operators (Gribonval and Nikolova, 2020) presented in Section 2.2.2, to show that this denoiser can be written as a proximal operator. Applying Theorem 3, we have that if h_σ is convex, the GS denoiser $D_\sigma = \nabla h_\sigma$ is linked to the proximity operator of some function $\phi_\sigma : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$. The next proposition shows that, if h_σ is strongly convex (which

is true as soon as the residual $\mathrm{Id} - D_\sigma$ is contractive), there exists a smooth and weakly convex ϕ_σ such that $D_\sigma = \mathrm{Prox}_{\phi_\sigma}$ is single-valued.

Proposition 25. *Let $g_\sigma : \mathbb{R}^n \to \mathbb{R}$ a $\mathcal{C}^{k+1}$ function with $k \geq 1$ and ∇g_σ L-Lipschitz with $L < 1$. Let $D_\sigma := \mathrm{Id} - \nabla g_\sigma = \nabla h_\sigma$. Then*

(i) there is $\phi_\sigma : \mathbb{R}^n \to \mathbb{R}^n \cup \{+\infty\}$, $\frac{L}{L+1}$-weakly convex, such that $\mathrm{Prox}_{\phi_\sigma}$ is one-to-one and

$$D_\sigma = \mathrm{Prox}_{\phi_\sigma} \tag{5.4}$$

Moreover D_σ is injective, $\mathrm{Im}(D_\sigma)$ is open and there is a constant $K \in \mathbb{R}$ such that ϕ_σ is defined on $\mathrm{Im}(D_\sigma)$ by

$$\begin{aligned} \forall x \in \mathrm{Im}(D_\sigma), \quad \phi_\sigma(x) &= g_\sigma(D_\sigma^{-1}(x)) - \frac{1}{2}\left|\left|D_\sigma^{-1}(x) - x\right|\right|^2 + K \\ &= h_\sigma^*(x) - \frac{1}{2}||x||^2 + K \end{aligned} \tag{5.5}$$

where h_σ^* stands for the convex conjugate of h_σ (defined Section 2.41).

(ii) $\forall x \in \mathbb{R}^n$, $\phi_\sigma(x) \geq g_\sigma(x) + K$ and for $x \in \mathrm{Fix}(D_\sigma)$, $\phi_\sigma(x) = g_\sigma(x) + K$.

(iii) ϕ_σ is $\mathcal{C}^k$ on $\mathrm{Im}(D_\sigma)$ and $\forall x \in \mathrm{Im}(D_\sigma)$, $\nabla\phi_\sigma(x) = {D_\sigma}^{-1}(x) - x = \nabla g_\sigma({D_\sigma}^{-1}(x))$. Moreover, $\nabla\phi_\sigma$ is $\frac{L}{1-L}$-Lipschitz on $\mathrm{Im}(D_\sigma)$

Remark 17. *(i) Note that, despite ϕ_σ being possibly nonconvex, $D_\sigma = \mathrm{Prox}_{\phi_\sigma}$ is one-to-one. Also note that D_σ is possibly not nonexpansive.*
(ii) We could also assume weaker conditions than $\mathrm{Id} - D_\sigma$ contractive. For instance, only assuming h_σ convex, we get the existence of ϕ_σ s.t. $D_\sigma(x) \in \mathrm{Prox}_{\phi_\sigma}(x)$ $\forall x$ but ϕ_σ looses its weak convexity and Lipschitz-gradient properties.

Proof. **(i)** D_σ being $L+1$-Lipschitz, with Theorem 3, there is $\phi_\sigma : \mathbb{R}^n \to \mathbb{R}^n \cup \{+\infty\}$ such that $\forall x \in \mathbb{R}^n$, $D_\sigma(x) \in \mathrm{Prox}_{\phi_\sigma}(x)$ and ϕ_σ $\left(1 - \frac{1}{L+1}\right) = \frac{L}{L+1}$ weakly convex. The weak convexity constant being smaller than 1, $x \to \phi_\sigma(x) + \frac{1}{2}||x-y||^2$ is convex and $\mathrm{Prox}_{\phi_\sigma}$ is one-to-one. According to (Gribonval and Nikolova, 2020, Theorem 4 (b)), there exists a constant $K \in \mathbb{R}$ such that for any $C \subseteq Im(D_\sigma)$ polygonally connected,

$$\forall y \in C, \quad \phi_\sigma(D_\sigma(y)) = \langle y, D_\sigma(y)\rangle - \frac{1}{2}||D_\sigma(y)||^2 - h_\sigma(y) + K \tag{5.6}$$

$$= g_\sigma(y) - \frac{1}{2}||y - D_\sigma(y)||^2 + K \tag{5.7}$$

Moreover, from Proposition 5,

$$\langle y, D_\sigma(y)\rangle = \langle y, \nabla h_\sigma(y)\rangle = h_\sigma(y) + h_\sigma^*(D_\sigma(y)). \tag{5.8}$$

Combining (5.6) and (5.8), we get

$$\phi_\sigma(D_\sigma(y)) = h_\sigma^*(D_\sigma(y)) - \frac{1}{2}||D_\sigma(y)||^2 + K \tag{5.9}$$

As ∇g_σ is $\mathcal{C}^2$ and $L < 1$-Lipschitz, $\forall x \in \mathbb{R}^n$, $J_{D_\sigma}(x) = \mathrm{Id} - \nabla^2 g_\sigma(x)$ is positive definite and D_σ is injective. We can consider its inverse D_σ^{-1} on $\mathrm{Im}(D_\sigma)$, and we get (5.5) from (5.7) Also, the inverse function theorem ensures that $\mathrm{Im}(D_\sigma)$ is open in $\mathbb{R}^n$. As D_σ is continuous,

$\mathrm{Im}(D_\sigma)$ is connected and open, thus polygonally connected. Therefore, (5.6) is true on the whole $\mathrm{Im}(D_\sigma)$.

(ii) We have

$$\begin{aligned}\phi_\sigma(x) &= \frac{1}{2}\left|\left|x - x\right|\right|^2 + \phi_\sigma(x) \\ &\geq \frac{1}{2}\left|\left|x - D_\sigma(x)\right|\right|^2 + \phi_\sigma(D_\sigma(x)) \\ &= g_\sigma(x) + K\end{aligned} \tag{5.10}$$

where the first inequality comes from the definition of the proximal operator $D_\sigma = \mathrm{Prox}_{\phi_\sigma}$ and the last equality is given by (5.5).

(iii) This is the application of (Gribonval and Nikolova, 2020, Corollary 6). For the Lipschitz property, let $x, y \in \mathrm{Im}(D_\sigma)$, there exists $u, v \in \mathbb{R}^n$ such that $x = D_\sigma(u)$ and $y = D_\sigma(v)$. Hence, we have

$$\begin{aligned}\left|\left|\nabla\phi_\sigma(x) - \nabla\phi_\sigma(y)\right|\right| &= \left|\left|{D_\sigma}^{-1}(x) - {D_\sigma}^{-1}(y) - (x - y)\right|\right| \\ &= \left|\left|u - D_\sigma(u) - (v - D_\sigma(v))\right|\right| \\ &= \left|\left|\nabla g_\sigma(u) - \nabla g_\sigma(v)\right|\right| \\ &\leq L\left|\left|u - v\right|\right|\end{aligned} \tag{5.11}$$

because g_σ is L-Lipschitz. Moreover, as $J_{D_\sigma}(x) = \nabla^2 h_\sigma(x) = \mathrm{Id} - \nabla^2 g_\sigma(x)$, $\forall u \in \mathbb{R}^n$, $\langle \nabla^2 h_\sigma(x)u, u\rangle = \left|\left|u\right|\right|^2 - \langle \nabla^2 g_\sigma(x)u, u\rangle \geq (1 - L)\left|\left|u\right|\right|^2$ and h_σ is $(1 - L)$-strongly convex. Thus continuing from last inequalities

$$\begin{aligned}\left|\left|\nabla\phi_\sigma(x) - \nabla\phi_\sigma(y)\right|\right| &\leq \frac{L}{1 - L}\left|\left|\nabla h_\sigma(u) - \nabla h_\sigma(v)\right|\right| \\ &= \frac{L}{1 - L}\left|\left|D_\sigma(u) - D_\sigma(v)\right|\right| \\ &= \frac{L}{1 - L}\left|\left|x - y\right|\right|.\end{aligned} \tag{5.12}$$

□

5.1.2 More details on the regularization ϕ_σ

Recall that in Chapter 4, the potential g_σ was parameterized as

$$g_\sigma(x) = \frac{1}{2}\left|\left|x - N_\sigma(x)\right|\right|^2, \tag{5.13}$$

where $N_\sigma : \mathbb{R}^n \to \mathbb{R}^n$ is a $\mathcal{C}^1$ neural network with ELU activations. In order to apply Proposition 25, we need g_σ to be at least $\mathcal{C}^2$. We choose the same parameterization, but we change ELU activations to softplus activations

$$s(x) = \frac{1}{\alpha}\log(1 + e^{\alpha x}) \tag{5.14}$$

which are $\mathcal{C}^\infty$. As seen in Section 4.1.2, as the softplus activation has Lipschitz gradient, the gradient of g_σ (or more precisely of its extended version g_σ^R (4.20)) is guaranteed to be L-Lipschitz for some $L > 0$. However, we are not guaranteed to have $L < 1$ by construction. As described in the experimental Section 5.3.1, in practice, we will softly enforce this condition during training. Assuming that $L < 1$, the potential ϕ_σ obtained with Proposition 25 then verifies:

(i) ϕ_σ is **lower-bounded**. As g_σ is non-negative, this follows directly from Proposition 25(ii).

(ii) ϕ_σ is $\frac{L}{L+1}$-**weakly convex** from Proposition 25(i)

(iii) ϕ_σ is $\mathcal{C}^1$ and $\nabla\phi_\sigma$ is $\frac{L}{1-1}$-**Lipschitz continuous** on $\mathrm{Im}(D_\sigma)$ from Proposition 25(iii)

(iv) ϕ_σ is **coercive** when g_σ is coercive. This follows directly from $\phi_\sigma(x) \geq g_\sigma(x)$ (Proposition 25(ii)).

(v) ϕ_σ **is real analytic and thus subanalytic on** $\mathrm{Im}(D_\sigma)$ (see Section 3.1.2 for more details on subanalytic functions). g_σ is parameterized with the softplus activation, which is a real analytic function. As explained in details in Section 3.1.2, by composition and sum of real analytic functions (Lemma 3(i)) N_σ and then subsequently g_σ are real analytic functions. Then, by Lemma 3(ii), ∇g_σ and thus D_σ are also real analytic. By the inverse function theorem (Lemma 3(iii)), as $\forall x \in \mathbb{R}^n, J_{D_\sigma}(x) > 0$, D_σ^{-1} is then real analytic on $\mathrm{Im}(D_\sigma)$. We finally obtain, using the expression of ϕ_σ on $\mathrm{Im}(D_\sigma)$ (5.5), again by sum and composition, that ϕ_σ is real analytic (and thus subanalytic) on $\mathrm{Im}(D_\sigma)$.

5.1.3 Relaxed Proximal Denoiser

Once trained, the Gradient-Step denoiser $D_\sigma = \mathrm{Id} - \nabla g_\sigma$ can be relaxed with a parameter $\gamma \in [0, 1]$

$$D_\sigma^\gamma = \gamma D_\sigma + (1-\gamma)\,\mathrm{Id} = \mathrm{Id} - \gamma \nabla g_\sigma. \tag{5.15}$$

Applying Proposition 25 with $g_\sigma^\gamma = \gamma g_\sigma$ which has a γL-Lipschitz gradient, we get that if $\gamma L < 1$, there exists a $\frac{\gamma L}{\gamma L+1}$-weakly convex $\phi_\sigma^\gamma : \mathbb{R}^n \to \mathbb{R}^n \cup \{+\infty\}$ such that

$$D_\sigma^\gamma = \mathrm{Prox}_{\phi_\sigma^\gamma}, \tag{5.16}$$

Hence, one can control the weak convexity of the regularization function by relaxing the proximal denoising operator D_σ^γ. However, for high values of γ, the relaxed denoiser tends to lose its denoising capacities.

5.2 PnP convergence analysis with Proximal Denoiser

PnP algorithms presented in Section 2.4.2 were built by replacing one proximal operation with a denoiser D_σ in proximal optimization algorithms. When used with the Proximal denoiser $D_\sigma = \mathrm{Prox}_{\phi_\sigma}$ that corresponds to the proximal operator of a nonconvex regularization function ϕ_σ, PnP algorithms become again real proximal optimization algorithms, with the noticeable particularity that a nonconvex function is involved. Moreover, as the denoiser writes $D_\sigma = \mathrm{Prox}_{\phi_\sigma}$ and not $D_\sigma = \mathrm{Prox}_{\tau\phi_\sigma}$ with a stepsize $\tau > 1$, we will need to fix the stepsize to $\tau = 1$. Indeed, Prox_ϕ does not give permit to calculate $\mathrm{Prox}_{\tau\phi}$ explicitly when $\tau \neq 1$. Therefore, to be a provable converging proximal splitting algorithm, the stepsize of the overall PnP algorithm has to be fixed to $\tau = 1$.

In this section we study the convergence of PnP-PGD, PnP-DRS and PnP-ADMM introduced in Section 2.4.2 with proximal denoiser. For that purpose, we target the objective function

$$F := \lambda f + \phi_\sigma. \tag{5.17}$$

where f is a (possibly nonconvex) data-fidelity term, λ is a regularization parameter and ϕ_σ is defined as in (5.5) from the function g_σ defined in (5.13).

To obtain the following convergence results, we make use of the convergence analysis of the standard PGD, DRS or ADMM algorithms in the nonconvex setting already developed in Section 3.2. Moreover, in Section 5.2.3, we propose a new version of the PnP-PGD algorithm that convergences for a wider range of regularization parameters.

5.2.1 Proximal PnP-PGD (ProxPnP-PGD)

We first study the convergence of the PnP-PGD algorithm with plugged Proximal Gradient-Step Denoiser (5.4):

$$x_{k+1} = D_\sigma(\mathrm{Id} - \lambda\nabla f)(x_k) = \mathrm{Prox}_{\phi_\sigma}(\mathrm{Id} - \lambda\nabla f)(x_k). \tag{5.18}$$

It corresponds to the PGD algorithm (2.48) for minimizing $\lambda f + \phi_\sigma$ with fixed stepsize $\tau = 1$.

At each iteration, $x_k \in \mathrm{Im}(D_\sigma)$, and thus ϕ_σ defined in (5.5) is tractable along the algorithm. The value of the objective function (5.17) at x_k is

$$F(x_k) = \lambda f(x_k) + g_\sigma(z_k) - \frac{1}{2}\,||z_k - x_k||^2 + K \tag{5.19}$$

where $z_k = (\mathrm{Id} - \lambda\nabla f)(x_{k-1}) = D_\sigma^{-1}(x_k)$. We assume here the data-fidelity term f differentiable with Lipschitz gradient, but not necessarily convex. As ϕ_σ is not only nonconvex but $\frac{L}{L+1}$-weakly convex (from Proposition 25(i)), we can use the Theorem 8 on the convergence of the PGD algorithm for nonconvex - weakly convex optimization.

Theorem 19 (Convergence of ProxPnP-PGD)**.** *Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be differentiable with L_f-Lipschitz gradient, bounded from below. Assume that L the Lipschitz constant of ∇g_σ verifies $L < 1$. Then, for $\lambda L_f < \frac{L+2}{L+1}$, the iterates (x_k) given by the iterative scheme* (5.18) *verify*

(i) $(F(x_k))$ is non-increasing and converges.

(ii) The sequence has finite length i.e. $\sum_{k=0}^{+\infty} ||x_{k+1} - x_k||^2 < +\infty$ *and* $||x_{k+1} - x_k||$ *converges to* 0 *at rate* $\min_{k<K} ||x_{k+1} - x_k|| = \mathcal{O}(1/\sqrt{K})$

(iii) If f is non-negative and subanalytic, and g_σ is coercive, then the iterates (x_k) converge towards a critical point of F.

Proof. This is a direct application of Theorem 8 with λf the smooth function and with ϕ_σ the weakly convex function. The stepsize condition

$$\tau < \max\left(\frac{2}{\lambda L_f + \frac{L}{L+1}}, \frac{1}{\lambda L_f}\right) \tag{5.20}$$

becomes, with $\tau = 1$,

$$\Leftrightarrow \lambda L_f + \frac{L}{L+1} < 2 \text{ or } \lambda L_f < 1 \tag{5.21}$$

$$\Leftrightarrow \lambda L_f < \frac{L+2}{L+1} \tag{5.22}$$

For (iii), as shown in Section 5.1.2, ϕ_σ is subanalytic on $\mathrm{Im}(D_\sigma)$, lower-bounded and coercive if g_σ is coercive. As f is assumed non-negative and subanalytic, by Lemma 5, $F = \lambda f + \phi_\sigma$ is then subanalytic (up to adding a constant to F to make ϕ_σ non-negative) and thus KŁ on $\mathrm{Im}(D_\sigma)$. This is enough for the proof of Theorem 8 to remain true because any limit point of (x_k) is a fixed point of the PnP-PGD operator and thus belongs to $\mathrm{Im}(D_\sigma)$. □

5.2.2 Proximal PnP-DRS and PnP-ADMM

Following the same strategy as for PnP-PGD, we now prove the convergence of the PnP-DRS algorithm (2.111) with plugged Proximal Denoiser $D_\sigma = \text{Prox}_{\phi_\sigma}$ and stepsize $\tau = 1$. The resulting algorithm, referred to *ProxPnP-DRS*, writes

$$\begin{cases} y_{k+1} \in \text{Prox}_{\lambda f}(x_k) & \text{(5.23a)} \\ z_{k+1} = D_\sigma(2y_{k+1} - x_k) = \text{Prox}_{\phi_\sigma}(2y_{k+1} - x_k) & \text{(5.23b)} \\ x_{k+1} = x_k + 2\beta(z_{k+1} - y_{k+1}) & \text{(5.23c)} \end{cases}$$

or equivalently

$$x_{k+1} \in \left(\beta\, \text{Rprox}_{\phi_\sigma} \circ \text{Rprox}_f + (1-\beta)\, \text{Id}\right)(x_k). \tag{5.24}$$

As shown in Lemma 1, for $\beta = \frac{1}{2}$, this scheme is equivalent to the ProxPnP-ADMM algorithm *i.e.* PnP-ADMM with Proximal Denoiser:

$$\begin{cases} z_{k+1} = D_\sigma(y_k - u_k) & \text{(5.25a)} \\ y_{k+1} \in \text{Prox}_{\lambda f}(u_k + z_{k+1}) & \text{(5.25b)} \\ u_{k+1} = u_k + z_{k+1} - y_{k+1}. & \text{(5.25c)} \end{cases}$$

The convergence of (5.25) will then directly follow from the convergence of (5.23).

Li and Pong (2016) and Themelis and Patrinos (2020) propose convergence proofs of the DRS algorithm for the minimization of the sum of two nonconvex functions, one of the two functions being differentiable with Lipschitz gradient. Their convergence results were summarized in Theorem 10. We will adapt their results to obtain convergence results of our ProxPnP-DRS algorithm. Depending on the differentiability of f, different convergence results can be derived.

Differentiable data-fidelity term

We first consider f differentiable with Lipschitz gradient. In this case, we can keep λf as the differentiable function in Theorem 10, and ϕ_σ as the other function (called g in Theorem 10). For $\tau = 1$, the Douglas-Rachford Envelope (3.54) then writes

$$F^{DR}(x, y, z) = \lambda f(y) + \phi_\sigma(z) + \langle y - x, y - z\rangle + \frac{1}{2}\,||y - z||^2 \tag{5.26}$$

that is to say, along the iterates

$$\begin{aligned} F^{DR}_{k+1} &= F^{DR}(x_k, y_{k+1}, z_{k+1}) = \lambda f(y_{k+1}) + g_\sigma(2y_{k+1} - x_k) \\ &\quad - \frac{1}{2}\,||z_{k+1} - (2y_{k+1} - x_k)||^2 + \langle y_{k+1} - x_k, y_{k+1} - z_{k+1}\rangle + \frac{1}{2}\,||y_{k+1} - z_{k+1}||^2 . \end{aligned} \tag{5.27}$$

Using Theorem 10, we prove convergence of ProxPnP-DRS (and thus ProxPnP-ADMM) towards stationary points of $F = \lambda f + \phi_\sigma$.

Theorem 20 (Convergence of ProxPnP-DRS with f differentiable)**.** *Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be differentiable with L_f-Lipschitz gradient, bounded from below. Assume that L the Lipschitz constant of ∇g_σ verifies $L < 1$. Then, for $\lambda L_f < 1$, and $\beta \in (0, 1)$, the iterates (y_k, z_k, x_k) given by the iterative scheme (5.23) verify*

(i) $(F^{DR}(x_{k-1}, y_k, z_k))$ is non-increasing and converges.

(ii) $x_k - x_{k-1} = \beta(y_k - z_k)$ *vanishes with rate* $\min_{k \leq K} ||y_k - z_k|| = \mathcal{O}(\frac{1}{\sqrt{K}})$.

(iii) If f *is non-negative and subanalytic, and* g_σ *is coercive, then the sequences* (y_k) *and* (z_k) *converge to the same critical point of* F.

Proof. Point (i) and (ii) directly follow from Theorem 10. For point (iii), Theorem 10 requires the iterates to remain bounded and F^{DR} to verify the KŁ property. First, as noticed in Section 5.1, if g_σ is coercive, ϕ_σ is coercive too and by (Li and Pong, 2016, Theorem 4) the iterates (x_k, y_k, z_k) remain bounded. Second, as also shown in Section 5.1, ϕ_σ is subanalytic on $\mathrm{Im}(D_\sigma)$ and non-negative. As f is assumed non-negative and lower-bounded, by Lemma 5, (up to adding a constant to make ϕ_σ non-negative) F^{DR} is then subanalytic and thus KŁ on $\mathbb{R}^n \times Im(D_\sigma) \times \mathbb{R}^n$. The proof of point (iii) in (Li and Pong, 2016, Theorem 2) uses the fact that F^{DR} is KŁ on any limit point (y^*, z^*, x^*) of the algorithm. We thus need to verify that $z^* \in Im(D_\sigma)$ for the proof to remain valid. This is indeed the case because, at the limit, we have $z^* = D_\sigma(2y^* - x^*)$. □

With this result, due to the fact that we are limited to $\tau = 1$, the stepsize condition for DRS convergence Theorem 9 becomes a restriction on the regularization parameter $\lambda L_f < 1$. In the following section, we use the fact that ϕ_σ is weakly convex to avoid this restriction.

Non-differentiable data-fidelity term

To cope with a possibly non-differentiable data-fidelity term, and to get rid of the restriction on the trade-off parameter λ, we inverse the denoising and proximal steps in (5.23):

$$\begin{cases} y_{k+1} = D_\sigma(x_k) = \mathrm{Prox}_{\phi_\sigma}(x_k) & \text{(5.28a)} \\ z_{k+1} = \mathrm{Prox}_{\lambda f}(2y_{k+1} - x_k) & \text{(5.28b)} \\ x_{k+1} = x_k + 2\beta(z_{k+1} - y_{k+1}) & \text{(5.28c)} \end{cases}$$

and adapt the Douglas-Rachford Envelope accordingly

$$F^{DR,2}(x, y, z) = \lambda f(z) + \phi_\sigma(y) + \langle y - x, y - z \rangle + \frac{1}{2} ||y - z||^2 \tag{5.29}$$

that is to say, along the iterates

$$\begin{aligned} F^{DR,2}_{k+1} &= F^{DR,2}(x_k, y_{k+1}, z_{k+1}) = \lambda f(z_{k+1}) + g_\sigma(x_k) \\ &\quad - \frac{1}{2} ||y_{k+1} - x_k)||^2 + \langle y_{k+1} - x_k, y_{k+1} - z_{k+1} \rangle + \frac{1}{2} ||y_{k+1} - z_{k+1}||^2 . \end{aligned} \tag{5.30}$$

Convergence of this algorithm would be ensured with Theorem 9 for ϕ_σ with Lipschitz gradient on $\mathbb{R}^n$. However, Proposition 25 shows that ϕ_σ is differentiable only on $\mathrm{Im}(D_\sigma)$. In our original publication (Hurault et al., 2022b), we assumed $L < 1/2$ and $\mathrm{Im}(D_\sigma)$ convex in order to keep the descent lemma for ϕ_σ valid on $\mathrm{Im}(D_\sigma)$. Under this assumption, we showed that we still have convergence of the iterates of (5.28). However, the convexity of $\mathrm{Im}(D_\sigma)$ is difficult to verify and unlikely to be true in practice.

Instead, we come back to and adapt the original and general Theorem 10 from (Themelis and Patrinos, 2020; Li and Pong, 2016) on the convergence of the DRS algorithm with non-convex functions. The main adaptation concerns Theorem 10, point (i) which follows from the sufficient decrease property of the Douglas-Rachford envelope F^{DR} shown in (Themelis and Patrinos, 2020, Theorem 1). We directly write the result in our particular case of interest, with the following theorem.

Theorem 21 (Convergence of ProxPnP-DRS with f non-differentiable)**.** *Let f proper, lsc, bounded from below. Assume that L the Lipschitz constant of ∇g_σ and $\beta \in (0,1)$ verify*

$$\beta(2L^3 - 3L^2 + 1) + (2L^2 + L - 1) < 0. \tag{5.31}$$

Then, for all $\lambda > 0$, the iterates (y_k, z_k, x_k) given by the iterative scheme (5.28) verify

(i) $(F^{DR,2}(x_{k-1}, y_k, z_k))$ *is non-increasing and converges.*

(ii) $x_k - x_{k-1} = \beta(y_k - z_k)$ *vanishes with rate* $\min_{k \leq K} ||y_k - z_k|| = \mathcal{O}(\frac{1}{\sqrt{K}})$.

(iii) If f is non-negative and subanalytic, and g_σ is coercive, then the sequences (y_k) and (z_k) converge to the same critical point of F.

Remark 18. *The standard Douglas-Rachford algorithm is for $\beta = 1/2$. The restriction on L is then*

$$2L^3 + L^2 + 2L - 1 < 0 \tag{5.32}$$

which is verified for approximately $L < 0.376$. Using a smaller β relaxes the constraint on L until $L < 0.5$ when $\beta \to 0$. For example for $\beta = 0.25$, the constraint is verified up to approximately $L < 0.45$.

Remark 19. *In addition to deal with non-differentiable data-fidelity terms, the advantage of this result, compared to Theorem 20, is that the restriction on λ disappears, and the convergence remains true for all regularization parameter λ. However, the restriction on L the Lipschitz constant of ∇g_σ is stronger, which should harm the denoising capacity of the denoiser.*

Proof. **(i)** As explained above, this point represents the main adaptation. Following the initial proof from (Themelis and Patrinos, 2020, Theorem 1), we derive a new sufficient decrease property for $F^{DR,2}$.

We first rewrite the $F^{DR,2}$ from (5.29) as

$$F^{DR,2}(x,y,z) = \lambda f(z) + \phi_\sigma(y) + \langle y - x, y - z\rangle + \frac{1}{2}||y - z||^2 \tag{5.33}$$

$$= \lambda f(z) + \phi_\sigma(y) + \frac{1}{2}||(2y - x) - z||^2 - \frac{1}{2}||x - y||^2. \tag{5.34}$$

As $z_{k+1} = \text{Prox}_{\tau f}(2y_{k+1} - x_k)$ (5.28b), denoting $\forall k > 0$ $F_k^{DR,2} = F^{DR,2}(x_{k-1}, y_k, z_k)$,

$$\begin{aligned} F_{k+1}^{DR,2} &= \min_{z \in \mathbb{R}^n} \lambda f(z) + \phi_\sigma(y_{k+1}) + \frac{1}{2}||(2y_{k+1} - x_k) - z||^2 - \frac{1}{2}||x_k - y_{k+1}||^2 \\ &= \min_{z \in \mathbb{R}^n} \lambda f(z) + \phi_\sigma(y_{k+1}) + \langle y_{k+1} - x_k, y_{k+1} - z\rangle + \frac{1}{2}||y_{k+1} - z||^2. \end{aligned} \tag{5.35}$$

The optimality condition for (5.28a) is

$$x_k - y_{k+1} \in \partial\phi_\sigma(y_{k+1}). \tag{5.36}$$

As ϕ_σ is differentiable on $\text{Im}(D_\sigma)$ and $y_{k+1} \in Im(D_\sigma)$,

$$x_k - y_{k+1} = \nabla\phi_\sigma(y_{k+1}). \tag{5.37}$$

The Douglas–Rachford envelope $F^{DR,2}$ then writes

$$F_{k+1}^{DR,2} = \min_{z\in\mathbb{R}^n} \lambda f(z) + \phi_\sigma(y_{k+1}) + \langle \nabla\phi_\sigma(y_{k+1}), z - y_{k+1}\rangle + \frac{1}{2}\left|\left|y_{k+1} - z\right|\right|^2. \tag{5.38}$$

We get, by comparing with the right term when $z = z_k$,

$$\begin{aligned} F_{k+1}^{DR,2} &\leq \lambda f(z_k) + \phi_\sigma(y_{k+1}) + \langle \nabla\phi_\sigma(y_{k+1}), z_k - y_{k+1}\rangle + \frac{1}{2}\left|\left|y_{k+1} - z_k\right|\right|^2 \\ &= \phi_\sigma(y_{k+1}) + \langle \nabla\phi_\sigma(y_{k+1}), y_k - y_{k+1}\rangle + \langle \nabla\phi_\sigma(y_{k+1}), z_k - y_k\rangle \\ &+ \lambda f(z_k) + \frac{1}{2}\left|\left|y_{k+1} - z_k\right|\right|^2. \end{aligned} \tag{5.39}$$

From Proposition 25, ϕ_σ is $M = \frac{L}{L+1}$ weakly convex. Then with Corollary 1,

$$\phi_\sigma(y_{k+1}) + \langle \nabla\phi_\sigma(y_{k+1}), y_k - y_{k+1}\rangle \leq \phi_\sigma(y_k) + \frac{M}{2}\left|\left|y_k - y_{k+1}\right|\right|^2 \tag{5.40}$$

and

$$F_{k+1}^{DR,2} \leq \phi_\sigma(y_k) + \lambda f(z_k) + \langle \nabla\phi_\sigma(y_{k+1}), z_k - y_k\rangle + \frac{M}{2}\left|\left|y_k - y_{k+1}\right|\right|^2 + \frac{1}{2}\left|\left|y_{k+1} - z_k\right|\right|^2. \tag{5.41}$$

Moreover

$$\langle \nabla\phi_\sigma(y_{k+1}), z_k - y_k\rangle = \langle \nabla\phi_\sigma(y_k), z_k - y_k\rangle + \langle \nabla\phi_\sigma(y_{k+1}) - \nabla\phi_\sigma(y_k), z_k - y_k\rangle \tag{5.42}$$

and using

$$F_k^{DR,2} = \lambda f(z_k) + \phi_\sigma(y_k) + \langle y_k - x_{k-1}, y_k - z_k\rangle + \frac{1}{2}\left|\left|y_k - z_k\right|\right|^2 \tag{5.43}$$

$$= \lambda f(z_k) + \phi_\sigma(y_k) + \langle \nabla\phi_\sigma(y_k), z_k - y_k\rangle + \frac{1}{2}\left|\left|y_k - z_k\right|\right|^2 \tag{5.44}$$

we get

$$F_{k+1}^{DR,2} \leq F_k^{DR,2} + \langle \nabla\phi_\sigma(y_{k+1}) - \nabla\phi_\sigma(y_k), z_k - y_k\rangle - \frac{1}{2}\left|\left|y_k - z_k\right|\right|^2 + \frac{1}{2}\left|\left|y_{k+1} - z_k\right|\right|^2 + \frac{M}{2}\left|\left|y_k - y_{k+1}\right|\right|^2. \tag{5.45}$$

Now using

$$\frac{1}{2}\left|\left|y_{k+1} - z_k\right|\right|^2 = \frac{1}{2}\left|\left|y_k - y_{k+1}\right|\right|^2 + \langle y_{k+1} - y_k, y_k - z_k\rangle + \frac{1}{2}\left|\left|y_k - z_k\right|\right|^2 \tag{5.46}$$

we get

$$F_{k+1}^{DR,2} \leq F_k^{DR,2} + \langle (\nabla\phi_\sigma(y_{k+1}) - \nabla\phi_\sigma(y_k)) - (y_{k+1} - y_k), z_k - y_k\rangle + \frac{1+M}{2}\left|\left|y_k - y_{k+1}\right|\right|^2. \tag{5.47}$$

From (5.28c) and (5.37)

$$z_k - y_k = \frac{1}{2\beta}(x_k - x_{k-1}) = \frac{1}{2\beta}(y_{k+1} - y_k) + \frac{1}{2\beta}(\nabla\phi_\sigma(y_{k+1}) - \nabla\phi_\sigma(y_k)). \tag{5.48}$$

And we obtain

$$F_{k+1}^{DR,2} - F_k^{DR,2} \leq \left(1 + M - \frac{1}{\beta}\right) \frac{1}{2} ||y_k - y_{k+1}||^2 + \frac{1}{2\beta} ||\nabla\phi_\sigma(y_{k+1}) - \nabla\phi_\sigma(y_k)||^2 . \quad (5.49)$$

Using the fact that $\nabla\phi_\sigma$ is $\frac{L}{1-L}$ Lipschitz on $\mathrm{Im}(D_\sigma)$ and that $\forall k > 0, y_k \in Im(D_\sigma)$, it simplifies to

$$F_{k+1}^{DR,2} - F_k^{DR,2} \leq \left(1 + M + \frac{1}{\beta}\left(\frac{L^2}{(1-L)^2} - 1\right)\right) \frac{1}{2} ||y_k - y_{k+1}||^2 . \quad (5.50)$$

The condition on the stepsize for $F_{k+1}^{DR,2} \leq F_k^{DR,2}$ is thus

$$\begin{aligned} & 1 + M + \frac{1}{\beta}\left(\frac{L^2}{(1-L)^2} - 1\right) < 0 \\ \Leftrightarrow\ & \frac{2L+1}{L+1} + \frac{1}{\beta}\frac{2L-1}{(1-L)^2} < 0 \\ \Leftrightarrow\ & \beta(2L^3 - 3L^2 + 1) + (2L^2 + L - 1) < 0. \end{aligned} \quad (5.51)$$

For completeness, recalling that $D_\sigma = \nabla h_\sigma$ with h_σ $1-L$ strongly convex, we have

$$||y_k - y_{k+1}|| = ||D_\sigma(x_k) - D_\sigma(x_{k-1})|| \geq (1-L) ||x_k - x_{k-1}|| \quad (5.52)$$

and the sufficient decrease writes

$$F_k^{DR,2} - F_{k+1}^{DR,2} \geq (1-L)\left(\frac{1}{\beta}\left(1 - \frac{L^2}{(1-L)^2}\right) - (1+M)\right) \frac{1}{2} ||x_k - x_{k-1}||^2 . \quad (5.53)$$

(ii) This point directly follows from the above sufficient decrease condition.

(iii) For the original DRS algorithm, this point was proven in (Li and Pong, 2016, Theorem 2). Given the above sufficient decrease property, assuming that (a) $F^{DR,2}$ verifies the KŁ property on $\mathrm{Im}(D_\sigma)$ and (b) that the iterates are bounded, the proof follows equally. With the same arguments as in the proof of Theorem 20, both assumptions (a) and (b) are verified when f is non-negative and subanalytic, and g_σ is coercive. □

5.2.3 Relaxed Proximal PnP-PGD (ProxPnP-αPGD)

Coming back to the Proximal PnP-PGD convergence result presented in Section 5.2.1, note that in Theorem 19, the usual condition on the stepsize became a condition on the regularization parameter $\lambda L_f < \frac{L+2}{L+1} < 2$. The value of the regularization trade-off parameter λ is then limited. This is an issue when restoring an image with mild degradations for which relevant solutions are obtained with a low amount of regularization and a dominant data-fidelity term in $F = \lambda f + g$.

We could argue that this is a not a problem as the regularization strength is also regulated by the σ parameter, which we are free to tune manually. However, it is observed in practice, for instance for GSRED algorithm in Chapter 4, that the performance of PnP method greatly benefits from the ability to tune the two regularization parameters.

Given the above limitation, our objective is to design a new convergent PnP algorithm with proximal denoiser, and with minimal restriction on the regularization parameter λ. Contrary to previous work on PnP convergence, we not only wish to adapt the denoiser but also the original optimization scheme of interest. In this section, we propose a relaxation of the general Proximal Gradient Descent algorithm, called αPGD, such that when used with the proximal denoiser, the corresponding PnP scheme *ProxPnP-*αPGD can converge for all regularization parameters λ.

αPGD algorithm

We first introduce the αPGD algorithm for solving the general problem

$$\min_{x\in\mathbb{R}^n} f(x) + g(x) \tag{5.54}$$

for $f, g : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ with f smooth and g weakly convex. The algorithms write, for $\alpha \in (0, 1)$,

$$\begin{cases} q_{k+1} = (1-\alpha)y_k + \alpha x_k & \text{(5.55a)} \\ x_{k+1} = \text{Prox}_{\tau g}(x_k - \tau\nabla f(q_{k+1})) & \text{(5.55b)} \\ y_{k+1} = (1-\alpha)y_k + \alpha x_{k+1}. & \text{(5.55c)} \end{cases}$$

Algorithm (5.55) with $\alpha = 1$ exactly corresponds to the PGD algorithm (2.48). This scheme is reminiscent of (Tseng, 2008) (taking $\alpha = \theta_k$ and $\tau = \frac{1}{\theta_k L_f}$ in Algorithm 1 of (Tseng, 2008)), which generalizes Nesterov-like accelerated proximal gradient methods (Beck and Teboulle, 2009a; Nesterov, 2013). Inspired by Lan and Zhou (2018), this scheme was derived from the Primal-Dual algorithm (Chambolle and Pock, 2011) with Bregman proximal operator (Chambolle and Pock, 2016). We now describe this derivation.

Primal-Dual derivation Suppose we target a solution of (5.54) for f L_f-smooth and strongly convex and g convex. As presented in Section 2.3.4, Chambolle and Pock (2016) target a saddle point of the equivalent dual problem proposed with a Bregman Primal-Dual algorithm

$$\begin{cases} y_{k+1} = \arg\min_y \frac{1}{\sigma} D_{h^Y}(y, y_k) + f^*(y) - \langle \bar{x}_k, y\rangle & \text{(5.56a)} \\ x_{k+1} = \arg\min_x \frac{1}{\tau} D_{h^X}(x, x_k) + g(x) + \langle x, y_{k+1}\rangle & \text{(5.56b)} \\ \bar{x}_{k+1} = x_{k+1} + \beta(x_{k+1} - x_k). & \text{(5.56c)} \end{cases}$$

Notice that $L_f f^*$ is 1-strongly convex with respect to the norm $||.||^2$. Indeed, f smooth and convex with a L_f-Lipschitz gradient implies f^* $1/L_f$ strongly-convex, *i.e.* $L_f f^*$ is 1-strongly convex. Then we can use $h^Y = L_f f^*$ (and $h^X = \frac{1}{2}||.||^2$) and the algorithm becomes

$$\begin{cases} y_{k+1} = \arg\min_y \left(\frac{L_f}{\sigma} + 1\right) f^*(y) - \langle \bar{x}_k + \frac{L_f}{\sigma}\nabla f^*(y^k), y\rangle & \text{(5.57a)} \\ x_{k+1} \in \arg\min_x \frac{1}{2\tau}||x - x^k||^2 + g(x) + \langle x, y_{k+1}\rangle & \text{(5.57b)} \\ \bar{x}_{k+1} = x_{k+1} + \beta(x_{k+1} - x_k) & \text{(5.57c)} \end{cases}$$

The optimality condition for the first update is

$$\left(\frac{L_f}{\sigma} + 1\right)\nabla f^*(y_{k+1}) = \bar{x}_k + \frac{L_f}{\sigma}\nabla f^*(y^k). \tag{5.58}$$

Using the fact that $(\nabla f)^{-1} = \nabla f^*$, with the change of variable $y_k \longleftarrow \nabla f^*(y_k)$, the previous algorithm can then be rewritten in a fully primal formulation:

$$\begin{cases} y_{k+1} = \frac{\bar{x}_k + \frac{L_f}{\sigma} y_k}{1 + \frac{L_f}{\sigma}} & \text{(5.59a)} \\ x_{k+1} \in \text{Prox}_{\tau g}(x_k - \tau\nabla f(y_{k+1})) & \text{(5.59b)} \\ \bar{x}_{k+1} = x_{k+1} + \beta(x_{k+1} - x_k). & \text{(5.59c)} \end{cases}$$

For $\alpha = \frac{1}{1+\frac{L_f}{\sigma}}$, $\beta = 1-\alpha$, the algorithm writes

$$\begin{cases} y_{k+1} = \alpha \bar{x}_k + (1-\alpha) y_k & \text{(5.60a)} \\ x_{k+1} \in \operatorname{Prox}_{\tau g}(x_k - \tau \nabla f(y_{k+1})) & \text{(5.60b)} \\ \bar{x}_{k+1} = x_{k+1} + (1-\alpha)(x_{k+1} - x_k). & \text{(5.60c)} \end{cases}$$

or

$$\begin{cases} y_{k+1} = \alpha x_k + (1-\alpha) y_k + \alpha(1-\alpha)(x_k - x_{k-1}) & \text{(5.61a)} \\ x_{k+1} \in \operatorname{Prox}_{\tau g}(x_k - \tau \nabla f(y_{k+1})) & \text{(5.61b)} \end{cases}$$

The latter is equivalent to (5.55) with y_k in place of q_k.

αPGD convergence Although the scheme is derived from the Bregman Primal Dual algorithm, in our case, f is convex and not strongly convex and g is weakly convex and not convex, thus the general convergence results from Chambolle and Pock (2016) of the convex Bregman Primal Dual algorithm cannot be adapted to αPGD and a new convergence result needs to be derived.

Theorem 22 (Convergence of αPGD (5.55))**.** *Assume f and g proper, lsc, lower-bounded, with f convex and L_f-smooth and g M-weakly convex. Then*[1] *for $\alpha \in (0,1)$ and $\tau < \min\left(\frac{1}{\alpha L_f}, \frac{\alpha}{M}\right)$, the updates* (5.55) *verify*

(i) $F(y_k) + \frac{\alpha}{2\tau}\left(1-\frac{1}{\alpha}\right)^2 ||y_k - y_{k-1}||^2$ *is non-increasing and converges.*

(ii) the sequence has finite length i.e. $\sum_{k=0}^{+\infty} ||y_{k+1} - y_k||^2 < +\infty$ *and* $||y_{k+1} - y_k||$ *converges to* 0 *at rate* $\min_{k<K} ||y_{k+1} - y_k|| = \mathcal{O}(1/\sqrt{K})$

(iii) All cluster points of the sequence y_k are stationary points of F. In particular, if g is coercive, then y_k has a subsequence converging towards a stationary point of F.

Remark 20. *With this theorem, αPGD is shown to verify convergence of the iterates and of the norm of the residual to* 0*. Note that we do not have here the analog of Theorem 7 on the single-point convergence using the KŁ hypothesis. Indeed, the general nonconvex convergence analysis with KŁ functions from Attouch et al. (2013), presented in Theorem 4, is not applicable. This is due to the fact that objective function $F(x_k)$ by itself does not decrease along the sequence, but $F(x_k) + \delta ||x_{k+1} - x_k||^2$ does (where $\delta = \frac{\alpha}{2\tau}\left(1-\frac{1}{\alpha}\right)^2$). Our situation is more similar to the variant of this result presented in (Ochs et al., 2014, Theorem 3.7). Indeed, with $\mathcal{F}(x,y) := F(x) + \delta ||x-y||^2$ and considering $\forall k \geq 1$, the sequence $z_k = (y_k, y_{k-1})$ (with y_k following our algorithm), we can easily show that z_k verifies the conditions H1 and H3 specified in (Ochs et al., 2014, Section 3.2). However, condition H2 does not extend to our algorithm. We plan as future work to derive a new version of (Ochs et al., 2014, Section 3.2) that fits to our case of interest.*

Remark 21. *When $\alpha = 1$, Algorithms αPGD* (5.55) *and PGD* (2.48) *are equivalent, but we get a slightly worst bound in Theorem 22 than in Theorem 20 ($\tau < \min\left(\frac{1}{L_f}, \frac{1}{M}\right) \leq \frac{2}{L_f+M}$). Nevertheless, when used with $\alpha < 1$, we next show that the relaxed algorithm is more relevant in the perspective of PnP with proximal denoiser.*

[1] As shown in the proof of Theorem 22, a better bound on τ can be found, but with little numerical gain.

Remark 22. *The data-fidelity term f is here assumed convex with Lipschitz gradient. This is verified by the classical L^2 data-fidelity term that we use in our experiments Section 5.3. However, this excludes several data-fidelity terms such as the L^1 or the Kullback-Leiber distances.*

Proof. **(i) and (ii) :** We can write (5.55b) as

$$\begin{aligned} x_{k+1} &\in \underset{y\in\mathbb{R}^n}{\arg\min}\, g(y) + \langle \nabla f(q_{k+1}), y - x_k\rangle + \frac{1}{2\tau}\,||y-x_k||^2 \\ &\in \underset{y\in\mathbb{R}^n}{\arg\min}\, g(y) + f(q_{k+1}) + \langle \nabla f(q_{k+1}), y - q_{k+1}\rangle + \frac{1}{2\tau}\,||y-x_k||^2 \\ &\in \underset{y\in\mathbb{R}^n}{\arg\min}\, \Phi(y) + \frac{1}{2\tau}\,||y-x_k||^2 \end{aligned} \tag{5.62}$$

with $\Phi(y) := g(y) + f(q_{k+1}) + \langle \nabla f(q_{k+1}), y - q_{k+1}\rangle$. As g is M-weakly convex, so does Φ. The three-points inequality of Corollary 1 (ii) applied to Φ thus gives $\forall y \in \mathbb{R}^n$,

$$\Phi(y) + \frac{1}{2\tau}\,||y-x_k||^2 \geq \Phi(x_{k+1}) + \frac{1}{2\tau}\,||x_{k+1}-x_k||^2 + \left(\frac{1}{2\tau} - \frac{M}{2}\right)||x_{k+1}-y||^2 \tag{5.63}$$

that is to say,

$$\begin{aligned} &g(y) + f(q_{k+1}) + \langle \nabla f(q_{k+1}), y - q_{k+1}\rangle + \frac{1}{2\tau}\,||y-x_k||^2 \geq g(x_{k+1}) + f(q_{k+1}) + \\ &\langle \nabla f(q_{k+1}), x_{k+1} - q_{k+1}\rangle + \frac{1}{2\tau}\,||x_{k+1}-x_k||^2 + \left(\frac{1}{2\tau} - \frac{M}{2}\right)||x_{k+1}-y||^2 . \end{aligned} \tag{5.64}$$

Using relation (5.55c), and the descent Lemma 15 as well as the convexity of f,

$$\begin{aligned} & f(q_{k+1}) + \langle \nabla f(q_{k+1}), x_{k+1} - q_{k+1}\rangle \\ =& f(q_{k+1}) + \left\langle \nabla f(q_{k+1}), \frac{1}{\alpha} y_{k+1} + \left(1 - \frac{1}{\alpha}\right) y_k - q_{k+1} \right\rangle \\ =& \frac{1}{\alpha}\Big(f(q_{k+1}) + \langle \nabla f(q_{k+1}), y_{k+1} - q_{k+1}\rangle\Big) \\ & + \left(1 - \frac{1}{\alpha}\right)\Big(f(q_{k+1}) + \langle \nabla f(q_{k+1}), y_k - q_{k+1}\rangle\Big) \\ \geq& \frac{1}{\alpha}\left(f(y_{k+1}) - \frac{L_f}{2}\,||y_{k+1}-q_{k+1}||^2\right) \\ & + \left(1 - \frac{1}{\alpha}\right)\Big(f(q_{k+1}) + \langle \nabla f(q_{k+1}), y_k - q_{k+1}\rangle\Big) \\ \geq& \frac{1}{\alpha}\left(f(y_{k+1}) - \frac{L_f}{2}\,||y_{k+1}-q_{k+1}||^2\right) + \left(1 - \frac{1}{\alpha}\right) f(y_k). \end{aligned} \tag{5.65}$$

Since $y_{k+1} - q_{k+1} = \alpha(x_{k+1} - x_k)$ (from relations (5.55a) and (5.55c)), by combining relations (5.64) and (5.65), we now have for all $y \in \mathbb{R}^n$,

$$\begin{aligned} &g(y) + \left(\frac{1}{\alpha} - 1\right) f(y_k) + \frac{1}{2\tau}\,||y-x_k||^2 + f(q_{k+1}) + \langle \nabla f(q_{k+1}), y - q_{k+1}\rangle \geq \\ &g(x_{k+1}) + \frac{1}{\alpha} f(y_{k+1}) + \left(\frac{1}{2\tau} - \frac{\alpha L_f}{2}\right)||x_{k+1}-x_k||^2 + \left(\frac{1}{2\tau} - \frac{M}{2}\right)||x_{k+1}-y||^2 . \end{aligned} \tag{5.66}$$

Using again the convexity of f we get for all $y \in \mathbb{R}^n$,

$$\begin{aligned} &g(y) + f(y) + \left(\frac{1}{\alpha} - 1\right) f(y_k) + \frac{1}{2\tau} ||y - x_k||^2 \geq \\ &g(x_{k+1}) + \frac{1}{\alpha} f(y_{k+1}) + \left(\frac{1}{2\tau} - \frac{\alpha L_f}{2}\right) ||x_{k+1} - x_k||^2 + \left(\frac{1}{2\tau} - \frac{M}{2}\right) ||x_{k+1} - y||^2 . \end{aligned} \tag{5.67}$$

Now, the weak convexity of g with relation (5.55c) gives

$$g(x_{k+1}) \geq \frac{1}{\alpha} g(y_{k+1}) + \left(1 - \frac{1}{\alpha}\right) g(y_k) - \frac{M}{2}(1 - \alpha) ||y_k - x_{k+1}||^2 . \tag{5.68}$$

Combining (5.67) and (5.68), and using $F = f + g$ leads to

$$\begin{aligned} \forall y \in \mathbb{R}^n \left(\frac{1}{\alpha} - 1\right) (F(y_k) - F(y)) + \frac{1}{2\tau} ||y - x_k||^2 \geq \\ \frac{1}{\alpha}(F(y_{k+1}) - F(y)) + \left(\frac{1}{2\tau} - \frac{\alpha L_f}{2}\right) ||x_{k+1} - x_k||^2 \\ + \left(\frac{1}{2\tau} - \frac{M}{2}\right) ||x_{k+1} - y||^2 - \frac{M}{2}(1 - \alpha) ||y_k - x_{k+1}||^2 . \end{aligned} \tag{5.69}$$

For $y = y_k$, we get

$$\begin{aligned} \frac{1}{\alpha}(F(y_k) - F(y_{k+1})) \geq -\frac{1}{2\tau} ||y_k - x_k||^2 + \left(\frac{1}{2\tau} - \frac{\alpha L_f}{2}\right) ||x_{k+1} - x_k||^2 \\ + \left(\frac{1}{2\tau} - \frac{M(2 - \alpha)}{2}\right) ||x_{k+1} - y_k||^2 . \end{aligned} \tag{5.70}$$

For constant $\alpha \in (0, 1)$, using that

$$y_k - x_{k+1} = \frac{1}{\alpha}(y_k - y_{k+1}) \tag{5.71}$$

$$y_k - x_k = \left(1 - \frac{1}{\alpha}\right) (y_k - y_{k-1}) \tag{5.72}$$

we get

$$\begin{aligned} F(y_k) - F(y_{k+1}) \geq -\frac{\alpha}{2\tau} \left(1 - \frac{1}{\alpha}\right)^2 ||y_k - y_{k-1}||^2 \\ + \alpha \left(\frac{1}{2\tau} - \frac{\alpha L_f}{2}\right) ||x_{k+1} - x_k||^2 \\ + \frac{1}{\alpha} \left(\frac{1}{2\tau} - \frac{M(2 - \alpha)}{2}\right) ||y_{k+1} - y_k||^2 . \end{aligned} \tag{5.73}$$

With the assumption $\tau \alpha L_f < 1$, the second term of the right-hand side is non-negative and therefore,

$$\begin{aligned} F(y_k) - F(y_{k+1}) &\geq -\frac{\alpha}{2\tau} \left(1 - \frac{1}{\alpha}\right)^2 ||y_k - y_{k-1}||^2 \\ &\quad + \frac{1}{\alpha} \left(\frac{1}{2\tau} - \frac{M(2 - \alpha)}{2}\right) ||y_{k+1} - y_k||^2 . \\ &= -\delta ||y_k - y_{k-1}||^2 + \delta ||y_{k+1} - y_k||^2 \\ &\quad + (\gamma - \delta) ||y_{k+1} - y_k||^2 \end{aligned} \tag{5.74}$$

with

$$\delta = \frac{\alpha}{2\tau}\left(1-\frac{1}{\alpha}\right)^2 \tag{5.75}$$

$$\gamma = \frac{1}{\alpha}\left(\frac{1}{2\tau}-\frac{M(2-\alpha)}{2}\right) \tag{5.76}$$

We now make use of the following lemma.

Lemma 7 (Bauschke and Combettes (2011b)). *Let $(a_n)_{n\in\mathbb{N}}$ and $(b_n)_{n\in\mathbb{N}}$ be two real sequences such that $b_n \geq 0\ \forall n \in \mathbb{N}$, (a_n) is bounded from below and $a_{n+1}+b_n \leq a_n\ \forall n \in \mathbb{N}$. Then $(a_n)_{n\in\mathbb{N}}$ is a non-increasing and convergent sequence and $\sum_{n\in\mathbb{N}} b_n < +\infty$*

To apply Lemma 7 we look for a stepsize satisfying

$$\gamma - \delta > 0 \quad i.e. \quad \tau < \frac{\alpha}{M}. \tag{5.77}$$

Hypothesis $\tau < \min\left(\frac{1}{\alpha\lambda L_f}, \frac{\alpha}{M}\right)$ gives that $(F(y_k)+\delta\,||y_k-y_{k-1}||^2)$ is a non-increasing and convergent sequence and that $\sum_k ||y_k-y_{k+1}||^2 < +\infty$.

Note that a slightly more precise bound can be found keeping the second term in (5.73). For the sake of completeness, we develop it here. Keeping the assumption $\tau\alpha L_f < 1$ in (5.73), we can use that

$$x_{k+1}-x_k = \frac{1}{\alpha}(y_{k+1}-y_k) + \left(1-\frac{1}{\alpha}\right)(y_k-y_{k-1}). \tag{5.78}$$

Then, by convexity of the squared ℓ_2 norm, for $0<\alpha<1$, we have

$$||y_{k+1}-y_k||^2 \leq \alpha\,||x_{k+1}-x_k||^2 + (1-\alpha)\,||y_k-y_{k-1}||^2 \tag{5.79}$$

and

$$||x_{k+1}-x_k||^2 \geq \frac{1}{\alpha}\,||y_{k+1}-y_k||^2 + \left(1-\frac{1}{\alpha}\right)||y_k-y_{k-1}||^2 \tag{5.80}$$

which gives finally

$$\begin{aligned}
&F(y_k)-F(y_{k+1}) \geq \\
&\left(\alpha\left(1-\frac{1}{\alpha}\right)\left(\frac{1}{2\tau}-\frac{\alpha L_f}{2}\right)-\frac{\alpha}{2\tau}\left(1-\frac{1}{\alpha}\right)^2\right)||y_k-y_{k-1}||^2 \\
&+\left(\frac{1}{2\tau}-\frac{\alpha L_f}{2}+\frac{1}{\alpha}(\frac{1}{2\tau}-\frac{M(2-\alpha)}{2})\right)||y_{k+1}-y_k||^2 \\
&=-\frac{1-\alpha}{2\alpha\tau}\left(1-\alpha^2\tau L_f\right)||y_k-y_{k-1}||^2 \\
&+\frac{1}{2\alpha\tau}\left(1+\alpha-\alpha^2\tau L_f-\tau M(2-\alpha)\right)||y_{k+1}-y_k||^2 \\
&=-\delta\,||y_k-y_{k-1}||^2+\delta\,||y_{k+1}-y_k||^2+(\gamma-\delta)\,||y_{k+1}-y_k||^2
\end{aligned} \tag{5.81}$$

with

$$\delta = \frac{1-\alpha}{2\alpha\tau}\left(1-\alpha^2\tau L_f\right) \tag{5.82}$$

$$\gamma = \frac{1}{2\alpha\tau}\left(1-\alpha^2\tau L_f+\alpha-\tau M(2-\alpha)\right). \tag{5.83}$$

The condition on the stepsize becomes

$$\begin{aligned} &\gamma - \delta > 0 \\ &\Leftrightarrow \tau < \frac{2\alpha}{\alpha^3 L_f + (2-\alpha)M}. \end{aligned} \tag{5.84}$$

And the overall condition is

$$\tau < \min\left(\frac{1}{\alpha L_f}, \frac{2\alpha}{\alpha^3 L_f + (2-\alpha)M}\right). \tag{5.85}$$

(iii) The proof of this result is an extension of the proof of Proposition 20 in the context of the classical PGD. Suppose that a subsequence (y_{k_i}) is converging towards y. Let us show that y is a critical point of F. From (5.55b), we have

$$\frac{x_{k+1} - x_k}{\tau} - \nabla f(q_{k+1}) \in \partial g(x_{k+1}). \tag{5.86}$$

First we show that $x_{k_i+1} - x_{k_i} \to 0$. We have $\forall k > 1$,

$$\begin{aligned} ||x_{k+1} - x_k|| &= \left|\left|\frac{1}{\alpha}y_{k+1} + (1-\frac{1}{\alpha})y_k - \frac{1}{\alpha}y_k - (1-\frac{1}{\alpha})y_{k-1}\right|\right| \\ &\leq \frac{1}{\alpha}\,||y_{k+1} - y_k|| + (\frac{1}{\alpha} - 1)\,||y_k - y_{k-1}|| \\ &\to 0. \end{aligned} \tag{5.87}$$

From (5.55a), we also get $||q_{k+1} - q_k|| \to 0$. Now, let us show that $x_{k_i} \to y$ and $q_{k_i} \to y$. First using (5.55c), we have

$$\begin{aligned} ||x_{k_i} - y|| &\leq ||x_{k_i+1} - y|| + ||x_{k_i+1} - x_{k_i}|| \\ &\leq \frac{1}{\alpha}\,||y_{k_i+1} - y|| + (\frac{1}{\alpha} - 1)\,||y_{k_i} - y|| + ||x_{k_i+1} - x_{k_i}|| \\ &\to 0. \end{aligned} \tag{5.88}$$

Second, from (5.55a), we get in the same way $q_{k_i} \to y$. From the continuity of ∇f, we get $\nabla f(q_{k_i}) \to \nabla f(y)$ and therefore

$$\frac{x_{k_i} - x_{k_i-1}}{\tau} - \nabla f(q_{k_i}) \to -\nabla f(y). \tag{5.89}$$

If we can also show that $g(x_{k_i}) \to g(y)$, we get from the closeness of the limiting subdifferential (Proposition 2) that $-\nabla f(y) \in \partial g(y)$ *i.e.* y is a critical point of F.

Using the fact that g is lsc and $x_{k_i} \to y$, we have

$$\liminf_{i\to\infty} g(x_{k_i}) \geq g(y). \tag{5.90}$$

On the other hand, with Equation (5.67) for $k+1 = k_i$, taking $i \to +\infty$, $||y - x_{k_i+1}|| \to 0$, $||y - x_{k_i}|| \to 0$, $f(y_{k_i}) \to f(y)$, $f(y_{k_i+1}) \to f(y)$ and we get

$$\limsup_{i\to\infty} g(x_{k_i}) \leq g(y), \tag{5.91}$$

and therefore

$$\lim_{i\to\infty} g(x_{k_i}) = g(y). \tag{5.92}$$

As f is lower-bounded, if g is coercive, so is F and by (i) the iterates (y_k) remain bounded and (y_k) admits a converging subsequence. □

ProxPnP-αPGD algorithm

Similar to the derivation of the PnP algorithms from first order algorithms in Section 2.4.2, the αPGD algorithm (5.55) gives birth to the PnP-αPGD algorithm by replacing the proximity operator $\text{Prox}_{\tau g}$ by a Gaussian denoiser D_σ. Now, similar to what was done Section 5.2.1 with the PGD algorithm, in this section, we study the convergence of this PnP-αPGD with our particular Proximal Gradient-Step Denoiser (5.4) $D_\sigma = \text{Prox}_{\phi_\sigma}$ and stepsize $\tau = 1$. This corresponds to the following algorithm, which we refer to ProxPnP-αPGD.

$$\begin{cases} q_{k+1} = (1-\alpha)y_k + \alpha x_k & \text{(5.93a)} \\ x_{k+1} = D_\sigma(x_k - \lambda\nabla f(q_{k+1})) = \text{Prox}_{\phi_\sigma}(x_k - \lambda\nabla f(q_{k+1})) & \text{(5.93b)} \\ y_{k+1} = (1-\alpha)y_k + \alpha x_{k+1}. & \text{(5.93c)} \end{cases}$$

Like ProxPnP-PGD (5.18), ProxPnP-αPGD scheme targets the critical points of the explicit functional $F = \lambda f + \phi_\sigma$ where ϕ_σ is obtained from the deep potential g_σ via Proposition 25. Applying the previous Theorem 22 on the convergence of the αPGD algorithm with $\tau = 1$ and $g = \phi_\sigma$ which is $M = \frac{L}{L+1}$ weakly convex, we get the following convergence result for ProxPnP-αPGD.

Corollary 5 (Convergence of ProxPnP-αPGD (5.93))**.** *Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ convex and differentiable with L_f-Lipschitz gradient, bounded from below. Assume that L the Lipschitz constant of ∇g_σ verifies $L < 1$. Let $M = \frac{L}{L+1}$ the weak convexity constant of ϕ_σ obtained from g_σ via Proposition 25. Then, for any $\alpha \in (0,1)$ such that*

$$\lambda L_f M < 1 \text{ and } M < \alpha < 1/(\lambda L_f) \tag{5.94}$$

the iterates given by the iterative scheme (5.93) *verify*

(i) $F(y_k) + \frac{\alpha}{2}\left(1 - \frac{1}{\alpha}\right)^2 ||y_k - y_{k-1}||^2$ *is non-increasing and converges.*

(ii) The sequence y_k as finite length i.e. $\sum_{k=0}^{+\infty} ||y_{k+1} - y_k||^2 < +\infty$ *and* $||y_{k+1} - y_k||$ *converges to* 0 *at rate* $\min_{k<K} ||y_{k+1} - y_k|| = \mathcal{O}(1/\sqrt{K})$

(iii) If g_σ is coercive, then (y_k) has a subsequence converging towards a critical point of F.

The existence of $\alpha \in (0,1)$ satisfying relation (5.94) is ensured as soon as $\lambda L_f M < 1$. As a consequence, when M gets small (*i.e.* ϕ_σ gets "more convex") λL_f can get arbitrarily large. This is a major improvement compared to ProxPnP-PGD which was limited (Theorem 19) to $\lambda L_f < 2$ even for convex ϕ ($M = 0$). To further exploit this property, we now consider the relaxed denoiser D_σ^γ (5.15) that is associated to a function ϕ_σ^γ with a tunable weak convexity constant M^γ.

Corollary 6 (Convergence of ProxPnP-αPGD with relaxed denoiser)**.** *Let ϕ_σ^γ the $M^\gamma = \frac{\gamma L}{\gamma L+1}$-weakly convex potential introduced in* (5.16) *and $L < 1$. Then, for $M^\gamma < \alpha < 1/(\lambda L_f)$, the iterates y_k given by the ProxPnP-αPGD* (5.93) *with γ-relaxed denoiser D_σ^γ defined in* (5.15) *verify the convergence properties (i)-(iii) of Corollary 5 for $F = \lambda f + \phi_\sigma^\gamma$.*

Therefore, using the γ-relaxed denoiser $D_\sigma^\gamma = \gamma D_\sigma + (1-\gamma)\,\text{Id}$, the overall convergence condition on λ is now

$$\lambda < \frac{1}{L_f}\left(1 + \frac{1}{\gamma L}\right). \tag{5.95}$$

Provided γ gets small, λ can be arbitrarily large. Small γ means small amount of regularization brought by denoising at each step of the PnP algorithm. Moreover, for small γ, the targeted regularization function ϕ_σ^γ gets close to a convex function. It has already been observed that deep convex regularization can be suboptimal compared to more flexible nonconvex ones (Cohen et al., 2021). Depending on the inverse problem, and on the necessary amount of regularization, the choice of the couple (γ, λ) will be of paramount importance for efficient restoration.

5.3 Experiments

5.3.1 Proximal GS denoiser

In this section, we learn a denoiser $D_\sigma = \mathrm{Id} - \nabla g_\sigma$ that verifies the conditions of Proposition 25, and that can thus be written as a proximal mapping. As explained in Section 5.1.2, we design the potential g_σ as

$$g_\sigma(x) = \frac{1}{2} ||x - N_\sigma(x)||^2 \tag{5.96}$$

where $N_\sigma : \mathbb{R}^n \to \mathbb{R}^n$ is a $\mathcal{C}^\infty$ neural network with softplus activations. Remind that in this Chapter, we do not regularize inverse problems directly with g_σ but with the potential ϕ_σ, obtained from g_σ with Proposition 25. As detailed in Sections 5.1.2 and 5.2, this choice of g_σ enables ϕ_σ to verify the sufficient regularity conditions required for the convergence of the PnP-PGD and PnP-DRS with Gradient-Step denoiser (5.2).

Denoising Network Architecture Similar to the original Gradient-Step denoiser presented in Chapter 4, we choose to parameterize N_σ with the architecture DRUNet Zhang et al. (2021) (represented in Figure 2.1), a UNet in which residual blocks are integrated. DRUNet takes the noise level σ as input, which is consistent with our formulation. In order to ensure continuous differentiability w.r.t. the input, we change RELU activations to Softplus, which is $\mathcal{C}^\infty$. We also limit the number of residual blocks to 2 at each scale to lower the computational burden.

Training details We first train the GS denoiser, in the same conditions as Chapter 4, with the L^2 loss

$$\mathcal{L}(\sigma) = \mathbb{E}_{x\sim p, \xi_\sigma \sim \mathcal{N}(0,\sigma^2)} \left[||D_\sigma(x + \xi_\sigma) - x||^2 \right], \tag{5.97}$$

where p is the distribution of a database of clean images. However, the resulting denoiser does not verify $\nabla g_\sigma = \mathrm{Id} - D_\sigma$ contractive *i.e.* with $L < 1$ Lipschitz gradient, as required by Proposition 25.

The link between the Lipschitz constant of N_σ and the one of ∇g_σ being difficult to establish, following Pesquet et al. (2021), we enforce $L < 1$ by regularizing the training loss of D_σ with the spectral norm of the Hessian of g_σ that reads $\nabla^2 g_\sigma = J_{(\mathrm{Id} - D_\sigma)}$. More specifically, we fine-tune the previously trained GS denoiser with the following loss:

$$\mathcal{L}_S(\sigma) = \mathbb{E}_{x\sim p, \xi_\sigma \sim \mathcal{N}(0,\sigma^2)} \Big[||D_\sigma(x + \xi_\sigma) - x||^2 + \mu \max(\left|\left| J_{(\mathrm{Id} - D_\sigma)}(x + \xi_\sigma) \right|\right|_S, 1 - \epsilon) \Big] \tag{5.98}$$

where $||.||_S$ denotes the spectral (or operator) norm. During training, it is estimated with 50 power iterations. In practice, we fine-tune during 10 epochs, with σ ranging in $[0, 25]$, using various μ and $\epsilon = 0.1$. The resulting denoiser is called Prox-DRUNet.

Denoising results We evaluate the PSNR performance of the proposed Prox-DRUNet denoiser, fine-tuned with the loss (5.98) with different values of μ. In Table 5.1, we compare, for various noise levels σ, our model with DRUNet equipped with the same architecture N_σ as our prox-DRUNet (2 residual blocks and softplus activations) and trained with a L^2 loss as in (5.97). We also present the results obtained with standard GS-DRUNet from Chapter 4, which corresponds to the same architecture trained with the loss (5.97), without the Lipschitz fine-tuning step. We also indicate the performance of the classical FFDNet Zhang et al. (2018) and DnCNN Zhang et al. (2017a) denoisers.

$\sigma(./255)$	5	10	15	20	25
FFDNet	39.95	35.81	33.53	31.99	30.84
DnCNN	39.80	35.82	33.55	32.02	30.87
DRUNet	40.19	36.10	33.85	32.34	31.21
GS-DRUNet	40.27	36.16	33.92	32.41	31.28
Prox-DRUNet ($\mu = 10^{-3}$)	40.12	35.93	33.60	32.01	30.82
Prox-DRUNet ($\mu = 10^{-2}$)	40.04	35.86	33.51	31.88	30.64

Table 5.1: Average PSNR performance of our prox-denoiser and compared methods on 256×256 center-cropped images from the CBSD68 dataset Martin et al. (2001), for various noise levels σ.

As can be seen in Table 5.1, the Lipschitz fine-tuning step inevitably decreases the denoising performance, especially for larger values of μ and higher noise levels. Nevertheless, for $\mu = 10^{-2}$ prox-DRUNet shows comparable performance with FFDNet and DnCNN. The decrease in PSNR when constraining the Lipschitz constant of $\mathrm{Id} - D_\sigma$ is thus limited.

Lipschitz constant In our experiments, the Lipschitz constant of $\nabla g_\sigma = \mathrm{Id} - D_\sigma$ is not hardly constrained to satisfy $L < 1$. This property is rather softly enforced by penalization with the loss function (5.98). We now investigate the potential gap between the theoretical assumption $L < 1$ and the practical considerations. We evaluate in Table 5.2 the maximum value of $\left|\left|J_{(\mathrm{Id} - D_\sigma)}(x)\right|\right|_S$ while denoising noisy images from the CBSD68 testing dataset. Here, the value $\left|\left|J_{(\mathrm{Id} - D_\sigma)}(x)\right|\right|_S$ is computed by running the power method until convergence. Table 5.2 first illustrates that GS-DRUNet ($\mu = 0$) does not check the $L < 1$ Lipschitz property, especially for very noisy images. Next, for a large enough penalization parameter ($\mu = 10^{-2}$), prox-DRUNet satisfies the $L < 1$ constraint, on the CBSD68 testing dataset and at all studied noise levels.

Tables 5.1 and 5.2 exhibit a clear trade-off, controlled by μ, between denoising performance and Lipschitz constant. Contrary to nonexpansive denoisers (see the paragraph below), our constrained proximal denoiser provides high quality denoising while empirically validating the Lipschitz constraint.

$\sigma(./255)$	0	5	10	15	20	25
GS-DRUNet ($\mu = 0$)	0.94	1.26	2.47	1.96	2.50	3.27
Prox-DRUNet ($\mu = 10^{-2}$)	0.87	0.92	0.95	0.99	0.96	0.96
Prox-DRUNet ($\mu = 10^{-3}$)	0.86	0.94	0.97	0.98	0.99	1.19

Table 5.2: Maximal value of $\left|\left|J_{(\mathrm{Id} - D_\sigma)}(x)\right|\right|_S$ obtained with proximal denoisers on 256×256 center-cropped CBSD68 dataset, for various noise levels σ.

We precise that in all the PnP experiments conducted with Prox-DRUNet, we empirically verified that we still have $||\nabla^2 g_\sigma(x_k)||_S < 1$ on all the iterates x_k where D_σ was evaluated.

Comparison with a nonexpansivity constraint In this paragraph, we compare the performance of the previous Prox-DRUNet denoiser, trained to have $\mathrm{Lip}(\mathrm{Id} - D_\sigma) < 1$, with the performance of the same network trained to be nonexpansive *i.e.* $\mathrm{Lip}(D_\sigma) \leq 1$. To do so, we use the same strategy to train DRUNet directly to be nonexpansive. More precisely, the pretrained DRUNet denoiser N_σ is fine-tuned to denoise while being 1-Lipschitz with the following loss:

$$\mathbb{E}_{x\sim p, \xi_\sigma \sim \mathcal{N}(0,\sigma^2)} \left[||N_\sigma(x+\xi_\sigma) - x||^2 + \mu \max(||J_{N_\sigma}(x+\xi_\sigma)||_S, 1-\epsilon) \right] \tag{5.99}$$

with $\epsilon = 0.1$ and different values of μ.

We analyze the denoising performance of the resulting denoiser nonexp-DRUNet (Table 5.3) as well as the maximal value of the spectral norm $||J_{N_\sigma}(x)||_S$, on the CBSD68 testset (Table 5.4). Table 5.4 illustrates that the penalization parameter has to be set to $\mu = 10^{-2}$ to make the Lipschitz constant lower than 1. On the other hand, as can be observed in Table 5.3, such setting severely degrades denoising performance.

From this observation, we argue that it is less harmful for the denoising performance to impose nonexpansivity on the denoiser residual $\mathrm{Id} - D_\sigma$ than on the denoiser itself.

$\sigma(./255)$	5	10	15	20	25
GS-DRUNet	40.27	36.16	33.92	32.41	31.28
prox-DRUNet ($\mu = 10^{-2}$)	40.04	35.86	33.51	31.88	30.64
prox-DRUNet ($\mu = 10^{-3}$)	40.12	35.93	33.60	32.01	30.82
nonexp-DRUNet ($\mu = 10^{-2}$)	34.92	32.90	31.42	30.30	29.42
nonexp-DRUNet ($\mu = 10^{-3}$)	39.71	35.71	33.50	32.00	30.89

Table 5.3: Average PSNR denoising performance of our prox-denoiser and compared methods on 256×256 center-cropped images from the CBSD68 dataset, for various noise levels σ.

$\sigma(./255)$	0	5	10	15	20	25
DRUNet ($\mu = 0$)	1.13	1.73	2.36	2.67	2.76	3.22
nonexp-DRUNet ($\mu = 10^{-2}$)	0.97	0.98	0.98	0.98	0.98	0.98
nonexp-DRUNet ($\mu = 10^{-3}$)	1.06	1.07	1.07	1.10	1.13	1.20

Table 5.4: Maximal value of $\left|\left|J_{D_\sigma(x)}\right|\right|_S$ obtained with contractive denoisers $D_\sigma = N_\sigma$ on 256×256 center-cropped CBSD68 dataset, for various noise levels σ.

5.3.2 PnP restoration

In this section, we apply, with the proximal denoiser Prox-DRUNET, the PnP algorithms ProxPnP-PGD (5.18), ProxPnP-αPGD (5.93), ProxPnP-DRSdiff (5.23) (*diff* specifies that this PnP-DRS is dedicated to differentiable data-fidelity terms f) and ProxPnP-DRS (5.28) for deblurring and super-resolution with Gaussian noise.

We use the same experimental protocol as the one described in Chapter 4, Section 4.3.2. Refer to this section for more details. We recall that we seek an estimate of a clean image $x \in \mathbb{R}^n$, from a degraded observation obtained as $y = Ax + \xi_\nu \in \mathbb{R}^m$, with A a $m \times n$ degradation matrix and ξ_ν a white Gaussian noise with zero mean and standard deviation ν. With this formulation, the data-fidelity term is $f(x) = \frac{1}{2}||Ax - y||^2$ and the Lipschitz constant of ∇f is $L_f = \left|\left|A^T A\right|\right|_S$. The blur kernels are normalized so that the Lipschitz constant $L_f = 1$. The minimized objective function is

$$F(x) = \lambda f(x) + \phi_\sigma(x). \tag{5.100}$$

We use for evaluation and comparison the 68 images from the CBSD68 dataset, center-cropped to the size 256×256 and Gaussian noise with 3 noise levels $\nu \in \{2.55, 7.65, 12.75\}/255$ *i.e.* $\nu \in \{0.01, 0.03, 0.05\}$.

Hyperparameter selection Convergence of ProxPnP-PGD, ProxPnP-αPGD, ProxPnP-DRSdiff and ProxPnP-DRS are respectively guaranteed by Theorem 19, Corollary 5, Theorem 20 and Theorem 21. For each algorithm, we propose default values for the different hyperparameters involved. These hyperparameter values are explicitly given in Table 5.5. Note that we use the same choice of hyperparameters for both deblurring and super-resolution. The λ parameter always satisfies the corresponding constraint required for convergence. As each algorithm uses its own set of parameters depending on the constraint on λ, each algorithm targets critical points of a different functional.

For both ProxPnP-PGD and ProxPnP-αPGD algorithm, we use the γ-relaxed version of the denoiser (5.15). In practice, we found that the same choice of parameters γ and σ is optimal for both PGD and αPGD, with values depending on the amount of noise ν in the input image. We thus choose $\lambda \in [0, \lambda_{lim}]$ where, following Theorem 19 and Corollary 5, for ProxPnP-PGD $\lambda_{lim}^{\text{PGD}} = \frac{1}{L_f}\frac{\gamma+2}{\gamma+1}$ and for ProxPnP-αPGD $\lambda_{lim}^{\alpha\text{PGD}} = \frac{1}{L_f}\frac{\gamma+1}{\gamma} \geq \lambda_{lim}^{\text{PGD}}$. For both $\nu = 0.01$ and $\nu = 0.03$, λ is set to its maximal allowed value λ_{lim}. As $\lambda_{lim}^{\alpha\text{PGD}} \geq \lambda_{lim}^{\text{PGD}}$, ProxPnP-$\alpha$PGD is expected to outperform ProxPnP-PGD at these noise levels. Finally, for ProxPnP-αPGD, α is set to its maximum possible value $1/(\lambda L_f)$.

For ProxPnP-DRSdiff, λ is also set to its maximal possible $\frac{1}{L_f}$ for theoretical convergence.

For ProxPnP-DRS, Theorem 21 requires $L < L_{max}(\beta)$ via the constraint (5.31). As D_σ is trained to ensure $L < 1$, we do not retrain the denoiser for a specific $L_{max}(\beta)$ value, but we use again the γ-relaxed version of the denoiser D_σ^γ (5.15) with $\gamma = L_{max}(\beta)$. $\gamma \nabla g_\sigma$ is then $L_{max}(\beta)$-Lipschitz and $D_\sigma^\gamma = \text{Prox}_{\phi_\sigma^\gamma}$. In practice, we find $\beta = 0.25$ to be a good compromise. For this choice of β, (5.31) is satisfied for $L_{max}(\beta) \leq 0.45$.

Deblurring For image deblurring, the degradation operator $A = H$ is a convolution performed with circular boundary conditions. As detailed in Section 4.3, the proximal operator of f can be efficiently calculated using the discrete Fourier transform. We use the same blur kernels than in Section 4.3 (8 real-world camera shake kernels, as well as the 9×9 uniform kernel and the 25×25 Gaussian kernel with standard deviation 1.6).

Super-resolution For single image super-resolution (SR), an efficient closed-form calculation of the proximal map for the data-fidelity term is provided in Section 4.3. We evaluate SR performance on the 4 isotropic Gaussian blur kernels from Table 4.4 with different standard deviations (0.7, 1.2, 1.6 and 2.0). We consider down-sampled images at scales $s = 2$ and $s = 3$.

$\nu(./255)$		2.55	7.65	12.75
PGD	γ	0.6	1	1
	$\lambda = \frac{\gamma+2}{\gamma+1}$	1.625	1.5	1.5
	σ/ν	1.25	0.75	0.5
αPGD	γ	0.6	1	1
	$\lambda = \frac{\gamma+1}{\gamma}$	2.66	2	2
	$\alpha = \frac{1}{\lambda}$	0.37	0.5	0.5
	σ/ν	1.25	0.75	0.5
DRS	β	0.25	0.25	0.25
	$\gamma = L_{max}(\beta)$	0.45	0.45	0.45
	λ	5	1.5	0.75
	σ/ν	2	1	0.5
DRSdiff	β	0.5	0.5	0.5
	λ	1.	1.	1.
	σ/ν	0.75	0.5	0.5

Table 5.5: Choice of the different hyperparameters involved for each ProxPnP algorithm. The same set of hyperparameters are used for both debluring and super-resolution experiments.

Numerical performance analysis We numerically evaluate in Table 5.7, for deblurring and Table 5.8 for super-resolution, the PSNR performance of our four ProxPnP algorithms. We give comparisons with the deep state-of-the-art PnP methods IRCNN (Zhang et al., 2017b) and DPIR (Zhang et al., 2021) which both apply the PnP-HQS algorithm with decreasing stepsize but without convergence guarantees. We also provide comparisons with the GSRED-PGD method presented in Chapter 4. We finally indicate the deblurring performance of *nonexp-PnP-PGD*, the PnP-PGD algorithm applied with the denoiser *nonexp-DRUNet* trained to be nonexpansive (see Section 5.3.1).

Observe that, among ProxPnP methods, ProxPnP-DRS and ProxPnP-αPGD give the best performance over the variety of kernels and noise levels. Indeed, ProxPnP-DRS convergence is guaranteed whatever be the value of λ, which can thus be tuned to optimize performance. ProxPnP-DRSdiff is, on the other hand, constrained to $\lambda < 1$. Similarly, ProxPnP-PGD is constrained to $\lambda < 2$ when ProxPnP-αPGD restriction on λ is relaxed. When a low amount of regularization (*i.e.* a large λ value) is necessary, an upper bound on λ can severely limit the restoration capacity of the algorithm. Indeed, we observe that when the input noise is low ($\nu = 2.55$), ProxPnP-DRSdiff and ProxPnP-PGD perform significantly worse. However, when the input noise is high, a stronger regularization is necessary (*i.e.* a small λ value) and all methods perform comparably.

We also provide visual comparisons for deblurring in Figure 5.1 and super-resolution in Figure 5.2. Along with the output images, for each ProxPnP algorithm, we plot the evolution of the corresponding provably-decreasing function. We recall in Table 5.6, for each algorithm, the formula of each function that is proved to decrease and converge along the iterations. We also plot the evolution of the norm of the residuals and of the PSNR along the iterates. These plots empirically confirm the theoretical convergence results. Observe that, despite being trained to guarantee convergence, ProxPnP-αPGD and ProxPnP-DRS globally compare with the performance of the state-of-the-art DPIR

method. Finally, note that, as for deblurring, the PnP performance is significantly reduced when plugging a nonexpansive denoiser (*nonexp-PnP-PGD*) rather than a nonexpansive residual.

Algorithm	Decreasing function
ProxPnP-PGD	$F(x)$
ProxPnP-αPGD	$F^{\alpha}(y) = F(y) + \frac{\alpha}{2}\left(1 - \frac{1}{\alpha}\right)^2 \|\|y - y_-\|\|^2$
ProxPnP-DRSdiff	$F^{DR}(x,y,z) = \lambda f(y) + \phi_\sigma(z) + \langle y - x, y - z\rangle + \frac{1}{2}\|\|y - z\|\|^2$
ProxPnP-DRS	$F^{DR,2}(x,y,z) = \lambda f(z) + \phi_\sigma(y) + \langle y - x, y - z\rangle + \frac{1}{2}\|\|y - z\|\|^2$

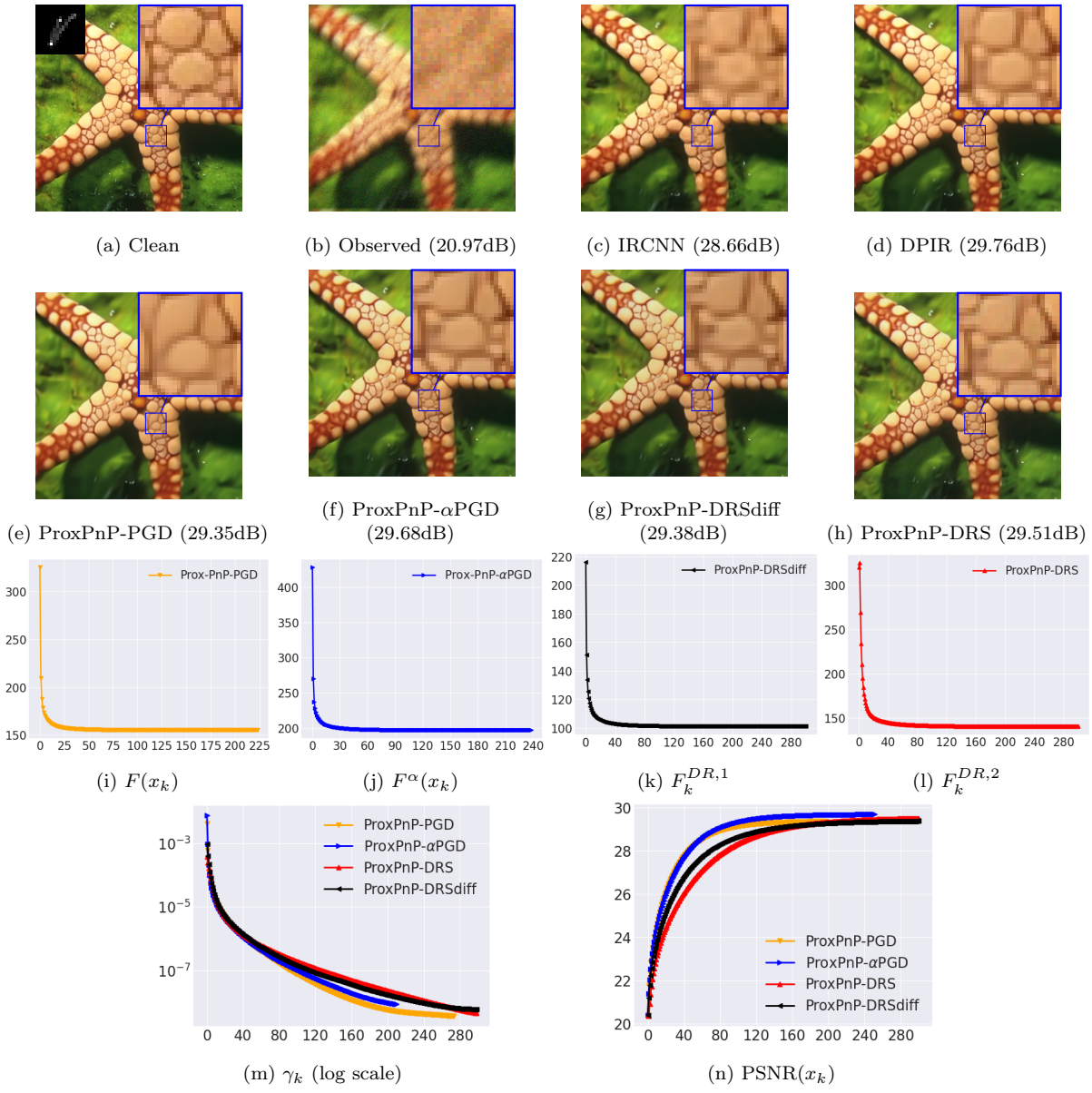
Table 5.6: For each ProxPnP algorithm, the corresponding convergence theorem exhibits a function that is proved to decrease along the iterates.

$\sigma(./255)$	2.55	7.65	12.75
IRCNN	31.42	28.01	26.40
DPIR	31.93	28.30	26.82
GSRED-PGD	31.70	28.28	26.86
ProxPnP-PGD	30.91	27.97	26.66
ProxPnP-αPGD	31.55	28.03	26.66
ProxPnP-DRSdiff	30.56	27.78	26.61
ProxPnP-DRS	31.51	28.01	26.63
nonexp-PnP-PGD	30.25	27.06	25.30

Table 5.7: PSNR (dB) of deblurring methods on CBSD68. PSNR are averaged over 10 blur kernels for various noise levels ν.

Method	$s = 2$			$s = 3$		
	2.55	7.65	12.75	2.55	7.65	12.75
IRCNN	26.97	25.86	25.45	25.60	24.72	24.38
DPIR	27.79	26.58	25.83	26.05	25.27	24.66
GS-PnP	27.88	26.81	26.01	25.97	25.35	24.74
ProxPnP-PGD	27.68	26.57	25.81	25.94	25.20	24.62
ProxPnP-αPGD	27.92	26.61	25.80	26.03	25.26	24.61
ProxPnP-DRSdiff	27.44	26.58	25.82	25.75	25.19	24.63
ProxPnP-DRS	27.95	26.58	25.81	26.13	25.27	24.65
nonexp-PnP-PGD	27.13	26.20	25.40	23.83	24.57	24.01

Table 5.8: PSNR (dB) of SR methods on CBSD68. PSNR averaged over 4 blur kernels for various scales s and noise levels ν.



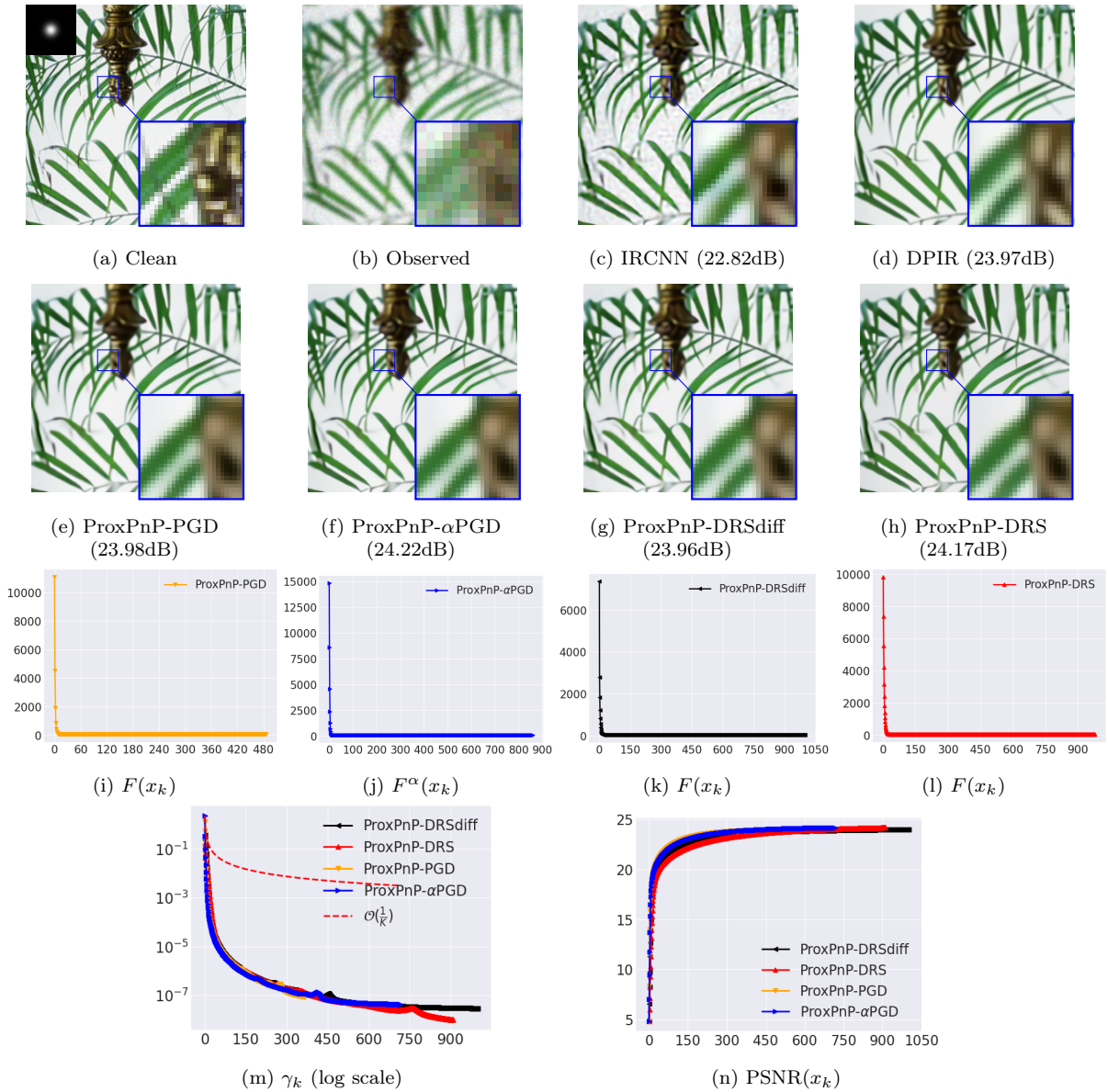
(a) Clean (b) Observed (20.97dB) (c) IRCNN (28.66dB) (d) DPIR (29.76dB)

(e) ProxPnP-PGD (29.35dB) (f) ProxPnP-αPGD (29.68dB) (g) ProxPnP-DRSdiff (29.38dB) (h) ProxPnP-DRS (29.51dB)

(i) $F(x_k)$ (j) $F^{\alpha}(x_k)$ (k) $F_k^{DR,1}$ (l) $F_k^{DR,2}$

(m) γ_k (log scale) (n) PSNR(x_k)

Figure 5.1: Deblurring with various methods of "starfish" degraded with the indicated blur kernel and input noise level $\nu = 0.03$. We also plot for each algorithm the evolution of the respective decreasing Lyapunov functions, the residual $\gamma_k = \min_{0 \leq i \leq k} ||x_{i+1} - x_i||^2 / ||x_0||^2$ and the PSNR.



(a) Clean (b) Observed (c) IRCNN (22.82dB) (d) DPIR (23.97dB)

(e) ProxPnP-PGD (23.98dB) (f) ProxPnP-αPGD (24.22dB) (g) ProxPnP-DRSdiff (23.96dB) (h) ProxPnP-DRS (24.17dB)

(i) $F(x_k)$ (j) $F^{\alpha}(x_k)$ (k) $F(x_k)$ (l) $F(x_k)$

(m) γ_k (log scale) (n) PSNR(x_k)

Figure 5.2: Super-resolution with various methods on "leaves" downsampled by 2, with the indicated blur kernel and input noise level $\nu = 0.03$. We plot the evolution of the respective Lyapunov functions, the residual $\gamma_k = \min_{0 \leq i \leq k} ||x_{i+1} - x_i||^2 / ||x_0||^2$ and the PSNR.

5.4 Conclusion

In this chapter, we provided new convergence results for PnP schemes without altering their restoration performance. We proposed to learn a denoiser D_σ which write as the proximity operator of a weakly convex regularization function. This proximal denoiser is not limited by nonexpansivity and competes with unconstrained state-of-the-art denoisers. We showed that, by plugging this denoiser, PnP-PGD, PnP-ADMM and PnP-DRS algorithms are guaranteed to converge to a stationary point of an explicit functional. Moreover, we introduced a new convergent plug-and-play algorithm built from a relaxed version of the Proximal Gradient Descent (PGD) algorithm, which converges for a wider range of regularization parameters. Quantitative experiments on ill-posed IR tasks, including data-fidelity terms that are not strongly convex, confirmed the relevance of the method, and showed that it allows to precisely monitor convergence of the numerical schemes.

In the last two chapters, the studied PnP and RED algorithms were applicable when the data-fidelity term was either smooth (PnP-PGD and RED-GD) or proximable (PnP-DRS and RED-PGD). In our applications, we only considered the L^2 data-fidelity term, which is both smooth and proximable. However, this is not true any more for Poisson Inverse problems, for which the Kullback-Leibler data-fidelity term is neither smooth, nor proximable. In the next chapter, we use a different notion of smoothness, defined by a Bregman divergence, for treating such Poisson inverse problems.

Chapter 6

Bregman Plug-and-Play for Poisson Inverse Problems

Contents

With the Bayesian variational formulation of image inverse problems (2.5), the data-fidelity terms is the negative log-likelihood $f(x) = -\log p(y|x)$ of the probabilistic observation model chosen to describe the physics of the acquisition. As explained in Section 2.1.2, in numerous applications such as Positron Emission Tomography (PET) or astronomical CCD cameras (Bertero et al., 2009), where images are obtained by counting particles (photons or electrons), it is common to use the Poisson noise model $y \sim \mathcal{P}(\alpha Ax)$ with parameter $\alpha > 0$. The corresponding negative log-likelihood is the Kullback-Leibler divergence

$$f(x) = \sum_{i=1}^{m} y_i \log\left(\frac{y_i}{\alpha(Ax)_i}\right) + \alpha(Ax)_i - y_i. \tag{6.1}$$

So far in this manuscript, given an observation model and its associated data-fidelity term, we proposed PnP and RED denoising priors for regularizing the inverse problem with an explicit prior. Moreover, when using deep denoisers corresponding to exact gradient-descent steps (Chapter 4) or proximal mappings (Chapter 5), we saw that RED and PnP methods become real first order proximal optimization algorithms with known convergence guarantees. A remarkable property of these priors is that they are decoupled from the degradation model represented by f in the sense that one learned prior can

serve as a regularizer for many inverse problems. However, for Poisson noise the data-fidelity term (6.1) is neither smooth with a Lipschitz gradient, nor proximable for $A \neq \mathrm{Id}$ (meaning that its proximal operator cannot be computed in a closed form), which limits the use of proximal algorithms for minimizing a variational objective of the form $\lambda f + g$. Different method have been proposed to address this limitation. PIDAL (Figueiredo and Bioucas-Dias, 2009, 2010) and related methods (Ng et al., 2010; Setzer et al., 2010) solve (6.1) using modified versions of ADMM. As f is proximable when $A = \mathrm{Id}$, the idea is to add a supplementary constraint in the minimization problem and to adopt an augmented Lagrangian framework. Similarly, (Boulanger et al., 2018) adopt the primal-dual algorithm (2.71) which also splits A from the f update. However, no convergence is established for nonconvex regularization.

In a different direction, Bauschke et al. (2017) address the minimization of objectives of the form $f + g$ by introducing a Proximal Gradient Descent (PGD) algorithm in the Bregman divergence paradigm, called Bregman Proximal Gradient (BPG). The benefit of BPG is that the smoothness condition on f for sufficient decrease of PGD is replaced by the "NoLip" condition "$Lh - f$ is convex" for a convex potential h. For instance, Bauschke et al. (2017) show that the data-fidelity term (6.1) satisfies the NoLip condition for the Burg's entropy $h(x) = -\sum_{i=1}^{n} \log(x_i)$. In the nonconvex setting, Bolte et al. (2018) prove global convergence of the algorithm to a critical point of the objective. The analysis however requires assumptions that are not verified by the Poisson data-fidelity term and Burg's entropy Bregman potential. Bregman optimization had been previously introduced and analyzed by Censor and Zenios (1992); Chen and Teboulle (1993); Eckstein (1993).

Our primary goal is to exploit the BPG algorithm for regularizing inverse problems using PnP and RED priors. A first Bregman adaptation for PnP and RED was proposed by Al-Shabili et al. (2022) but without any theoretical convergence analysis. Moreover, the interaction between the data-fidelity, the Bregman potential, and the denoiser was not explored. Our analysis is built on the interpretation of the plugged denoiser as a Bregman proximal operator.

$$\mathrm{Prox}_{g}^{h}(y) = \arg\min_{x} g(x) + D_h(x, y), \tag{6.2}$$

where $D_h(x, y)$ is the Bregman divergence associated with h. In Section 6.1, we show that by selecting a suitable noise model, other than the traditional Gaussian one, the MAP denoiser can be expressed as a Bregman proximal operator. Remarkably, the corresponding noise distribution belongs to an exponential family, which allows for a closed-form posterior mean (MMSE) denoiser generalizing the Tweedie's formula. Using this interpretation, we derive a RED prior tailored to the Bregman geometry. By presenting a prior compatible with the noise model, we highlight the limitation of the decoupling between prior and data-fidelity suggested in the existing PnP literature.

In order to safely use our MAP and MMSE denoisers in the BPG method, we introduce the Bregman Score denoiser, which generalizes the Gradient Step denoiser from Chapter 4. Our denoiser provides an approximation of the log prior of the noisy distribution of images. Moreover, based on the characterization of the Bregman Proximal operator from (Gribonval and Nikolova, 2020), we show under mild conditions that our denoiser can be expressed as the Bregman proximal operator of an explicit nonconvex potential. In Section 6.2, we review the BPG algorithm from (Bauschke et al., 2017) and prove new nonconvex convergence results, extending the analysis from (Bolte et al., 2018). In Section 6.3, we use the Bregman Score Denoiser within the BPG algorithm and propose a Bregman extension of RED and PnP algorithms. Elaborating on the results from (Bolte et al., 2018) for the BPG algorithm in the nonconvex setting, we demonstrate that these algorithms are

guaranteed to converge towards stationary points of an explicit functional. We finally show in Section 6.4 the relevance of the proposed framework in the context of Poisson inverse problems.

The results from this chapter were published in (Hurault et al., 2023b) and the code is available at `https://github.com/samuro95/BregmanPnP`.

6.1 Bregman denoising prior

The overall objective of this chapter is to efficiently solve ill-posed image restoration (IR) problems involving a data-fidelity term f verifying the NoLip assumption for some convex potential h

$$\textbf{NoLip} \quad \text{There is } L > 0 \text{ such that } Lh - f \text{ is convex on } int\,\mathrm{dom}\, h. \tag{6.3}$$

PnP provides an elegant framework for solving ill-posed inverse problems with a denoising prior. However, the intuition and efficiency of PnP methods inherit from the fact that Gaussian noise is well suited for the Euclidean L^2 distance, the latter naturally arising in the MAP formulation of the Gaussian denoising problem. When the Euclidean distance is replaced by a more general Bregman divergence, the noise model needs to be adapted accordingly for the prior.

In Section 6.1.1, we first discuss the choice of the noise model associated to a Bregman divergence, leading to Bregman formulations of the MAP and MMSE estimators. Then we introduce in Section 6.1.2 the Bregman Score Denoiser that will be used to regularize inverse problems.

Notation For convenience, we assume throughout our analysis that the convex potential $h : C_h \subseteq \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is $\mathcal{C}^2$ and of Legendre type

Definition 10 (Legendre function, Rockafellar (1997))**.** *Let $h : C \subseteq \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a proper lower semi-continuous convex function. It is called:*

(i) essentially smooth, if h is differentiable on $\mathrm{int}\,\mathrm{dom}(h)$*, with moreover* $\left|\left|\nabla h(x^k)\right|\right| \to \infty$ *for every sequence* $(x^k)_{k\in\mathbb{N}}$ *of* $\mathrm{int}\,\mathrm{dom}(h)$ *converging towards a boundary point of* $\mathrm{dom}(h)$*.*

(ii) of Legendre type if h is essentially smooth and strictly convex on $\mathrm{int}\,\mathrm{dom}(h)$*.*

We also recall the following properties of Legendre functions, which are used without justification in our analysis (see (Rockafellar, 1997, Section 26) for more details):

- h is of Legendre type if and only if its convex conjugate h^* is of Legendre type.
- For h of Legendre type, $dom(\nabla h) = \mathrm{int}\,\mathrm{dom}(h)$, ∇h is a bijection from $\mathrm{int}\,\mathrm{dom}(h)$ to $\mathrm{int}\,\mathrm{dom}(h^*)$ and $(\nabla h)^{-1} = \nabla h^*$.

From h of Legendre type, $D_h(x, y)$ denotes its associated Bregman divergence

$$D_h : \mathbb{R}^n \times int\,\mathrm{dom}\, h \to [0, +\infty] : (x, y) \to \begin{cases} h(x) - h(y) - \langle \nabla h(y), x - y \rangle & \text{if } x \in \mathrm{dom}(h) \\ +\infty & \text{otherwise.} \end{cases} \tag{6.4}$$

Examples We are interested in the two following Legendre functions

- L^2 potential : $h(x) = \frac{1}{2}||x||^2$, $C = \mathrm{dom}(h) = \mathrm{dom}(h^*) = \mathbb{R}^n$, $D_h(x,y) = \frac{1}{2}||x-y||^2$
- Burg's entropy : $h(x) = -\sum_{i=1}^n \log(x_i)$, $C = \mathrm{dom}(h) = \mathbb{R}^n_{++}$, $\mathrm{dom}(h^*) = \mathbb{R}^n_{--}$ and $D_h(x,y) = \sum_{i=1}^n \frac{x_i}{y_i} - \log\left(\frac{x_i}{y_i}\right) - 1$.

6.1.1 Bregman noise model

We consider the following observation noise model, referred to as *Bregman noise*[1],

$$\text{for } x, y \in \mathrm{dom}(h) \times \mathrm{int}\,\mathrm{dom}(h) \qquad p_{Y|X}(y|x) := \exp\left(-\gamma D_h(x,y) + \rho(x)\right). \tag{6.5}$$

We assume that there is $\gamma > 0$ and a normalizing function $\rho : \mathrm{dom}(h) \to \mathbb{R}$ such that the expression (6.5) defines a probability measure. For instance, for $h(x) = \frac{1}{2}||x||^2$, $\gamma = \frac{1}{\sigma^2}$ and $\rho = 0$, we retrieve the Gaussian noise model with variance σ^2. As we will show in Section 6.4, for h given by Burg's entropy, $p_{Y|X}$ corresponds to a multivariate Inverse Gamma ($\mathcal{IG}$) distribution.

Given a noisy observation $y \in \mathrm{int}\,\mathrm{dom}(h)$, *i.e.* a realization of a random variable Y with conditional probability $p_{Y|X}$, we now reconsider, the two optimal MAP and MMSE denoising estimators, that were derived in Section 2.4.1 for the specific context of Gaussian noise.

Maximum-a-posteriori (MAP) estimator The MAP denoiser selects the mode of the a-posteriori probability distribution $p_{X|Y}$. Given the prior p_X, it writes

$$\hat{x}_{MAP}(y) = \arg\min_x -\log p_{X|Y}(x|y) = \arg\min_x -\log p_X(x) - \log p_{Y|X}(y|x) \tag{6.6}$$

$$= \mathrm{Prox}^h_{-\frac{1}{\gamma}(\rho + \log p_X)}(y). \tag{6.7}$$

Under the Bregman noise model (6.5), the MAP denoiser writes as the *Bregman proximal operator* (see relation (6.2)) of $-\frac{1}{\gamma}(\log p_X + \rho)$. This acknowledges for the fact that the introduced Bregman noise is the adequate noise model for generalizing PnP methods within the Bregman framework.

Posterior mean (MMSE) estimator The MMSE denoiser is the expected value of the posterior probability distribution and the optimal Bayes estimator for the L^2 score. Note that our Bregman noise conditional probability (6.5) belongs to the regular *exponential family of distributions*

$$p_{Y|X}(y|x) = p_0(y)\exp\left(\langle x, T(y)\rangle - \psi(x)\right) \tag{6.8}$$

with $T(y) = \gamma\nabla h(y)$, $\psi(x) = \gamma h(x) - \rho(x)$ and $p_0(y) = \exp\left(\gamma h(y) - \gamma\langle \nabla h(y), y\rangle\right)$. Indeed, we have

$$\begin{aligned} p_{Y|X}(y|x) &= p_0(y)\exp\left(\langle x, T(y)\rangle - \psi(x)\right) \\ &= \exp\left(\gamma h(y) - \gamma\langle \nabla h(y), y\rangle\right)\exp\left(\gamma\langle x, \nabla h(y)\rangle - \gamma h(x) + \rho(x)\right) \\ &= \exp\left(-\gamma(h(x) - h(y) - \langle \nabla h(y), x - y\rangle) + \rho(x)\right) \\ &= \exp\left(-\gamma D_h(x,y) + \rho(x)\right). \end{aligned} \tag{6.9}$$

[1]The Bregman divergence being non-symmetric, the order of the variables (x, y) in D_h is important. Distributions of the form (6.5) with reverse order in D_h have been characterized in (Banerjee et al., 2005) but this analysis does not apply here.

It is shown in (Efron, 2011) (for $T = \mathrm{Id}$ and generalized in (Kim and Ye, 2021) for $T \neq \mathrm{Id}$) that the corresponding posterior mean estimator verifies a generalized Tweedie formula

$$\nabla T(y).\hat{x}_{MMSE}(y) = -\nabla \log p_0(y) + \nabla \log p_Y(y). \tag{6.10}$$

As $T = \nabla h$ and using

$$\nabla \log p_0(y) = \gamma \nabla h(y) - \gamma \nabla h(y) - \gamma \nabla^2 h(y).y = -\gamma \nabla^2 h(y).y, \tag{6.11}$$

we get

$$\gamma \nabla^2 h(y).\hat{x}_{MMSE}(y) = \gamma \nabla^2 h(y).y + \nabla \log p_Y(y). \tag{6.12}$$

As h is strictly convex, $\nabla^2 h(y)$ is invertible and

$$\hat{x}_{MMSE}(y) = y + \frac{1}{\gamma}(\nabla^2 h(y))^{-1}.\nabla \log p_Y(y). \tag{6.13}$$

Note that for the Gaussian noise model, we have $h(x) = \frac{1}{2}||x||^2$, $\gamma = 1/\sigma^2$ and (6.13) falls back to the more classical Tweedie formula (2.74) of the Gaussian posterior mean denoiser $\hat{x}_{MMSE}(y) = y + \sigma^2 \nabla \log p_\sigma(y)$.

Moreover, in the specific Euclidean L^2 case, we explained in Section 2.4.1 that with the **Denoising Score Matching** (DSM) result from (Vincent, 2011), training a deep denoiser by minimizing the denoising L^2 comes back exactly to approximating the MMSE denoiser. We now prove the same result for a general Bregman noise model.

Proposition 26. *Denoting the denoiser $\mathcal{B}_\gamma^\theta$ with training parameters θ, with the noise conditional probability* (6.5)*, we have:*

$$\mathbb{E}_{y\sim p_Y(y)}\left[\frac{1}{2}\left|\left|\mathcal{B}_\gamma^\theta(y) - \hat{x}_{MMSE}(y)\right|\right|^2\right] = \mathbb{E}_{(x,y)\sim p_{X,Y}}\left[\frac{1}{2}\left|\left|\mathcal{B}_\gamma^\theta(y) - x\right|\right|^2\right] + const \tag{6.14}$$

where const denotes a constant w.r.t. θ. Minimizing w.r.t. θ the left term, which we call Posterior Mean Matching (PMM) comes back to minimizing the right term, referred to Denoising Score Matching (DSM) in Vincent (2011).

Proof. Departing from the PMM term and using Tweedie's formula (6.13)

$$PMM(\theta) = \mathbb{E}_{y\sim p_Y(y)}\left[\frac{1}{2}\left|\left|\mathcal{B}_\gamma^\theta(y) - \hat{x}_{MMSE}(y)\right|\right|^2\right] \tag{6.15}$$

$$= \mathbb{E}_{y\sim p_Y(y)}\left[\frac{1}{2}\left|\left|\mathcal{B}_\gamma^\theta(y) - y - \frac{1}{\gamma}(\nabla^2 h(y))^{-1}.\nabla \log p_Y(y)\right|\right|^2\right]. \tag{6.16}$$

We set

$$s_\gamma^\theta(y) := \gamma \nabla^2 h(y).\left(\mathcal{B}_\gamma^\theta(y) - y\right) \tag{6.17}$$

so that

$$PMM(\theta) = \mathbb{E}_{y\sim p_Y(y)}\left[\frac{1}{2\gamma^2}\left|\left|(\nabla^2 h(y))^{-1}.\left(s_\gamma^\theta(y) - \nabla \log p_Y(y)\right)\right|\right|^2\right]. \tag{6.18}$$

The rest of the proof follows the same arguments as Vincent (2011). We can develop (6.18) as

$$PMM(\theta) = \mathbb{E}_{y\sim p_Y(y)}\left[\frac{1}{2\gamma^2}\left|\left|(\nabla^2 h(y))^{-1} s_\gamma^\theta(y)\right|\right|^2\right] - S(\theta) + \text{const} \tag{6.19}$$

with

$$S(\theta) = \mathbb{E}_{y\sim p_Y(y)}\left[\frac{1}{2\gamma^2}\Big\langle(\nabla^2 h(y))^{-1}.s_\gamma^\theta(y), (\nabla^2 h(y))^{-1}.\nabla \log p_Y(y)\Big\rangle\right] \tag{6.20}$$

$$= \frac{1}{2\gamma^2}\int_y \Big\langle(\nabla^2 h(y))^{-1}.s_\gamma^\theta(y), (\nabla^2 h(y))^{-1}.\frac{\nabla p_Y(y)}{p_Y(y)}\Big\rangle p_Y(y)dy \tag{6.21}$$

$$= \frac{1}{2\gamma^2}\int_y \Big\langle(\nabla^2 h(y))^{-1}.s_\gamma^\theta(y), (\nabla^2 h(y))^{-1}.\nabla p_Y(y)\Big\rangle dy. \tag{6.22}$$

Using Bayes formula

$$\nabla p_Y(y) = \nabla_y \int_x p_{Y|X}(y|x)p_X(x)dx \tag{6.23}$$

$$= \int_x p_X(x)\nabla_y p_{Y|X}(y|x)dx \tag{6.24}$$

$$= \int_x p_X(x)p_{Y|X}(y|x)\nabla_y \log p_{Y|X}(y|x)dx \tag{6.25}$$

$$= \int_x p_{X,Y}(x,y)\nabla_y \log p_{Y|X}(y|x)dx. \tag{6.26}$$

And thus we obtain

$$S(\theta) = \frac{1}{2\gamma^2}\int_y \Big\langle(\nabla^2 h(y))^{-1}.s_\gamma^\theta(y), (\nabla^2 h(y))^{-1}.\int_x p_{X,Y}(x,y)\nabla_y \log p_{Y|X}(y|x)dx\Big\rangle dy \tag{6.27}$$

$$= \frac{1}{2\gamma^2}\int_y\int_x \langle(\nabla^2 h(y))^{-1}.s_\gamma^\theta(y), (\nabla^2 h(y))^{-1}.\nabla_y \log p_{Y|X}(y|x)\Big\rangle p_{X,Y}(x,y)dxdy \tag{6.28}$$

$$= \mathbb{E}_{(x,y)\sim p_{X,Y}}\left[\frac{1}{2\gamma^2}\Big\langle(\nabla^2 h(y))^{-1}.s_\gamma^\theta(y), (\nabla^2 h(y))^{-1}.\nabla_y \log p_{Y|X}(y|x)\Big\rangle\right]. \tag{6.29}$$

Coming back to (6.19),

$$\begin{aligned} PMM(\theta) = {} & \mathbb{E}_{y\sim p_Y(y)}\left[\frac{1}{2\gamma^2}\left|\left|(\nabla^2 h(y))^{-1}s_\gamma^\theta(y)\right|\right|^2\right] \\ & - \mathbb{E}_{(x,y)\sim p_{X,Y}}\left[\frac{1}{2\gamma^2}\Big\langle(\nabla^2 h(y))^{-1}.s_\gamma^\theta(y), (\nabla^2 h(y))^{-1}.\nabla_y \log p_{Y|X}(y|x)\Big\rangle\right] \\ & + \text{const} \end{aligned} \tag{6.30}$$

Using the fact that

$$\mathbb{E}_{y\sim p_Y(y)}\left[\frac{1}{2\gamma^2}\left|\left|(\nabla^2 h(y))^{-1}s_\gamma^\theta(y)\right|\right|^2\right] = \mathbb{E}_{(x,y)\sim p_{X,Y}}\left[\frac{1}{2\gamma^2}\left|\left|(\nabla^2 h(y))^{-1}s_\gamma^\theta(y)\right|\right|^2\right] \tag{6.31}$$

and factorizing, we get

$$\begin{aligned} PMM(\theta) &= \mathbb{E}_{(x,y)\sim p_{X,Y}}\left[\frac{1}{2\gamma^2}\left|\left|(\nabla^2 h(y))^{-1}.\left(s_\gamma^\theta(y) - \nabla_y \log p_{Y|X}(y|x)\right)\right|\right|^2\right] + \text{const} \\ &= \mathbb{E}_{(x,y)\sim p_{X,Y}}\left[\left|\left|\mathcal{B}_\gamma^\theta(y) - y + \frac{1}{\gamma}(\nabla^2 h(y))^{-1}.\nabla_y \log p_{Y|X}(y|x)\right|\right|^2\right] + \text{const.} \end{aligned} \tag{6.32}$$

$$\tag{6.33}$$

Now from the Bregman noise model (6.5),

$$\nabla_y \log p_{Y|X}(y|x) = -\gamma \nabla_y D_h(x, y) \tag{6.34}$$
$$= \gamma \nabla^2 h(y).(y - x). \tag{6.35}$$

And thus

$$PMM(\theta) = \mathbb{E}_{(x,y)\sim p_{X,Y}} \left[\left|\left| \mathcal{B}_\gamma^\theta(y) - x \right|\right|^2 \right] + \text{const.} \tag{6.36}$$

□

Therefore, with this Proposition and the Tweedie formula verified by the MMSE denoiser, we get: given an off-the-shelf "Bregman denoiser" $\mathcal{B}_\gamma$ specifically devised to remove Bregman noise (6.5) of level γ, if the denoiser is trained with the L^2 loss, then it provides an approximation of the score

$$-\nabla \log p_Y(y) \approx \gamma \nabla^2 h(y). \left(y - \mathcal{B}_\gamma(y)\right). \tag{6.37}$$

6.1.2 Bregman Score Denoiser

We propose to define a denoiser following the form of the MMSE (6.13)

$$\mathcal{B}_\gamma(y) = y - (\nabla^2 h(y))^{-1} \cdot \nabla g_\gamma(y), \tag{6.38}$$

with $g_\gamma : \mathbb{R}^n \to \mathbb{R}$ a nonconvex potential parameterized by a neural network. The denoiser takes the form of a step of Natural Gradient-Descent (NGD) (Amari, 1998). The term $(\nabla^2 h(y))^{-1} \cdot \nabla g_\gamma(y)$ corresponds to the steepest direction of g_γ in the Riemannian metric defined by the Hessian of h. This denoiser is a Bregman generalization of the Gaussian noise Gradient-Step denoiser studied in the Chapter 4.

Using the MMSE Tweedie formula (6.13) and the Denoising Score Matching generalization derived in Proposition 26, training the denoiser (6.13) with the Bregman noise (6.5) by minimizing the L^2 loss comes back to minimize

$$\mathbb{E}_{y\sim p_Y(y)} \left[\left|\left| (\nabla^2 h(y))^{-1}. \left(\nabla g_\gamma(y) - \frac{1}{\gamma} \nabla \log p_Y(y) \right) \right|\right|^2 \right]. \tag{6.39}$$

We get at optimality $\nabla g_\gamma \approx -\frac{1}{\gamma} \nabla \log p_Y$, *i.e.* the score is properly approximated with an explicit conservative vector field. We refer to this denoiser as the *Bregman Score denoiser*.

Is the Bregman Score Denoiser a Bregman proximal operator?

Using the characterization of Euclidean proximity operator from Theorem 3, we proved in Chapter 5 that the Gaussian noise Gradient-Step denoiser can be the proximal operator of a nonconvex potential. We want to generalize this property to our Bregman Score denoiser (6.38).

A characterization of Bregman proximity operators A first step to achieve this result is to extend Theorem 3 in the Bregman framework. Remind that in Theorem 3, it was proven (by Gribonval and Nikolova (2020)) that an operator $T : \mathbb{R}^n \to \mathbb{R}^n$ is a Euclidean proximity operator if and only if there is a convex function ψ such that for all x, $T(x) \in \partial\psi(x)$. In the following theorem, we generalize this result for T a Bregman proximity operator, where we define the Bregman proximity operator as (6.2).

Theorem 23. *Let h of Legendre type on $\mathbb{R}^n$. Let $T : \operatorname{int}\operatorname{dom}(h) \to \mathbb{R}^n$. The following properties are equivalent:*

(i) There is $\phi : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ such that for each $y \in \operatorname{int}\operatorname{dom}(h)$

$$T(y) \in \underset{x\in\mathbb{R}^n}{\arg\min}\{D_h(x, y) + \phi(x)\}. \tag{6.40}$$

(ii) There is a l.s.c $\psi : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$, convex on $\operatorname{int}\operatorname{dom}(h^*)$ such that $T(\nabla h^*(z)) \in \partial\psi(z)$ for each $z \in \operatorname{int}\operatorname{dom}(h^*)$.

When they hold, ϕ (resp. ψ) can be chosen given ψ (resp. ϕ) with $\forall z \in \operatorname{int}\operatorname{dom}(h^*)$

$$\psi(z) = \langle T(\nabla h^*(z)), z\rangle - h(T(\nabla h^*(z))) - \phi(T(\nabla h^*(z))) \tag{6.41}$$

or equivalently, $\forall y \in \operatorname{int}\operatorname{dom}(h)$

$$\psi(\nabla h(y)) = \langle T(y), \nabla h(y)\rangle - h(T(y)) - \phi(T(y)). \tag{6.42}$$

To summarize, $T : \mathbb{R}^n \to \mathbb{R}^n$ is a Bregman proximity operator (with potential h) if and only if there is a convex function ψ such that for all x, $T(\nabla h^* x) \in \partial\psi(x)$. Note that, in the Euclidean case $h(x) = \frac{1}{2}||x||^2$, we retrieve the result from Theorem 3.

Remark 23. *Theorem 23 is an extension of (Gribonval and Nikolova, 2020, Corollary 5 a)) when h is strictly convex and thus ∇h invertible. In this corollary, the authors state that (i) is equivalent to*

(iii) There is a l.s.c $g : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ convex such that $\nabla h(T^{-1}(x)) \in \partial g(x)$ for each $x \in \operatorname{Im}(T)$.

However, (ii) and (iii) are not equivalent. We show here that (ii) implies (iii) but the converse is not true. Let ψ convex, defined from (ii). Thanks to the Legendre-Fenchel identity, we have that

$$\begin{aligned}
&\forall z \in \operatorname{int}\operatorname{dom}(h^*),\ T(\nabla h^*(z)) \in \partial\psi(z) && (6.43)\\
\Leftrightarrow\ &\forall z \in \operatorname{int}\operatorname{dom}(h^*),\ z \in \partial\psi^*(T(\nabla h^*(z))) && (6.44)\\
\Leftrightarrow\ &\forall y \in \operatorname{int}\operatorname{dom}(h),\ \nabla h(y) \in \partial\psi^*(T(y)) && (6.45)\\
\Rightarrow\ &\forall x \in \operatorname{Im}(T),\ \nabla h(T^{-1}(x)) \in \partial\psi^*(x). && (6.46)
\end{aligned}$$

Therefore (ii) implies (iii) with $g = \psi^*$. However, the last line is just an implication.

Remark 24. *The Bregman divergence being non-symmetric, the Bregman proximity operator could also be defined with the Bregman divergence in the other order* i.e. $\arg\min_{x\in\mathbb{R}^n}\{D_h(y, x) + \phi(x)\}$. *As done in Gribonval and Nikolova (2020), a characterization can also be derived in this order but this out of the scope of this work.*

Proof. We now prove Theorem 23. We follow the same order of arguments than (Gribonval and Nikolova, 2020). We first prove, in the following Lemma, a general result reminiscent to (Gribonval and Nikolova, 2020, Theorem 3) for a general form of divergence function and then apply this result to Bregman divergences.

Lemma 8. *Let $a : \mathcal{Y} \subseteq \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$, $b : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$, $A : \mathcal{Y} \to \mathcal{Z}$ bijection from $\mathcal{Y}$ to $\mathcal{Z}$ (with $\mathcal{Z} \subset \mathbb{R}^n$). Consider $T : \mathcal{Y} \to \mathbb{R}^n$. Let $D(x, y) := a(y) - \langle x, A(y)\rangle + b(x)$. The following properties are equivalent:*

(i) There is $\phi : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ such that for each $y \in \mathcal{Y}$

$$T(y) \in \operatorname*{arg\,min}_{x \in \mathbb{R}^n} \{D(x, y) + \phi(x)\}. \tag{6.47}$$

(ii) There is a l.s.c $\psi : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ convex such that $T(A^{-1}(z)) \in \partial\psi(z)$ for each $z \in \mathcal{Z}$.

When they hold, ϕ (resp. ψ) can be chosen given ψ (resp. ϕ) with

$$\forall z \in \mathcal{Z}, \quad \psi(z) = \langle T(A^{-1}(z)), z\rangle - b(T(A^{-1}(z))) - \phi(T(A^{-1}(z))). \tag{6.48}$$

Proof. We follow the same arguments as the proof of (Gribonval and Nikolova, 2020, Theorem 3(c)).

(i) ⇒ (ii) :

Define

$$\rho(z) = \begin{cases} \langle T(A^{-1}(z)), z\rangle - b(T(A^{-1}(z))) - \phi(T(A^{-1}(z))) & \text{if } z \in \mathcal{Z} \\ +\infty & \text{else.} \end{cases} \tag{6.49}$$

Let $z \in \mathcal{Z}$ and $y = A^{-1}(z)$. From (i), $T(y)$ is the global minimizer of $x \to D(x, y) + \phi(x)$ or of $x \to -\langle x, A(y)\rangle + b(x) + \phi(x)$ and $\forall z' \in \mathcal{Z}, y' = A^{-1}(z')$,

$$\begin{aligned} \rho(z') - \rho(z) &= \langle T(A^{-1}(z')), z'\rangle - b(T(A^{-1}(z'))) - \phi(T(A^{-1}(z'))) \\ &\quad - \langle T(A^{-1}(z)), z\rangle + b(T(A^{-1}(z))) + \phi(T(A^{-1}(z))) \\ &= \langle T(y'), A(y')\rangle - b(T(y')) - \phi(T(y')) \\ &\quad - \langle T(y), A(y)\rangle + b(T(y)) + \phi(T(y)) \\ &= \langle T(y), A(y') - A(y)\rangle + \langle T(y'), A(y')\rangle - b(T(y')) - \phi(T(y')) \\ &\quad - \langle T(y), A(y')\rangle + b(T(y)) + \phi(T(y)) \\ &\geq \langle T(y), A(y') - A(y)\rangle = \langle T(A^{-1}(z)), z' - z\rangle. \end{aligned} \tag{6.50}$$

By definition of the subdifferential, this shows that

$$T(A^{-1}(z)) \in \partial\rho(z). \tag{6.51}$$

Let $\tilde{\rho}$ be the lower convex envelope of ρ (pointwise supremum of all the convex l.s.c functions below ρ). $\tilde{\rho}$ is proper convex l.s.c. and $\forall z \in \mathcal{Z}, \partial\rho(z) \neq \emptyset$. By (Gribonval and Nikolova, 2020, Proposition 3), $\forall z \in \mathcal{Z}$, $\rho(z) = \tilde{\rho}(z)$ and $\partial\rho(z) = \partial\tilde{\rho}(z)$. Thus, for $\psi = \tilde{\rho}$, we get (ii).

(ii) ⇒ (i) :

Define $\eta : \mathcal{Y} \to \mathbb{R}$ by

$$\eta(y) := \langle T(y), A(y)\rangle - b(T(y)) - \psi(A(y)). \tag{6.52}$$

By (ii), $\forall z, z' \in \mathcal{Z}$,

$$\psi(z) - \psi(z') \geq \langle T(A^{-1}(z')), z - z'\rangle \tag{6.53}$$

which gives $\forall y, y' \in \mathcal{Y}$,

$$\psi(A(y)) - \psi(A(y')) \geq \langle T(y'), A(y) - A(y') \rangle. \tag{6.54}$$

This yields

$$\begin{aligned} \eta(y') - \eta(y) &= \langle T(y'), A(y') \rangle - b(T(y')) - \psi(A(y')) - \langle T(y), A(y) \rangle + b(T(y)) + \psi(A(y)) \\ &\geq \langle T(y') - T(y), A(y) \rangle - b(T(y')) + b(T(y)). \end{aligned} \tag{6.55}$$

We can define $\phi : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ obeying $\mathrm{dom}(\phi) = \mathrm{Im}(T)$ with

$$\phi(x) = \begin{cases} \eta(y) \text{ for } y \in T^{-1}(x) & \text{if } x \in \mathrm{Im}(T) \\ +\infty & \text{otherwise.} \end{cases} \tag{6.56}$$

For $x' \in \mathrm{Im}(T)$, $x' = T(y')$, using the previous inequality with η, we get

$$\begin{aligned} \phi(x') - \phi(T(y)) &= \phi(T(y')) - \phi(T(y)) \\ &= \eta(y') - \eta(y) \\ &\geq \langle T(y') - T(y), A(y) \rangle - b(T(y')) + b(T(y)) \\ &= \langle x' - T(y), A(y) \rangle - b(x') + b(T(y)), \end{aligned} \tag{6.57}$$

that is to say, $\forall x' \in \mathrm{Im}(T)$

$$\phi(x') + b(x') - \langle x', A(y) \rangle \geq \phi(T(y)) + b(T(y)) - \langle T(y), A(y) \rangle. \tag{6.58}$$

Given the definition of ϕ, this is also true for $x' \notin \mathrm{Im}(T)$. Adding $a(y)$ on both sides, we get the desired result. □

Theorem 23 is the specialization of the previous lemma with Bregman divergences. Given h of Legendre-type, the divergence $D(x, y) = a(y) - \langle x, A(y) \rangle + b(x)$ becomes the Bregman divergence $D_h(x, y)$ defined in (6.4) for $A(y) := \nabla h(y)$, $a(y) := \langle \nabla h(y), y \rangle - h(y)$ and $b(x) := h(x)$ if $x \in \mathrm{dom}(h)$ and $+\infty$ otherwise. □

Application to the Bregman Score Denoiser We now apply this characterization to prove that the proposed Bregman Score Denoiser (6.38) can explicitly write as the Bregman proximal operator of a nonconvex potential.

Proposition 27. *Let h be $\mathcal{C}^2$ and of Legendre type. Let $g_\gamma : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ $\mathcal{C}^2$ and $\mathcal{B}_\gamma(y) : \mathrm{int}\,\mathrm{dom}(h) \to \mathbb{R}^n$ defined from h and g_γ in (6.38). Assume $\mathrm{Im}(\mathcal{B}_\gamma) \subset \mathrm{int}\,\mathrm{dom}(h)$ and that J_{B_γ} the Jacobian of B_γ is positive definite on $\mathrm{int}\,\mathrm{dom}(h)$. Then for $\phi_\gamma : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ defined by*

$$\phi_\gamma(x) := \begin{cases} g_\gamma(y) - D_h(x, y) \quad \textit{for } y = \mathcal{B}_\gamma^{-1}(x) & \textit{if } \; x \in \mathrm{Im}(\mathcal{B}_\gamma) \\ +\infty & \textit{otherwise} \end{cases} \tag{6.59}$$

we have that for each $y \in \mathrm{int}\,\mathrm{dom}(h)$

$$\mathcal{B}_\gamma(y) \in \underset{x \in \mathbb{R}^n}{\arg\min} \{ D_h(x, y) + \phi_\gamma(x) \} \tag{6.60}$$

Remark 25. *This result is the Bregman generalization of Proposition 25 on the characterization of the Gradient Step denoiser as a proximity operator. The condition J_{B_γ} positive definite on* $\operatorname{int}\operatorname{dom}(h)$ *falls back to the condition ∇g_γ strictly nonexpansive in Proposition 25.*

Remark 26. *It is enough to assume J_{B_γ} positive semi-definite, to get the same result changing $y = \mathcal{B}_\gamma^{-1}(x)$ by $y \in \mathcal{B}_\gamma^{-1}(x)$ in the definition* (6.59) *of ϕ_γ. We nevertheless prefer to write it as above to ensure that ϕ_γ is real analytic when $\mathcal{B}_\gamma$ is real analytic (see Section 6.4.2).*

Proof. In order to apply Theorem 23, we define $\psi_\gamma : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ by

$$\psi_\gamma(y) = \begin{cases} -h(y) + \langle \nabla h(y), y \rangle - g_\gamma(y) & \text{if } y \in \operatorname{int}\operatorname{dom}(h) \\ +\infty & \text{otherwise.} \end{cases} \tag{6.61}$$

Eventually, Proposition 27 is an application of Theorem 23 with $T = \mathcal{B}_\gamma : \mathbb{R}^n \to \mathbb{R}^n$ defined on $\operatorname{int}\operatorname{dom}(h)$ by

$$\mathcal{B}_\gamma(y) = \nabla(\psi_\gamma \circ \nabla h^*) \circ \nabla h(y). \tag{6.62}$$

We verify that the above expression corresponds to the initial definition of $\mathcal{B}_\gamma$ from (6.38). Indeed, for $y \in \operatorname{int}\operatorname{dom}(h)$,

$$\begin{aligned} \mathcal{B}_\gamma(y) &= \nabla(\psi_\gamma \circ \nabla h^*) \circ \nabla h(y) && (6.63)\\ &= \nabla^2 h^*(\nabla h(y)).\nabla \psi_\gamma \circ \nabla h^* \circ \nabla h(y) && (6.64)\\ &= \nabla^2 h^*(\nabla h(y)).\nabla \psi_\gamma(y). && (6.65) \end{aligned}$$

h is assumed strictly convex on $\operatorname{int}\operatorname{dom}(h)$, for $y \in \operatorname{int}\operatorname{dom}(h)$, the Hessian of h, denoted as $\nabla^2 h(y)$ is invertible. By differentiating,

$$\nabla h^*(\nabla h(y)) = y \tag{6.66}$$

we get

$$\nabla^2 h^*(\nabla h(y)) = (\nabla^2 h(y))^{-1}, \tag{6.67}$$

so that

$$\mathcal{B}_\gamma(y) = (\nabla^2 h(y))^{-1}.\nabla \psi_\gamma(y). \tag{6.68}$$

With the definition (6.61), we directly get

$$\mathcal{B}_\gamma(y) = (\nabla^2 h(y))^{-1}.\nabla \psi_\gamma(y) = y - (\nabla^2 h(y))^{-1}.\nabla g_\gamma(y). \tag{6.69}$$

and we retrieve the definition of $\mathcal{B}_\gamma$ from (6.38).

For applying Theorem 23, we need to verify that $\psi_\gamma \circ \nabla h^*$ is convex on $\operatorname{int}\operatorname{dom}(h^*)$. We first calculate for $y \in \operatorname{int}\operatorname{dom}(h)$

$$\begin{aligned} J_{\mathcal{B}_\gamma}(y) &= \nabla\left(\nabla(\psi_\gamma \circ \nabla h^*) \circ \nabla h(y)\right) && (6.70)\\ &= \nabla^2 h(y).\nabla^2(\psi_\gamma \circ \nabla h^*)(\nabla h(y)). && (6.71) \end{aligned}$$

By Proposition 14 and bijectivity of ∇h^* between $\operatorname{int}\operatorname{dom}(h^*)$ and $\operatorname{int}\operatorname{dom}(h)$, we have

$$\begin{aligned} & \psi_\gamma \circ \nabla h^* \text{ strictly convex on } \operatorname{int}\operatorname{dom}(h^*) && (6.72)\\ \Leftrightarrow\ & \forall z \in \operatorname{int}\operatorname{dom}(h^*), \forall u \in \mathbb{R}^n,\ \ \langle \nabla^2(\psi_\gamma \circ \nabla h^*)(z)u, u\rangle > 0 && (6.73)\\ \Leftrightarrow\ & \forall y \in \operatorname{int}\operatorname{dom}(h), \forall u \in \mathbb{R}^n,\ \ \langle \nabla^2(\psi_\gamma \circ \nabla h^*)(\nabla h(y))u, u\rangle > 0. && (6.74) \end{aligned}$$

Now, using the fact that $\nabla^2 h(y)$ is positive definite, it follows

$$\begin{aligned}
&\Leftrightarrow \ \forall y \in \operatorname{int}\operatorname{dom}(h), \forall u \in \mathbb{R}^n, \ \ \langle \nabla^2 h(y).\nabla^2(\psi_\gamma \circ \nabla h^*)(\nabla h(y))u, u\rangle > 0 && (6.75)\\
&\Leftrightarrow \ \forall y \in \operatorname{int}\operatorname{dom}(h), \forall u \in \mathbb{R}^n, \ \ \langle J_{\mathcal{B}_\gamma}(y)u, u\rangle > 0 && (6.76)\\
&\Leftrightarrow \ J_{\mathcal{B}_\gamma} \text{ positive definite on } \operatorname{int}\operatorname{dom}(h). && (6.77)
\end{aligned}$$

Therefore, assuming the latter, from Theorem 23, we get that there is $\phi_\gamma : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ such that for each $y \in \operatorname{int}\operatorname{dom}(h)$

$$\mathcal{B}_\gamma(y) \in \arg\min\{D_h(x, y) + \phi_\gamma(x)\}. \tag{6.78}$$

Finally, Theorem 23 also indicates that ϕ_γ can be chosen given ψ_γ with

$$\phi_\gamma(x) = \langle \mathcal{B}_\gamma(y), \nabla h(y)\rangle - h(\mathcal{B}_\gamma(y)) - \psi_\gamma \circ \nabla h^*(\nabla h(y)) \ \text{ for } y \in \mathcal{B}_\gamma^{-1}(x). \tag{6.79}$$

As we assumed $J_{\mathcal{B}_\gamma}$ positive definite (and not only positive semi-definite), we get that this inverse is unique: $\forall y \in \operatorname{int}\operatorname{dom}(h)$,

$$\begin{aligned}
\phi_\gamma(\mathcal{B}_\gamma(y)) &= \langle \mathcal{B}_\gamma(y), \nabla h(y)\rangle - h(\mathcal{B}_\gamma(y)) - \psi_\gamma \circ \nabla h^*(\nabla h(y))\\
&= \langle \mathcal{B}_\gamma(y) - y, \nabla h(y)\rangle - h(\mathcal{B}_\gamma(y)) + h(y) + \langle y, \nabla h(y)\rangle - h(y) - \psi_\gamma(y)\\
&= -D_h(\mathcal{B}_\gamma(y), y) + \langle y, \nabla h(y)\rangle - h(y) - \psi_\gamma(y)\\
&= -D_h(\mathcal{B}_\gamma(y), y) + g_\gamma(y).
\end{aligned} \tag{6.80}$$

□

More details on the positive definite assumption Proposition 27 requires the positive definiteness of $J_{\mathcal{B}_\gamma}$ on $int\operatorname{dom}(h)$. As detailed in Lemma 2, in the Euclidean case $h(x) = \frac{1}{2}||x||^2$, the positive definiteness is verified by the true MMSE denoiser. We also show that in Section 6.4, that this is verified by the MMSE denoiser when h is Burg's entropy. We suppose that the positive definiteness of the Jacobian of the MMSE denoiser is true for any h Legendre function, but we still did not manage to prove it.

To conclude this section, the Bregman Score Denoiser provides, via *exact* gradient or proximal mapping, two distinct explicit nonconvex priors g_γ and ϕ_γ that can be used for regularizing image inverse problems in a plug-and-play fashion.

6.2 Bregman Proximal Gradient (BPG) algorithm

Let F and $\mathcal{R}$ be two proper, lower semi-continuous and lower-bounded functions with F of class $\mathcal{C}^1$ on $\operatorname{int}\operatorname{dom}(h)$. It is proposed in Bauschke et al. (2017) to minimize $\Psi = F + \mathcal{R}$ using the following BPG algorithm:

$$x^{k+1} \in T_\tau(x_k) = \underset{x\in\mathbb{R}^n}{\arg\min}\{\mathcal{R}(x) + \langle x - x^k, \nabla F(x^k)\rangle + \frac{1}{\tau} D_h(x, x^k)\}. \tag{6.81}$$

Recalling the general expression of proximal operators defined in relation (6.2), when $\nabla h(x_k) - \tau\nabla F(x_k) \in \operatorname{dom}(h^*)$, the previous iteration can be written as

$$x^{k+1} \in \operatorname{Prox}^h_{\tau\mathcal{R}} \circ \nabla h^*(\nabla h - \tau\nabla F)(x_k). \tag{6.82}$$

With formulation (6.82), the BPG algorithm generalizes the Proximal Gradient Descent (PGD) algorithm with a different geometry defined by h.

If F verifies the NoLip condition (6.3) for some $L > 0$ and if $\tau < \frac{1}{L}$, one can prove that the objective function Ψ decreases along the iterates (6.82). Global convergence of the iterates is also shown in (Bolte et al., 2018). However, they take assumptions on F and h that are not satisfied in the context of Poisson inverse problems. For instance, h is assumed strongly convex on the full domain $\mathbb{R}^n$, which is not satisfied by Burg's entropy. Additionally, F is assumed to have Lipschitz-gradient on bounded subsets of $\mathbb{R}^n$, which is not true for F the Poisson data-fidelity term (6.1). Following the structure of their proof, we extend in Proposition 28 and Theorem 24 the convergence theory from (Bolte et al., 2018) for a more general set of assumptions (Assumption 4) described below, which is verified for Poisson inverse problems. For the rest of the section, we take the following general assumptions.

Assumption 1.

(i) $h : \mathbb{R}^n \to \mathbb{R}$ *is of Legendre-type.*

(ii) $F : \mathbb{R}^n \to \mathbb{R}$ *is proper,* $\mathcal{C}^1$ *on* $\operatorname{int}\operatorname{dom}(h)$*, with* $\operatorname{dom}(h) \subset \operatorname{dom}(F)$*.*

(iii) $\mathcal{R} : \mathbb{R}^n \to \mathbb{R}$ *is proper, lower semi-continuous with* $\operatorname{dom}\mathcal{R} \cap \operatorname{int}\operatorname{dom}(h) \neq \emptyset$*.*

(iv) $\Psi = F + \mathcal{R}$ *is lower-bounded, coercive and verifies the Kurdyka-Łojasiewicz (KŁ) property.*

(v) For $x \in \operatorname{int}\operatorname{dom}(h)$*,* $T_\tau(x)$ *(defined in* (6.81)*) is nonempty and included in* $\operatorname{int}\operatorname{dom}(h)$*.*

Note that, since $\mathcal{R}$ is nonconvex, the mapping T_τ is not single-valued in general. Assumption 1(v) is required for the algorithm to be well-posed. As shown in (Bauschke et al., 2017; Bolte et al., 2018), one sufficient condition for $T_\tau(x) \neq \emptyset$ is the supercoercivity of the function $h + \lambda\mathcal{R}$ for all $\lambda > 0$, that is $\lim_{||x||\to+\infty} \frac{h(x)+\lambda\mathcal{R}(x)}{||x||} = +\infty$. As $T_\tau(x) \subset \operatorname{dom}(h)$, $T_\tau(x) \subset \operatorname{int}\operatorname{dom}(h)$ is true when $\operatorname{dom}(h)$ is open (which is the case for Burg's entropy for example).

The convergence of the BPG algorithm in the nonconvex setting is studied by the authors of (Bolte et al., 2018). Under the main assumption that $Lh - F$ is convex on $\operatorname{int}\operatorname{dom}(h)$, they show first the sufficient decrease property (and thus convergence) of the function values, and second, global convergence of the iterates. However, $Lh - F$ is not convex on the full domain of $\operatorname{int}\operatorname{dom}(h)$ but only on the compact subset $\operatorname{dom}(\mathcal{R})$. One can verify that all the iterates (6.81) belong to the convex set

$$\operatorname{Conv}(\operatorname{dom}\mathcal{R}) \cap \operatorname{int}\operatorname{dom}(h), \tag{6.83}$$

where $\operatorname{Conv}(E)$ stands for the convex envelope of E. We will show that it is enough to assume $Lh - F$ convex on this convex subset, and we take the following assumption.

Assumption 2. *There is* $L > 0$ *such that,* $Lh - F$ *is convex on* $\operatorname{Conv}(\operatorname{dom}\mathcal{R}) \cap \operatorname{int}\operatorname{dom}(h)$*.*

Equipped with this set of assumptions,, we can prove a result similar to (Bolte et al., 2018, Proposition 4.1).

Proposition 28. *Under Assumptions 1 and 2, let* $(x^k)_{k\in\mathbb{N}}$ *be a sequence generated by* (6.81) *with* $0 < \tau L < 1$*. Then the following properties hold*

(i) $(\Psi(x^k))_{k\in\mathbb{N}}$ *is non-increasing and converges.*

(ii) $\sum_k D_h(x^{k+1}, x^k) < \infty$ *and* $\min_{0\leq k\leq K} D_h(x^{k+1}, x^k) = O(1/K)$*.*

Proof. We adapt here the proof from (Bolte et al., 2018) to the case where $Lh - F$ is not globally convex but only convex on the convex subset $\operatorname{Conv}(\operatorname{dom}\mathcal{R}) \cap \operatorname{int}\operatorname{dom}(h)$.

Sufficient decrease property We first show that the sufficient decrease property of $\Psi(x_k)$ holds. This is true because the characteristic inequality of convex differentiable functions from Proposition 14 holds when convexity holds only on a subset.

Using Proposition 14, we have $Lh - F$ convex on C, if and only if, $\forall x, y \in C$, $D_{Lh-F}(x, y) \geq 0$, *i.e.* $D_F(x, y) \leq LD_h(x, y)$. The rest of the proof is identical to the one of (Bolte et al., 2018) and we recall it here for sake of completeness. Given the optimality conditions in (6.81), all the iterates $x_k \in C$ satisfy

$$\mathcal{R}(x^{k+1}) + \langle x^{k+1} - x^k, \nabla F(x_k)\rangle + \frac{1}{\tau} D_h(x^{k+1}, x^k) \leq \mathcal{R}(x^k). \tag{6.84}$$

Using $D_F(x, y) \leq LD_h(x, y)$, we get

$$\begin{aligned}(\mathcal{R}(x^{k+1}) + F(x^{k+1})) - (\mathcal{R}(x^k) + F(x^k)) &\leq -\frac{1}{\tau} D_h(x^{k+1}, x^k) + LD_h(x^{k+1}, x^k) \\ &= (L - \frac{1}{\tau}) D_h(x^{k+1}, x^k) \leq 0\end{aligned} \tag{6.85}$$

which, together with the fact that Ψ is lower bounded, proves (i). Summing the previous inequality from $k = 0$ to $K - 1$ gives

$$0 \leq \sum_{k=0}^{K-1} D_h(x^{k+1}, x^k) \leq \frac{\tau}{1 - \tau L}(\Psi(x_0) - \Psi(x_K)) \leq \frac{\tau}{1 - \tau L}(\Psi(x_0) - \inf_{x \in C} \Psi(x)) < +\infty. \tag{6.86}$$

Thus $(D_h(x^{k+1}, x^k))_k$ is summable and converges to 0 when $k \to +\infty$. Finally

$$\min_{0 \leq k \leq K} D_h(x^{k+1}, x^k) \leq \frac{1}{K+1} \sum_{k=0}^{K} D_h(x^{k+1}, x^k) \tag{6.87}$$

$$\leq \frac{1}{K+1} \frac{\tau}{1 - \tau L}(\Psi(x_0) - \inf_{x \in C} \Psi(x)). \tag{6.88}$$

□

Global convergence To prove global convergence of the iterates upon the Kurdyka-Łojasiewicz (KŁ) property, (Bolte et al., 2018, Theorem 4.1) is based on the hypotheses (a) $\mathrm{dom}(h) = \mathbb{R}^n$ and h is strongly convex on $\mathbb{R}^n$ and (b) ∇h and ∇F are Lipschitz continuous on any bounded subset of $\mathbb{R}^n$. These assumptions are not verified for h being the Burg's entropy $h(x) = -\sum \log(x_i)$ or F the Poisson data-fidelity term (6.1). Indeed, in that case, $\mathrm{dom}(h) = \mathbb{R}^n_{++}$, and h is strongly convex only on bounded sets. Moreover, F and h are not Lipschitz continuous near 0. However, thanks to the proven decrease of the iterates and as Ψ is assumed coercive, the iterates remain bounded. We can adopt the following weaker assumptions to ensure that the iterates do not tend to $+\infty$ or 0.

Assumption 3.

(i) h is strongly convex on any bounded convex subset of its domain.

(ii) For all $\alpha > 0$, ∇h and ∇F are Lipschitz continuous on $\{\Psi(x) \leq \alpha\}$.

Under these assumptions, we prove the equivalent of (Bolte et al., 2018, Theorem 4.1).

Theorem 24. *Under Assumptions 1, 2 and 3, the sequence* $(x^k)_{k\in\mathbb{N}}$ *generated by* (6.81) *with* $0 < \tau L < 1$ *converges to a critical point of* Ψ.

Proof. The proof of this theorem is a direct application of the abstract Theorem 4 on the convergence with the KŁ property. We need to verify that the sequence generated by our algorithm verifies the assumptions H1, H2 and H3 from Theorem 4. Then we can conclude on the single-point convergence using the fact that Ψ is KŁ.

H1:. From (6.85), we get

$$\Psi(x_k) - \Psi(x_{k+1}) \geq \left(\frac{1}{\tau} - L\right) D_h(x^{k+1}, x^k). \tag{6.89}$$

Besides, h is assumed to be strongly convex on any bounded convex subset of its domain. Furthermore, notice that $\mathrm{Conv}(C(x_0))\cap\mathrm{dom}(h)$ is a convex subset of $\mathrm{dom}(h)$ as intersection of convex sets. Therefore, there is $\sigma_h > 0$ such that

$$\forall x, y \in \mathrm{Conv}(C(x_0)) \cap \mathrm{dom}(h), \quad D_h(x, y) \geq \sigma_h \left|\left|x - y\right|\right|^2. \tag{6.90}$$

With the convention $D_h(x, y) = +\infty$ if $x \notin \mathrm{dom}(h)$ or $y \notin \mathrm{int}\,\mathrm{dom}(h)$,

$$\forall x, y \in \mathrm{Conv}(C(x_0)), \quad D_h(x, y) \geq \sigma_h \left|\left|x - y\right|\right|^2. \tag{6.91}$$

As $\forall k \geq 1, x_k \in C(x_0)$, we get

$$\Psi(x_k) - \Psi(x_{k+1}) \geq \sigma_h \left(\frac{1}{\tau} - L\right) \left|\left|x_{k+1} - x_k\right|\right|^2, \tag{6.92}$$

which proves (H1).

H2:. Given (6.81), the optimality condition for the update of x_{k+1} is

$$0 \in \partial\mathcal{R}(x^{k+1}) + \nabla F(x^k) + \frac{1}{\tau}(\nabla h(x^{k+1}) - \nabla h(x^k)). \tag{6.93}$$

For

$$\omega_{k+1} = \nabla F(x^{k+1}) - \nabla F(x^k) + \frac{1}{\tau}(\nabla h(x^k) - \nabla h(x^{k+1})) \tag{6.94}$$

we have

$$\omega_{k+1} \in \partial\Psi(x_{k+1}) = \partial\mathcal{R}(x^{k+1}) + \nabla F(x^{k+1}) \tag{6.95}$$

and

$$\left|\left|\omega_{k+1}\right|\right| \leq \left|\left|\nabla F(x^{k+1}) - \nabla F(x^k)\right|\right| + \frac{1}{\tau}\left|\left|\nabla h(x^k) - \nabla h(x^{k+1})\right|\right|. \tag{6.96}$$

By assumption, ∇F and ∇h are Lipschitz continuous on $C(x_0) = \{\Psi(x) < \Psi(x_0)\}$. As seen before, $\forall k \geq 1$, $x_k \in C(x_0)$. Thus, there is $b > 0$ such that

$$\left|\left|\omega_{k+1}\right|\right| \leq \left|\left|\nabla F(x^{k+1}) - \nabla F(x^k)\right|\right| + \frac{1}{\tau}\left|\left|\nabla h(x^k) - \nabla h(x^{k+1})\right|\right| \leq b\left|\left|x^{k+1} - x^k\right|\right|. \tag{6.97}$$

H3:. By coercivity of Ψ and decrease of the iterates $\Psi(x_k)$ (see Proposition 28), the iterates remain bounded. Let (x^{k_i}) be a subsequence converging towards x^*. Using the

optimality in the update of x_k we have

$$\mathcal{R}(x^k) + \langle x^k - x^{k-1}, \nabla F(x^{k-1})\rangle + \frac{1}{\tau} D_h(x^k, x^{k-1}) \tag{6.98}$$

$$\leq \mathcal{R}(x^*) + \langle x^* - x^{k-1}, \nabla F(x^{k-1})\rangle + \frac{1}{\tau} D_h(x^*, x^{k-1}) \tag{6.99}$$

$$\Leftrightarrow \mathcal{R}(x^k) \leq \mathcal{R}(x^*) + \langle x^* - x^{k-1}, \nabla F(x^{k-1})\rangle + \frac{1}{\tau} D_h(x^*, x^{k-1}) - \frac{1}{\tau} D_h(x^k, x^{k-1}). \tag{6.100}$$

From (6.92) and the fact that $(\Psi(x_k))_k$ converges, we have that $\left|\left|x^k - x^{k-1}\right|\right| \to 0$. Thus $(x^{k_i-1})_i$ also converges towards x^*. In addition, since h is continuously differentiable, $D_h(x^*, x^{k-1}) = h(x^*) - h(x^{k-1}) - \langle \nabla h(x^{k-1}), x^* - x^{k-1}\rangle \to 0$. Passing to the limit in (6.100), we get

$$\limsup_{i \to +\infty} \mathcal{R}(x^{k_i}) \leq \mathcal{R}(x^*). \tag{6.101}$$

By lower semicontinuity of $\mathcal{R}$ and continuity of F, we get the desired result:

$$\mathcal{R}(x^{k_i}) + F(x^{k_i}) \to \mathcal{R}(x^*) + F(x^*). \tag{6.102}$$

□

Backtracking The convergence actually requires controlling the NoLip constant. In order to avoid small stepsizes, we propose to adapt the backtracking strategy of (Beck, 2017, Chapter 10) to the BPG algorithm. Given $\gamma \in (0,1)$, $\eta \in [0,1)$ and an initial stepsize $\tau_0 > 0$, the following backtracking update rule on τ is applied at each iteration k:

$$\text{while} \quad \Psi(x_k) - \Psi(T_\tau(x_k)) < \frac{\gamma}{\tau} D_h(T_\tau(x_k), x_k), \quad \tau \longleftarrow \eta\tau. \tag{6.103}$$

Proposition 29. *At each iteration of the algorithm, the backtracking procedure (6.103) is finite and with backtracking, the convergence results of Proposition 28 and Theorem 24 still hold.*

Proof. For a given stepsize τ, we showed in equation (6.85) that

$$\Phi(x_k) - \Phi(T_\tau(x_k)) \geq \left(\frac{1}{\tau} - L\right) D_h(T_\tau(x_k), x_k). \tag{6.104}$$

Taking $\tau < \frac{1-\gamma}{L}$, we get $\frac{1}{\tau} - L > \frac{\gamma}{\tau}$ so that

$$\Phi(x_k) - \Phi(T_\tau(x_k)) > \frac{\gamma}{\tau} D_h(T_\tau(x_k), x_k). \tag{6.105}$$

Hence, when $\tau < \frac{1-\gamma}{L}$, the sufficient decrease condition is satisfied and the backtracking procedure ($\tau \longleftarrow \eta\tau$) must end. Replacing the former sufficient decrease condition (6.104) with (6.105), the rest of the proofs from Proposition 28 and Theorem 24 are identical. □

6.3 PnP and RED restoration with Bregman Score Denoiser

We now regularize inverse problems with the explicit prior provided by the Bregman Score Denoiser (6.38). We consider two variants of BPG for regularizing inverse problems

involving a data-fidelity term f satisfying some NoLip assumption. These algorithms respectively correspond to Bregman generalizations of the RED Gradient-Descent (2.103) and the Plug-and-Play PGD (2.109) algorithms. For the rest of this section, we consider the following assumptions.

Assumption 4.

(i) $h : C_h \to \mathbb{R} \cup \{+\infty\}$ *is of class* $\mathcal{C}^2$ *and of Legendre type.*

(ii) $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ *is proper, lower-bounded, coercive, of class* $\mathcal{C}^1$ *on* $\operatorname{int}\operatorname{dom}(h)$*, with* $\operatorname{dom}(h) \subset \operatorname{dom}(f)$*, and is subanalytic.*

(iii) ***NoLip :*** $L_f h - f$ *is convex on* $\operatorname{int}\operatorname{dom}(h)$*.*

(iv) h *is assumed strongly convex on any bounded convex subset of its domain and for all* $\alpha > 0$*,* ∇h *and* ∇f *are Lipschitz continuous on* $\{x \in \operatorname{dom}(h), \Psi(x) \leq \alpha\}$*.*

(v) g_γ *given by the Bregman Score Denoiser* (6.38) *and its associated* ϕ_γ *obtained from Proposition 27 are lower-bounded and subanalytic.*

Even though the Poisson data-fidelity term (6.1) is convex, our convergence results also hold for more general nonconvex data-fidelity terms. Assumption 4 (iv) generalizes (Bolte et al., 2018, Assumption D) and allows proving global convergence of the iterates. The subanalytic assumption of f, g_γ and ϕ_γ allows for the Kurdyka-Łojasiewicz (KŁ) property to be verified by the objective functions $f + g_\gamma$ and $f + \phi_\gamma$ (see Section 3.1.2 for more details).

6.3.1 Bregman Regularization-by-Denoising (B-RED)

We first generalize the RED Gradient-Descent (RED-GD) algorithm (4.24) in the Bregman framework. Classically, RED-GD is a simple gradient-descent algorithm applied to the functional $\lambda f + g_\gamma$ where the gradient ∇g_γ is assumed to be implicitly given by an image denoiser $\mathcal{B}_\gamma$ (parameterized by γ) via $\nabla g_\gamma = \operatorname{Id} - \mathcal{B}_\gamma$. Instead, our Bregman Score Denoiser (6.38) provides an explicit regularizing potential g_γ whose gradient approximates the score via the Tweedie formula (6.13). We propose to minimize $F_{\lambda,\gamma} = \lambda f + g_\gamma$ on $\operatorname{dom}(h)$ using the Bregman Gradient Descent algorithm

$$x_{k+1} = \nabla h^*(\nabla h - \tau \nabla F_{\lambda,\gamma})(x_k) \tag{6.106}$$

which also writes as the BPG algorithm (6.81) with $\mathcal{R} = 0$

$$x_{k+1} = \operatorname*{arg\,min}_{x \in \mathbb{R}^n} \{\langle x - x_k, \lambda \nabla f(x_k) + \nabla g_\gamma(x_k) \rangle + \frac{1}{\tau} D_h(x, x_k)\}. \tag{6.107}$$

As previously detailed, in the context of Poisson inverse problems, for h being Burg's entropy, $F_{\lambda,\gamma} = \lambda f + g_\gamma$ verifies the NoLip condition only on *bounded* convex subsets of $dom(h)$. Thus, we select C a non-empty closed bounded convex subset of $\overline{\operatorname{dom}(h)}$. For the algorithm (6.107) to be well-posed and to verify a sufficient decrease of $(F_{\lambda,\gamma}(x^k))$, the iterates need to verify $x_k \in C$. We propose to modify (6.107) as the Bregman version of

Projected Gradient Descent, which corresponds to the BPG algorithm (6.81) with $\mathcal{R} = i_C$, the characteristic function of the set C:

$$\textbf{(B-RED)} \qquad x^{k+1} \in T_\tau(x_k) = \operatorname*{arg\,min}_{x\in\mathbb{R}^n} \{i_C(x) + \langle x - x^k, \nabla F_{\lambda,\gamma}(x^k)\rangle + \frac{1}{\tau} D_h(x, x^k)\}. \tag{6.108}$$

For general convergence of B-RED, we need the following assumptions

Assumption 5.

(i) *C is a non-empty closed, bounded, convex and semi-algebraic subset of $\overline{\operatorname{dom}(h)}$ such that $C \cap \operatorname{int}\operatorname{dom}(h) \neq \emptyset$.*

(ii) *g_γ has Lipschitz continuous gradient on C and there is $L_\gamma > 0$ such that $L_\gamma h - g_\gamma$ is convex on $C \cap \operatorname{int}\operatorname{dom}(h)$.*

In (Bolte et al., 2018; Bauschke et al., 2017), the NoLip constant L needs to be known to set the stepsize of the BPG algorithm as $\tau L < 1$. In practice, the NoLip constant which depends on f, g_γ and C is either unknown or over-estimated. In order to avoid small stepsize, we adapt the backtracking strategy (exposed in the previous section) to automatically adjust the stepsize while keeping convergence guarantees. Using the general nonconvex convergence analysis of BPG realized in Section 6.2, we can show sufficient decrease of the objective and convergence of the iterates of B-RED.

Theorem 25. *Under Assumption 4 and Assumption 5, the iterates (x_k) given by the B-RED algorithm* (6.108) *with the backtracking procedure* (4.32) *verify*

(i) *$(F_{\lambda,\gamma}(x_k))$ is non-increasing and converges.*

(ii) *$\sum_k D_h(x^{k+1}, x^k) < \infty$ and $\min_{0\leq k\leq K} D_h(x^{k+1}, x^k) = O(1/K)$.*

(iii) *(x_k) converges to a critical point of $\Psi = i_C + F_{\lambda,\gamma}$.*

Proof. The B-RED algorithm corresponds to the BPG algorithm (6.81) with $F = F_{\lambda,\gamma} = \lambda f + g_\gamma$ and $\mathcal{R} = i_C$. Theorem 25 is a direct application of Proposition 29, that is to say, of the convergence results of Proposition 28 and Theorem 24 with backtracking. Given Assumptions 4 and 5, we verify that Assumptions 1, 2 and 3, required for Proposition 28 and Theorem 24, are verified:

Assumption 1. $\mathcal{R} = i_C$ verifies $\operatorname{dom}(\mathcal{R}) \cap \operatorname{int}\operatorname{dom}(h) = C \cap \operatorname{int}\operatorname{dom}(h) \neq \emptyset$. Moreover, $\mathcal{R}$ is semi-algebraic as the indicator function of a closed semi-algebraic set. g_γ and f being both assumed subanalytic and lower-bounded, by Lemma 5, $\lambda f + g_\sigma$ is then subanalytic (up to adding a constant to make f and g_σ non-negative). $\Psi = F_{\lambda,\gamma} + i_C$ is then also subanalytic, and thus KL. Ψ is also lower-bounded and coercive as f is lower-bounded and coercive and g_γ is lower-bounded. Finally, for $x \in \operatorname{int}\operatorname{dom}(h)$, $T_\tau(x)$ is non-empty as $h + \lambda i_C$ is supercoercive.

Assumption 2. By summing convex functions, using Assumption 4(iii) and Assumption 5(ii), $L = \lambda L_f + L_\gamma$ verifies $Lh - (\lambda f + g_\gamma)$ convex on $\operatorname{Conv}(\operatorname{dom}\mathcal{R}) \cap \operatorname{int}\operatorname{dom}(h) = C \cap \operatorname{int}\operatorname{dom}(h)$.

Assumption 3. As g_γ is assumed to have globally Lipschitz continuous gradient (Assumption 5(ii)), this follows directly from Assumption 4(iv).

□

6.3.2 Bregman Plug-and-Play (B-PnP)

We now consider the equivalent of PnP Proximal Gradient Descent algorithm in the Bregman framework. Given a denoiser $\mathcal{B}_\gamma$ with $\mathrm{Im}(\mathcal{B}_\gamma) \subset \mathrm{dom}(h)$ and $\lambda > 0$ such that $\mathrm{Im}(\nabla h - \lambda \nabla f) \subseteq \mathrm{dom}(\nabla h^*)$, it writes

$$\textbf{(B-PnP)} \qquad x^{k+1} = \mathcal{B}_\gamma \circ \nabla h^*(\nabla h - \lambda \nabla f)(x_k). \tag{6.109}$$

We use again as $\mathcal{B}_\gamma$ the Bregman Score Denoiser (6.38). Assuming that $J_{\mathcal{B}_\gamma}$ is positive definite on $\mathrm{int}\,\mathrm{dom}(h)$, Proposition 27 states that the Bregman Score denoiser $\mathcal{B}_\gamma$ is the Bregman proximal operator of some nonconvex potential ϕ_γ verifying (6.59). The algorithm B-PnP (6.109) then becomes

$$x^{k+1} \in \mathrm{Prox}^h_{\phi_\gamma} \circ \nabla h^*(\nabla h - \lambda \nabla f)(x_k), \tag{6.110}$$

which writes as a Bregman Proximal Gradient algorithm, with stepsize $\tau = 1$,

$$x^{k+1} \in \underset{x \in \mathbb{R}^n}{\arg\min} \{\phi_\gamma(x) + \langle x - x^k, \lambda \nabla f(x^k) \rangle + D_h(x, x^k)\}. \tag{6.111}$$

Similar to what happened in Chapter 5 in the Euclidean case, with Proposition 27, we have $\mathcal{B}_\gamma(y) \in \mathrm{Prox}^h_{\phi_\gamma}(y)$ *i.e.* a proximal step on ϕ_γ *with stepsize* 1. We are thus forced to keep a fixed stepsize $\tau = 1$ in the BPG algorithm (6.111) and no backtracking is possible. Using Section 6.2, we can show that B-PnP converges towards a stationary point of $\lambda f + \phi_\gamma$.

Theorem 26. *Assume Assumption 4 and $J_{\mathcal{B}_\gamma}$ positive definite on* $\mathrm{int}\,\mathrm{dom}(h)$. *Then for* $\mathrm{Im}(\nabla h - \lambda \nabla f) \subseteq \mathrm{dom}(\nabla h^*)$, $\mathrm{Im}(\mathcal{B}_\gamma) \subseteq \mathrm{dom}(h)$ *and* $\lambda L_f < 1$ *(with L_f specified in Assumption 4), the iterates (x_k) given by the B-PnP algorithm* (6.109) *verify*

(i) $((\lambda f + \phi_\gamma)(x_k))_k$ *is non-increasing and converges.*

(ii) $\sum_k D_h(x^{k+1}, x^k) < \infty$ *and* $\min_{0 \le k \le K} D_h(x^{k+1}, x^k) = O(1/K)$.

(iii) (x_k) *converges to a critical point of* $\lambda f + \phi_\gamma$.

Remark 27. *The condition* $\mathrm{Im}(\mathcal{B}_\gamma) \subseteq \mathrm{dom}(h)$ *and the required positive definiteness of $J_{\mathcal{B}_\gamma}$ come from Proposition 27 while the condition* $\mathrm{Im}(\nabla h - \lambda \nabla f) \subseteq \mathrm{dom}(\nabla h^*)$ *allows the algorithm B-PnP* (6.109) *to be well-posed. These assumptions will be discussed with more details in the context of Poisson image restoration in Section 6.4.*

Proof. It corresponds to the BPG algorithm (6.81) with $F = \lambda f$ and $\mathcal{R} = \phi_\gamma$. Theorem 26 is a direct application of the convergence results of Proposition 28 and Theorem 24. We now denote $\Psi = \lambda f + \psi_\gamma$. Given Assumptions 4, we verify that Assumptions 1, 2 and 3 are verified:

Assumption 1. For $\mathcal{R} = \phi_\gamma$, we have $Im(\mathcal{B}_\gamma) \subset \mathrm{dom}(\phi_\gamma)$ and as for $y \in \mathrm{int}\,\mathrm{dom}(h)$, $Im(\mathcal{B}_\gamma) \subset \mathrm{int}\,\mathrm{dom}(h)$ (by equation 6.60), we get $\mathrm{dom}(\mathcal{R}) \cap \mathrm{int}\,\mathrm{dom}(h) \neq \emptyset$. ϕ_γ and f being both assumed subanalytic and lower-bounded, by Lemma 5, $\lambda f + \phi_\sigma$ is then subanalytic (up to adding a constant to make f and ϕ_σ non-negative) and thus KŁ. Ψ is coercive because f is coercive and ϕ_γ is lower-bounded. Finally, $T_\tau(x)$ well-posed is ensured via Proposition 27.

Assumption 2. This is the NoLip property of f given by Assumption 4.

Assumption 3. This is directly given by Assumption 4(iv).

□

6.4 Application to Poisson inverse problems

We consider ill-posed inverse problems involving the Poisson data-fidelity term f introduced in (6.1). The Euclidean geometry (*i.e.* $h(x) = \frac{1}{2}||x||^2$) is not suitable for such f which does not have a Lipschitz gradient. In (Bauschke et al., 2017, Lemma 7), it is shown that an adequate Bregman potential h (in the sense that there exists L_f such that $L_f h - f$ is convex) for (6.1) is the Burg's entropy

$$h(x) = -\sum_{i=1}^{n} \log(x_i), \tag{6.112}$$

for which $\mathrm{dom}(h) = \mathbb{R}^n_{++}$ and $L_f h - f$ is convex on $\operatorname{int}\mathrm{dom}(h) = \mathbb{R}^n_{++}$ for $L_f \geq ||y||_1$. For further computation, note that the Burg's entropy (6.112) satisfies $\nabla h(x) = \nabla h^*(x) = -\frac{1}{x}$ and $\nabla^2 h(x) = \frac{1}{x^2}$.
The Bregman score denoiser associated to the Burg's entropy is presented in Section 6.4.1. The corresponding Bregman RED and PnP algorithms are applied to Poisson Image deblurring in Section 6.4.2.

6.4.1 Bregman Score Denoiser with Burg's entropy

We now specify the study of Section 6.1 to the case of the Burg's entropy (6.112). In this case, the Bregman noise model (6.5) writes for $x, y \in \mathbb{R}^n_{++}$ as

$$\begin{aligned} p(y|x) &= \exp\left(-\gamma D_h(x,y) + \rho(x)\right) \\ &= \exp(\rho(x)) \exp\Big(-\gamma(h(x) - h(y) - \langle \nabla h(y), x - y\rangle)\Big) \\ &= \exp(\rho(x)) \exp\left(-\gamma\Big(\sum_{i=1}^{n} -\log(x_i) + \log(y_i) - 1 + \frac{x_i}{y_i}\Big)\right) \\ &= \exp(\rho(x) + n\gamma) \prod_{i=1}^{n} \left(\frac{x_i}{y_i}\right)^{\gamma} \exp\left(-\gamma \frac{x_i}{y_i}\right). \end{aligned} \tag{6.113}$$

For $\gamma > 1$, this is a product of *Inverse Gamma* ($\mathcal{IG}(\alpha, \beta)$) distributions with parameters $\beta_i = \gamma x_i$ and $\alpha_i = \gamma - 1$. This noise model has mean (for $\gamma > 2$) $\frac{\gamma}{\gamma-2}x$ and variance (for $\gamma > 3$) $\frac{\gamma^2}{(\gamma-2)^2(\gamma-3)}x^2$. In particular, for large γ, the noise becomes centered on x with signal-dependent variance x^2/γ. We suppose $\gamma > 3$ for the rest of this section, such that $p(y|x)$ has finite second-order moment.

Furthermore, using Burg's entropy (6.112), the optimal posterior mean (6.13) and the Bregman Score Denoiser (6.38) respectively write, for $y \in \mathbb{R}^n_{++}$,

$$\hat{x}_{MMSE}(y) = y - \frac{1}{\gamma} y^2 \nabla(-\log p_Y)(y) \tag{6.114}$$

$$\mathcal{B}_\gamma(y) = y - y^2 \nabla g_\gamma(y). \tag{6.115}$$

Bregman Proximal Operator Considering Burg's entropy (6.112) in Proposition 27 we get that, if $J_{\mathcal{B}_\gamma}(y)$ is positive definite on $\operatorname{int}\mathrm{dom}(h) = \mathbb{R}^n_{++}$, the Bregman Score Denoiser (6.115) satisfies $\mathcal{B}_\gamma(y) = \mathrm{Prox}^h_{\phi_\gamma}$.

In order to apply the B-PnP algorithm, we need to verify the positive definiteness of $J_{\mathcal{B}_\gamma}(y)$. We here investigate the validity of the assumption. First, when specialized to

Burg's entropy, we show that it can be verified using the following condition on the deep potential g_γ.

Lemma 9.

$$\forall y \in \operatorname{int}\operatorname{dom}(h), J_{\mathcal{B}_\gamma}(y) \text{ positive definite} \tag{6.116}$$
$$\Leftrightarrow \ \forall y \in \operatorname{int}\operatorname{dom}(h), \quad \forall d \in \mathbb{R}^n, \quad \langle \nabla^2 J_{\mathcal{B}_\gamma}(y)d, d\rangle > 0 \tag{6.117}$$
$$\Leftrightarrow \ \forall y \in \mathbb{R}^n_{++}, \quad \forall d \in \mathbb{R}^n, \quad \langle y^4 \nabla^2 g_\gamma(y)d, d\rangle < \sum_{i=1}^n \left(y^2(1-2y\nabla g_\gamma(y))\right)_i d_i^2. \tag{6.118}$$

Proof. We now derive the calculations leading to the above equivalence. We saw in (6.70) that

$$J_{\mathcal{B}_\gamma}(y) = \nabla^2 h(y).\nabla^2(\psi_\gamma \circ \nabla h^*)(\nabla h(y)) \tag{6.119}$$

such that

$$\forall y \in \operatorname{int}\operatorname{dom}(h), \quad \forall d \in \mathbb{R}^n, \quad \langle \nabla^2 J_{\mathcal{B}_\gamma}(y)d, d\rangle > 0 \tag{6.120}$$
$$\Leftrightarrow \forall y \in \operatorname{int}\operatorname{dom}(h), \quad \forall d \in \mathbb{R}^n, \quad \langle \nabla^2(\psi_\gamma \circ \nabla h^*)(\nabla h(y))d, d\rangle > 0. \tag{6.121}$$

Differentiating $\eta_\gamma := \psi_\gamma \circ \nabla h^*$ twice gives

$$\nabla \eta_\gamma(x) = \nabla^2 h^*(x).\nabla \psi_\gamma(\nabla h^*(x)) \tag{6.122}$$

and

$$\nabla^2 \eta_\gamma(x) = (\nabla^2 h^*(x))^2.\nabla^2 \psi_\gamma(\nabla h^*(x)) + \nabla^3 h^*(x).\nabla \psi_\gamma(\nabla h^*(x)) \tag{6.123}$$

Evaluating on $\nabla h(y)$

$$\nabla^2 \eta_\gamma(\nabla h(y)) = (\nabla^2 h^*(\nabla h(y)))^2.\nabla^2 \psi_\gamma(y) + \nabla^3 h^*(\nabla h(y)).\nabla \psi_\gamma(y). \tag{6.124}$$

For Burg's entropy and $y \in \operatorname{int}\operatorname{dom}(h) = \mathbb{R}^n_{++}$,

$$\nabla^2 \eta_\gamma(\nabla h(y)) = y^4 \nabla^2 \psi_\gamma(y) + 2y^3 \mathrm{Diag}(\nabla \psi_\gamma(y)). \tag{6.125}$$

Using the definition (6.61) of ψ_γ specialized for Burg's entropy

$$\psi_\gamma(y) = -h(y) - g_\gamma(y) - 1, \tag{6.126}$$

$$\nabla \psi_\gamma(y) = -\nabla h(y) - \nabla g_\gamma(y) = \frac{1}{y} - \nabla g_\gamma(y) \tag{6.127}$$

and

$$\nabla^2 \psi_\gamma(y) = -\nabla^2 h(y) - \nabla^2 g_\gamma(y) = -\frac{1}{y^2} - \nabla^2 g_\gamma(y), \tag{6.128}$$

we get

$$\nabla^2 \eta_\gamma(\nabla h(y)) = y^2 \left(y^2 \nabla^2 \psi_\gamma(y) + 2y Diag(\nabla \psi_\gamma(y))\right) \tag{6.129}$$
$$= y^2 \left(-1 - y^2 \nabla^2 g_\gamma(y) + 2 - 2y\nabla g_\gamma(y)\right) \tag{6.130}$$

and the equivalence (6.116) follows. □

We now show that this condition holds for the MMSE denoiser.

Lemma 10. *For h Burg's entropy, and thus for the noise model* (6.113)*, the MMSE denoiser $\hat{x}_{MMSE}(y) = E[x|y]$ satisfies the condition "$J_{\mathcal{B}_\gamma}(y)$ is positive definite on* $\operatorname{int}\operatorname{dom}(h)$*" from Proposition 27.*

Proof. Recall that the MMSE denoiser corresponds to the Bregman Score Denoiser

$$\mathcal{B}_\gamma(y) = \nabla(\psi_\gamma \circ \nabla h^*) \circ \nabla h(y) = \nabla \eta_\gamma \circ \nabla h(y) \tag{6.131}$$

in the particular case where we have exactly

$$g_\gamma(y) = -\frac{1}{\gamma} \log p_Y(y). \tag{6.132}$$

Note that in the Euclidean case (i.e. for the L^2 Bregman divergence $h = \frac{1}{2}||.||^2$ and $p_Y = p_\sigma = p * \mathcal{N}(0, \sigma^2 \operatorname{Id})$), it is shown in Gribonval (2011) to be verified by the MMSE denoiser. We now show the same result for Burg's entropy Bregman potential. For $x \in \mathbb{R}^n_{++}$

$$h(x) = -\sum_{i=1}^{n} \log(x_i), \tag{6.133}$$

Recall that, in this case, the Bregman noise model writes

$$p(y|x) = \exp(\rho(x) + n\gamma) \prod_{i=1}^{n} \left(\frac{x_i}{y_i}\right)^\gamma \exp\left(-\gamma \frac{x_i}{y_i}\right) = \alpha(x) \prod_{i=1}^{n} (y_i)^{-\gamma} \exp\left(-\gamma \frac{x_i}{y_i}\right). \tag{6.134}$$

where $\alpha(x) = \exp(\rho(x) + n\gamma) \prod_{i=1}^{n} x_i^\gamma > 0$. We verify that $\forall y \in int\operatorname{dom}(h) = \mathbb{R}^n_{++}$

$$\langle \nabla^2 \eta_\gamma(\nabla h(y)) d, d \rangle > 0. \tag{6.135}$$

We showed in Lemma 9 that, $\forall y \in \mathbb{R}^n_{++}$

$$\nabla^2 \eta_\gamma(\nabla h(y)) = Diag\left(y^2(1 - 2y\nabla g_\gamma(y))\right) - y^4 \nabla^2 g_\gamma(y). \tag{6.136}$$

Similar to Gribonval (2011), for simplicity, we first write the proof in the single variable case ($n = 1$).

We have $\nabla g_\gamma(y) = \frac{-1}{\gamma} \frac{p'_Y(y)}{p_Y(y)}$ and $\nabla^2 g_\gamma(y) = \frac{1}{\gamma} \frac{[p'_Y(y)]^2 - p''_Y(y) p_y(y)}{p^2_Y(y)}$, and thus

$$\eta''_\gamma(x) = y^2 \left(1 + \frac{2y}{\gamma} \frac{p'_Y(y)}{p_Y(y)} - \frac{y^2}{\gamma} \frac{[p'_Y(y)]^2 - p''_Y(y) p_y(y)}{p^2_Y(y)}\right) \tag{6.137}$$

$$= \frac{y^2}{\gamma p^2_Y(y)} \left(\gamma p^2_Y(y) + 2y p'_Y(y) p_Y(y) - y^2 [p'_Y(y)]^2 + y^2 p''_Y(y) p_y(y)\right) \tag{6.138}$$

Moreover

$$p_Y(y) = \int_z \alpha(z) p_X(z) y^{-\gamma} \exp\left(-\gamma \frac{z}{y}\right) dz = \int_x u(z, y) dz. \tag{6.139}$$

$$p'_Y(y) = \frac{-\gamma}{y} p_Y(y) + \frac{\gamma}{y^2} \int_z z u(z, y) dz = \frac{-\gamma}{y} p_Y(y) + \frac{\gamma}{y^2} I^{(1)}(y) \tag{6.140}$$

where we denote $I^{(1)}(y) = \int_z zu(z,y)dz$.

$$p''_Y(y) = (\gamma^2+\gamma)\frac{1}{y^2}p_Y(y) - 2(\gamma^2+\gamma)\frac{1}{y^3}I^{(1)}(y) + \gamma^2\frac{1}{y^4}I^{(2)}(y), \tag{6.141}$$

where we denote $I^{(2)}(y) = \int_z z^2u(z,y)dz$.

Eventually, we get, after simplification,

$$\eta''_\gamma(x) = \frac{y^2}{\gamma p_Y^2(y)}\frac{\gamma^2}{y^2}\left(I^{(2)}(y)p_Y(y) - [I^{(1)}(y)]^2\right) \tag{6.142}$$

$$= \frac{\gamma}{p_Y^2(y)}\int_z\int_{z'}\left(\frac{z^2}{2} + \frac{z'^2}{2} - zz'\right)u(z,y)u(z',y)dzdz' \tag{6.143}$$

$$= \frac{\gamma}{p_Y^2(y)}\int_z\int_{z'}\frac{(z-z')^2}{2}u(z,y)u(z',y)dzdz' \tag{6.144}$$

$$= \frac{\gamma y^{-2\gamma}}{p_Y^2(y)}\int_z\int_{z'}\frac{(z-z')^2}{2}\exp\left(-\gamma\frac{z+z'}{y}\right)\alpha(z)\alpha(z')p_X(z)p_X(z')dzdz' \geq 0 \tag{6.145}$$

As $\alpha(x) > 0$, the previous term is 0 if and only if $p_X(z)p_X(z') = 0$ when $z \neq z'$, which would imply $p_X(z) = 0\ \forall z$. This is impossible since p_X is a proper pdf.

The extension for $n > 1$ is straightforward and $(z-z')^2$ becomes $\langle z-z', d\rangle^2$. □

With the Denoising Score Matching result of Proposition 26, we showed that a denoiser trained to denoiser the Bregman noise with the L^2 loss actually approximates the MMSE denoiser. The fact that the Jacobian condition is verified by the MMSE then justifies that it is also likely to be verified by a denoiser properly trained with L^2 loss. We will check the validation of the hypothesis of positive definiteness of the Jacobian after training our Bregman Score Denoiser.

Denoising in practice Like in Chapters 4 and 5, we parameterize the deep potential g_γ as

$$g_\gamma(y) = \frac{1}{2}\left|\left|y - N_\gamma(y)\right|\right|^2, \tag{6.146}$$

where N_σ is the deep convolutional neural network architecture DRUNet (Zhang et al., 2021), with 2 residual blocks at each scale and Softplus activations. We condition the network N_γ on γ similarly to what is done for DRUNet. We stack to the 3 color channels of the input image an additional channel containing an image with constant pixel value equal to $1/\gamma$. ∇g_γ is computed with automatic differentiation. We train $\mathcal{B}_\gamma$ to denoise images corrupted with random Inverse Gamma noise of level γ, sampled from clean images via $p(y|x) = \prod_{i=1}^n \mathcal{IG}(\alpha_i,\beta_i)(x_i)$. To sample $y_i \sim \mathcal{IG}(\alpha_i,\beta_i)$, we sample $z_i \sim \mathcal{G}(\alpha_i,\beta_i)$ and take $y_i = 1/z_i$. Denoting as p the distribution of a database of clean images, training is performed with the L^2 loss

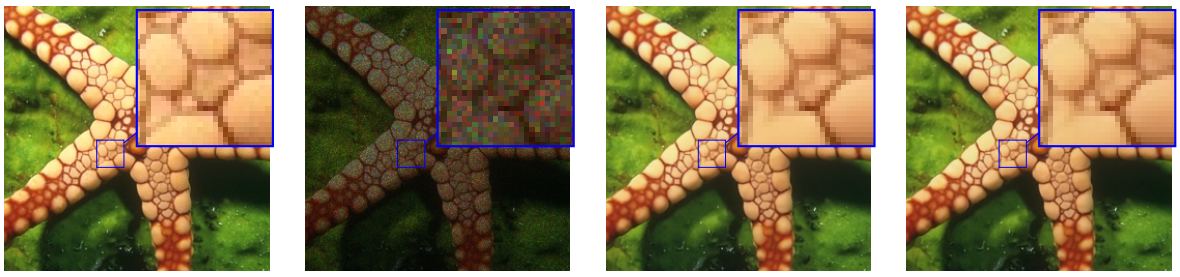
$$\mathcal{L}(\gamma) = \mathbb{E}_{x\sim p, y\sim\mathcal{IG}_\gamma(x)}\left[\left|\left|\mathcal{B}_\gamma(y) - x\right|\right|^2\right], \tag{6.147}$$

with $1/\gamma$ uniformly sampled in $(0, 0.1)$. Training is performed with ADAM during 1200 epochs. The learning rate is initialized with learning rate 10^{-4} and is divided by 2 at epochs 300, 600 and 900.

Denoising performance We evaluate the performance of the proposed Bregman Score DRUNet (B-DRUNet) denoiser (6.146). We compare the performance of B-DRUNet (6.146) with the same network DRUNet directly trained to denoise inverse Gamma noise with L^2 loss. Qualitative and quantitative results presented in Figure 6.1 and Table 6.1 show that the Bregman Score Denoiser (B-DRUNet), although constrained to be written as (6.115) with a conservative vector field ∇g_γ, performs on par with the unconstrained denoiser (DRUNet).

γ	10	25	50	100	200
DRUNet	28.42	30.91	32.80	34.76	36.79
B-DRUNet	28.38	30.88	32.76	34.74	36.71

Table 6.1: Average denoising PSNR performance of Inverse Gamma noise denoisers B-DRUNet and DRUNet on 256×256 center-cropped images from the CBSD68 dataset, for various noise levels γ.



(a) Clean (b) Noisy (17.85 dB) (c) DRUNet (32.56 dB) (d) B-DRUNet (32.56 dB)

Figure 6.1: Denoising of a 256×256 image corrupted with Inverse Gamma noise of level $\gamma = 25$.

Validation of the positive definiteness hypothesis Given the trained Bregman Score Denoiser $\mathcal{B}_\gamma = y - y^2 \nabla g_\gamma(y)$, we propose to empirically verify the validity of the hypothesis of from Proposition 27 i.e. $J_{\mathcal{B}_\gamma}$ positive definite on $\operatorname{int}\operatorname{dom}(h) = \mathbb{R}^n_{++}$. We showed in Lemma 9 that this is equivalent to $\forall y \in \mathbb{R}^n_{++}, \forall d \in \mathbb{R}^n$

$$\left\langle \nabla^2 \eta_\gamma(\nabla h(y)) d, d \right\rangle = \sum_{i=1}^{n} \left(y^2(1 - 2y\nabla g_\gamma(y)) \right)_i d_i^2 - \left\langle y^4 \nabla^2 g_\gamma(y) d, d \right\rangle \geq 0. \tag{6.148}$$

In order to verify this assumption locally, we represent in Figure 6.2

$$c(\gamma, \xi) = \min_{x \in \{X_i\}, d \in \{D_i\}} \left\langle \nabla^2 \eta_\gamma \left(\frac{-1}{y_\xi} \right) d, d \right\rangle \tag{6.149}$$

$$= \min_{x \in \{X_i\}, d \in \{D_i\}} \sum_{i=1}^{n} \left(y_\xi^2(1 - 2y_\xi \nabla g_\gamma(y_\xi)) \right)_i d_i^2 - \left\langle y_\xi^4 \nabla^2 g_\gamma(y_\xi) d, d \right\rangle \tag{6.150}$$

where the minimum is taken over

- $\{X_i\}_i$: the images from an evaluation dataset of clean images, here the 68 images from CSBD68.

- $\{D_i\}$: 100 random images realizations of a uniform random variable $D_i \sim \mathcal{U}[0,1]^n$.

and y_ξ is obtained from x by interpolating between $\hat{y}_\xi \sim p_\xi(y|x)$ a noisy version of x and $\mathcal{B}_\gamma(\hat{y}_\xi)$ with $\alpha \sim \mathcal{U}[0,1]$ as

$$y_\xi = \alpha \hat{y}_\xi + (1-\alpha)\mathcal{B}_\gamma(\hat{y}_\xi). \tag{6.151}$$

In the figures, we represent $c(\gamma, \xi)$ with respect to:

- in the x-axis: the denoiser strength, i.e. the γ parameter of the denoiser.
- in the y-axis: the noise level ξ in the input image. Figure 6.2(a), the noise model is Inverse Gamma noise $p_\xi(y|x) = \mathcal{IG}(\xi - 1, \xi x)$ i.e. the noise model used for training the denoiser. In figure 6.2(b), the noise model is Poisson $p_\xi(y|x) = \mathcal{P}(\xi x)$ i.e. the noise model of the degradation on which our plug-and-play algorithms are evaluated, and in Figure 6.2(c), the noise model is Gaussian $p_\xi(y|x) = \mathcal{N}(x, \xi^2 \,\mathrm{Id})$.

We observe that for a large variety of noise levels γ and even far away from the image manifold i.e. for large noise, we verify the positive definite assumption.

6.4.2 Bregman Plug-and-Play for Poisson Image Deblurring

We now derive the explicit B-RED and B-PnP algorithms in the context of Poisson image restoration. Choosing $C = [0, R]^n$ for some $R > 0$, the B-RED and B-PnP algorithms (6.108) and (6.109) write

$$\textbf{(B-RED)} \quad x_i^{k+1} = \arg\min\{x \nabla F_{\lambda,\gamma}(x^k)_i + \frac{1}{\tau}\left(\frac{x}{x_i^k} - \log\frac{x}{x_i^k}\right) : x \in [0, R]\} \tag{6.152}$$

$$= \begin{cases} \frac{x_i^k}{1+\tau x_i^k \nabla F_{\lambda,\gamma}(x^k)_i} & \text{if } \; 0 \leq \frac{x_i^k}{1+\tau x_i^k \nabla F_{\lambda,\gamma}(x^k)_i} \leq R \\ R \quad \text{else} \end{cases} \tag{6.153}$$

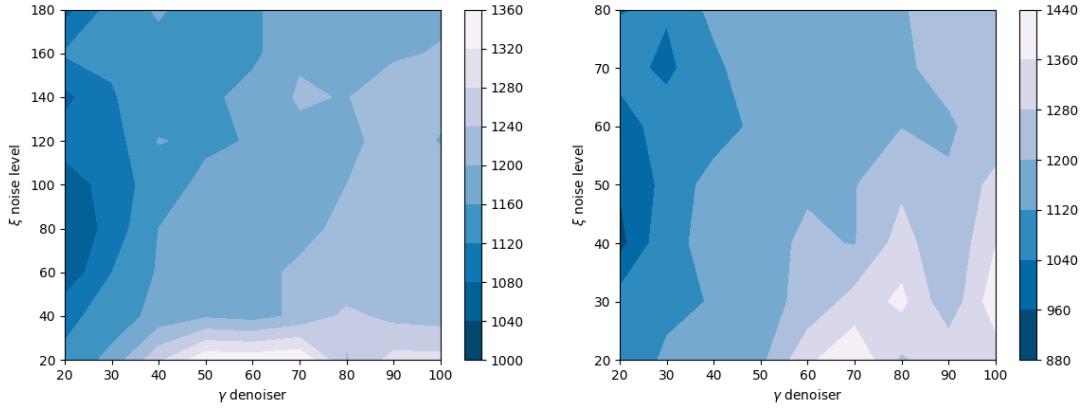
$$\textbf{(B-PnP)} \quad x^{k+1} = \mathcal{B}_\gamma\left(\frac{x^k}{1 + \tau x^k \nabla f(x^k)}\right). \tag{6.154}$$

Verification of the assumptions for convergence In this paragraph, we verify that the Burg's entropy h in (6.112) and the Poisson data likelihood f defined in (6.1) verify all the assumptions required for convergence of B-RED (6.152) and B-PnP (6.154) in Theorems 25 and 26. We remind the expressions of h and f:

$$h(x) = -\sum_{i=1}^{n} \log(x_i), \tag{6.155}$$

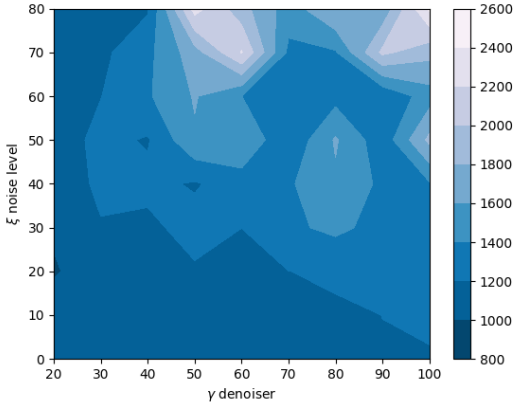
$$f(x) = \sum_{i=1}^{m} y_i \log\left(\frac{y_i}{\alpha(Ax)_i}\right) + \alpha(Ax)_i - y_i, \tag{6.156}$$

for $A \in \mathbb{R}^{m \times n}$. Note that as done in Bauschke et al. (2017), denoting $(a_i)_{1 \leq i \leq n}$ the columns of A, we assume that $a_i \neq 0_m$ and $\forall 1 \leq j \leq m, \sum_{i=1}^n a_{i,j} > 0$ such that $Ax \in \mathbb{R}^m_{++}$ if $x \in \mathbb{R}^n_{++}$. This is verified for A representing the blur with circular boundary conditions with the (normalized) kernels used in Section 4.3.2.



(a) Inverse Gamma noise $p_\xi(\hat{y}_\xi|x) = \mathcal{IG}(\xi - 1, \xi x)$.

(b) Poisson noise $p_\xi(\hat{y}_\xi|x) = \mathcal{IG}(\xi x)$.



(c) Gaussian noise $p_\xi(\hat{y}_\xi|x) = \mathcal{IG}(x, \xi^2\,\mathrm{Id})$.

Figure 6.2: Plot of $c(\gamma, \xi) = \min_{y_\xi, d} \left\langle \nabla^2 \eta_\gamma \left(\frac{-1}{y_\xi} \right) d, d \right\rangle$ w.r.t the denoiser parameter γ and the noise level in the input image ξ. From $\{x_i\}$ the natural images from the CBSD68 testset and 100 random $d \sim \mathcal{U}[0,1]^n$, the image y_ξ is obtained from $x \in \{x_i\}$ via the interpolation $y = \alpha \hat{y}_\xi + (1-\alpha)\mathcal{B}_\gamma(\hat{y}_\xi)$ where $\hat{y}_\xi$ is a noisy version of x sampled with (a) Inverse Gamma (b) Poisson (c) Gaussian noise distributions. $\mathcal{B}_\gamma$ is the trained Bregman Score Denoiser.

We first verify **Assumptions 4**.

(ii) f is real analytic and thus subanalytic on $\mathbb{R}^n_{++}$.

(iii) It is shown in (Bauschke et al., 2017, Lemma 7) that f verifies the NoLip assumption, *i.e.* $L_f h - f$ is convex on $\mathbb{R}^n_{++}$, for $L_f \geq ||y||_1$. y stands for the Poisson degraded observation, appearing in the definition of f.

(iv) First, h is strongly convex on every bounded subset of $\mathbb{R}^n_{++}$. Indeed, for C a bounded subset of $\mathbb{R}^n_{++}$, as $\nabla^2 h(x) = \frac{1}{x^2}\,\mathrm{Id}$, we have $\forall x \in C, \forall d \in \mathbb{R}^n, \langle \nabla^2 h(x) d, d \rangle > \frac{1}{||z||^2_{\infty,C}} ||d||^2$. Second, h and f are Lipschitz continuous everywhere on $\mathbb{R}^n_{++}$ except

close to 0. For both B-RED and B-PnP $\Psi(x) \to +\infty$ when $x \to 0$ and $\{\Psi(x) \leq \alpha\}$ avoids the case $x \to 0$.

(v) We remind the parameterizations $\mathcal{B}_\gamma = \mathrm{Id} - \nabla g_\gamma$ and $g_\gamma(y) = \frac{1}{2}\left|\left|x - N_\gamma(x)\right|\right|^2$ with a U-Net N_γ (with softplus activations). As explained in details in Section 3.1.2, by composition and sum of real analytic functions (Lemma 3(i)) N_γ and then subsequently g_γ are real analytic functions.

By the inverse function theorem (Lemma 3(iii)), as $\forall y \in \operatorname{int}\operatorname{dom}(h)$, $J_{\mathcal{B}_\gamma}(y) > 0$, $\mathcal{B}_\gamma^{-1}$ is then real analytic on $Im(\mathcal{B}_\gamma)$. We finally obtain, using the expression (6.59) of ϕ_γ on $Im(\mathcal{B}_\gamma)$, again by sum and composition, that ϕ_γ is real analytic (and thus subanalytic) on its domain.

Eventually, g_γ is non-negative and we now prove that we have $\forall y \in \mathbb{R}^n$, $\phi_\gamma(y) \geq g_\gamma(y)$. If $y \notin \operatorname{int}\operatorname{dom}(h)$, as $\mathrm{Im}(\mathcal{B}_\gamma) \subset \operatorname{int}\operatorname{dom}(h)$, $\phi_\gamma(y) = +\infty$, this is verified. If $y \in \operatorname{dom}(h)$, as $D_h(y, y) = 0$, we have

$$\begin{aligned} \phi_\gamma(y) &= \phi_\gamma(y) + D_h(y, y) \\ &\geq \phi_\gamma(\mathcal{B}_\gamma(y)) + D_h(\mathcal{B}_\gamma(y), y) \\ &= g_\gamma(y), \end{aligned} \tag{6.157}$$

where the inequality comes from (6.60) and the last equality from (6.59). Therefore ϕ_γ is also lower-bounded.

Second, we verify **Assumption 5** required for the convergence of B-RED.

(i) $[0, R]^n$ is a non-empty closed, bounded, convex and semi-algebraic subset of $\mathbb{R}^n_{++}$.

(ii) With the parameterization $g_\gamma(y) = \frac{1}{2}\left|\left|x - N_\gamma(x)\right|\right|^2$ with a neural network N_γ. g_γ can be shown to have Lipschitz gradient on $B(0, R)$ (see Section 4.1.2). We have $\forall x \in (0, R]^n$, $\forall d \in \mathbb{R}^n$,

$$\langle \nabla^2 g_\gamma(x) d, d\rangle \leq Lip(g_\gamma)\left|\left|d\right|\right|^2 \leq Lip(g_\gamma) R^2 \sum_{i=1}^n \frac{d_i^2}{x_i^2} = Lip(g_\gamma) R^2 \langle \nabla^2 h(x) d, d\rangle, \tag{6.158}$$

which proves that, for $L_\gamma = Lip(g_\gamma)R^2$, $L_\gamma h - g_\gamma$ is convex on $(0, R]^n$.

We finally discuss the additional assumptions required for the convergence of B-PnP in Theorem 26.

(i) $\mathrm{Im}(\mathcal{B}_\gamma) \subseteq \operatorname{dom}(h) = \mathbb{R}^n_{++}$. We train the denoiser $\mathcal{B}_\gamma$ to restore images in $[\epsilon, 1]^n$ (with $\epsilon = 10^{-3}$), the denoiser is thus softly enforced to have its image in this range. In practice, we empirically verify during the iterations that we always get $x_k > 0$.

(ii) *$J_{\mathcal{B}_\gamma}$ positive definite on* $\operatorname{int}\operatorname{dom}(h)$. See the paragraph "Validation of the positive definiteness hypothesis" Section 6.4.1.

(iii) $\mathrm{Im}(\nabla h - \lambda \nabla f) \subseteq \operatorname{dom}(\nabla h^*)$. We now show that this condition is true if $\lambda \left|\left|y\right|\right|_1 < 1$. For $x > 0$

$$\nabla h(x) - \lambda \nabla f(x) = -\frac{1}{x} - \lambda \nabla f(x) = -\frac{x + x\lambda \nabla f(x)}{x}. \tag{6.159}$$

Thus we need to verify that $\forall 1 \leq i \leq n$, $1 + \lambda x_i \nabla f(x)_i > 0$. For f the Poisson data-fidelity term, using $(Ax)_j = \sum_{k=1}^{n} a_{j,k} x_k$, we have $\forall 1 \leq i \leq n$,

$$\nabla f(x)_i = \sum_{j=1}^{m} -y_j \frac{a_{j,i}}{\sum_{k=1}^{n} a_{j,k} x_k} + \alpha a_{j,i} \tag{6.160}$$

and

$$1 + \lambda x_i \nabla f(x)_i = 1 + \alpha\lambda \sum_{j=1}^{m} a_{j,i} x_i - \lambda \sum_{j=1}^{m} y_j \frac{a_{j,i} x_i}{\sum_{k=1}^{n}, a_{j,k} x_k}. \tag{6.161}$$

We assumed that A has positive entries and $\sum_{j=1}^{m} a_{j,i} = r_i > 0$. Therefore, using $0 \leq \frac{a_{j,i} x_i}{\sum_{k=1}^{n}, a_{j,k} x_k} < 1$, we get

$$1 + \lambda x_i \nabla f(x)_i \geq 1 - \lambda \left|\left|y\right|\right|_1 \tag{6.162}$$

which is positive if $\lambda \left|\left|y\right|\right|_1 < 1$.

(iv) *The stepsize condition* $\lambda L_f < 1$. Using the NoLip constant proposed in (Bauschke et al., 2017, Lemma 7) $L_f = \left|\left|y\right|\right|_1$, the condition boils down to $\lambda \left|\left|y\right|\right|_1 < 1$. The condition $\lambda \left|\left|y\right|\right|_1 < 1$ is too restrictive in practice, as the value of $\left|\left|y\right|\right|_1$ can be huge, especially for large images. This is due to the fact that the NoLip constant $L_f \geq \left|\left|y\right|\right|_1$ can be largely over-estimated. Indeed, in the proof of (Bauschke et al., 2017, Lemma 7) as well as in the proof of the previous point, the upper bound

$$\frac{a_{j,i} x_i}{\sum_{k=1}^{n}, a_{j,k} x_k} < 1 \tag{6.163}$$

can be loose in practice. For B-RED this is not a problem, as we use automatic stepsize backtracking. However, for B-PnP, backtracking is not possible as the stepsize is fixed to $\tau = 1$. In order to still guarantee the convergence, we adopt the following backtracking-like strategy to adjust the regularization parameter λ :

- Choose an initial value for $\lambda > 0$.
- At each iteration of the B-PnP algorithm, check sufficient decrease of the objective function $F = \lambda f + \phi_\gamma$ i.e.

$$F(x_k) - F(x_{k+1}) \leq \delta D_h(x_{k+1}, x_k). \tag{6.164}$$

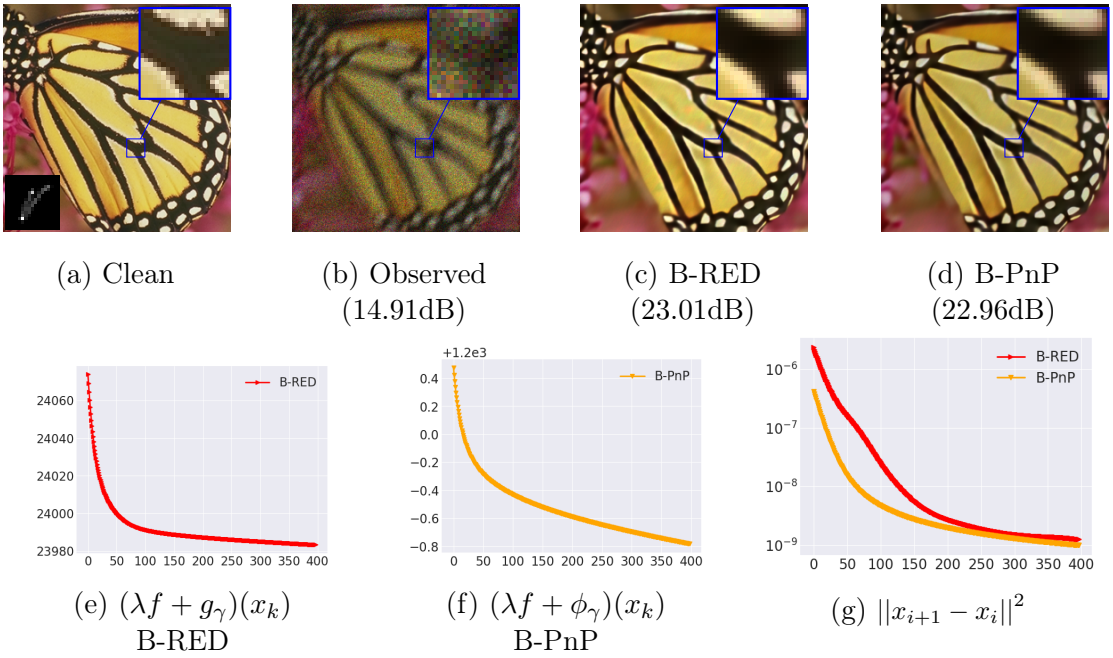
  If at some iteration, this condition is not satisfied before convergence, we alert the user and restart the algorithm with $\lambda \longleftarrow \eta\lambda$ for some $\eta \in (0, 1)$. We also let the user know that, for optimal performance, it might be necessary to adjust the regularization parameter γ of the denoiser, in order to compensate for this decrease of λ.

In our deblurring experiments, for the proposed value of λ, over the variety of blur kernels and noise levels experimented, the sufficient decrease property was always verified and this backtracking algorithm was never activated. This illustrates that $\left|\left|y\right|\right|_1$ is a rough approximation of the NoLip constant.

Poisson deblurring Equipped with the Bregman Score Denoiser, we investigate the practical performance of the B-RED and B-PnP algorithms for image deblurring with Poisson noise. In this context, the degradation operator A is a convolution with a blur kernel.

The hyperparameters γ, λ are optimized for each algorithm and for each noise level α by grid search. In practice, we first initialize the algorithm with 100 steps with large τ and γ so as to quickly initialize the algorithm closer to a relevant stationary point. More details are given in the next paragraph. The algorithm terminates when the relative difference between consecutive values of the objective function is less than 10^{-8} or the number of iterations exceeds $K = 500$.

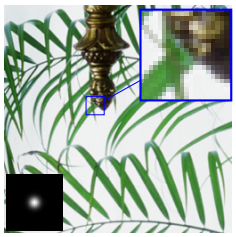
We show in Figures 6.3 and 6.4 that both B-PnP and B-RED algorithms provide good visual reconstruction. Moreover, we observe that, in practice, both algorithms satisfy the sufficient decrease property of the objective function as well as the convergence of the iterates.



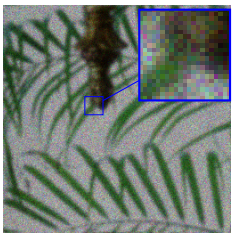
(a) Clean (b) Observed (14.91dB) (c) B-RED (23.01dB) (d) B-PnP (22.96dB)

(e) $(\lambda f + g_\gamma)(x_k)$ B-RED (f) $(\lambda f + \phi_\gamma)(x_k)$ B-PnP (g) $||x_{i+1} - x_i||^2$

Figure 6.3: Deblurring from the indicated motion kernel and Poisson noise with $\alpha = 40$.



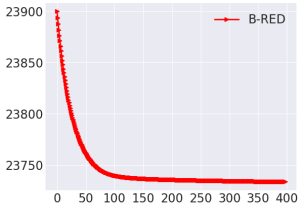
(a) Clean



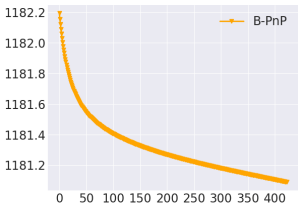
(b) Observed (16.21dB)

(c) B-RED (24.50dB)

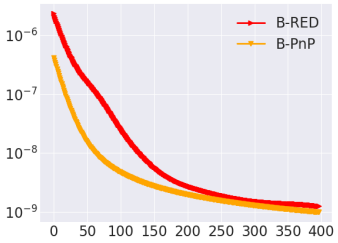
(d) B-PnP (24.51dB)



(e) $(\lambda f + g_\gamma)(x_k)$
B-RED



(f) $(\lambda f + \phi_\gamma)(x_k)$
B-PnP

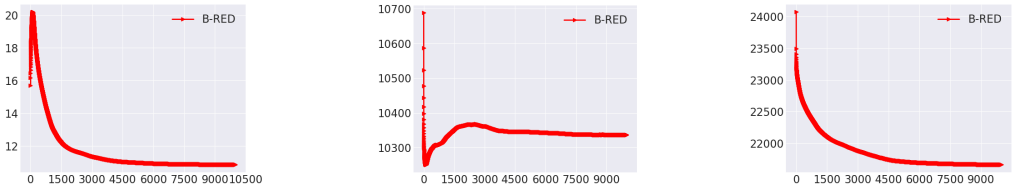


(g) $||x_{i+1} - x_i||^2$

Figure 6.4: Deblurring from the indicated Gaussian kernel and Poisson noise with $\alpha = 60$.

Choice of hyperparameters For B-RED, stepsize backtracking is performed with $\gamma = 0.8$ and $\eta = 0.5$.

When performing plug-and-play image deblurring with our Bregman Score Denoiser trained with Inverse Gamma noise, for the right choice of hyperparameters λ and γ, we may observe the following behavior. The algorithm first converges towards a meaningful solution. After hundreds of iterations, it can converge towards a different stationary point that does not correspond to a visually good reconstruction. We illustrate this behavior in Figure 6.5 where we plot the evolution of the PSNR and of the function values $f(x_k)$ and $g_\gamma(x_k)$ along the algorithm.



(a) PSNR(x_k) (b) $f(x_k)$ (c) $g_\gamma(x_k)$

Figure 6.5: Evolution of the PSNR, $f(x_k)$ and $g_\gamma(x_k)$ when deblurring with B-RED with the initialization parameters from Table 6.2 and without hyperparameter update after 100 iterations. We observe a first phase of fast decrease of both the data-fidelity term and regularization term values, resulting in a fast PSNR increase. After approximately 100 iterations, the regularization keeps on decreasing and the iterates converge towards a different stationary point with low PSNR.

This phenomenon can be mitigated by using small stepsize, large γ and small λ values at the expense of slowing down significantly the algorithm. To circumvent this issue, we propose to first initialize the algorithm with 100 steps with initial τ, γ and λ values and then to change these parameters for the actual algorithm. Note that it is possible for B-RED to change the stepsize τ but not for B-PnP which has fixed stepsize $\tau = 1$. For B-PnP, as done in Chapter 5 in the Euclidean setting, we propose to multiply g_θ by a parameter $0 < \eta < 1$ such that the Bregman Score denoiser becomes $\mathcal{B}^\eta_\gamma(y) = y - \eta(\nabla^2 h(y))^{-1}.\nabla g_\gamma(y)$. The convergence of B-PnP with this denoiser follows identically. The overall hyperparameters λ, γ, τ and η for B-PnP and B-RED algorithms for initialization and for the actual algorithm are given in Table 6.2.

	α	20	40	60
B-RED	Initialization $\tau = 1$ $\gamma = 50$	$\lambda = 1.5$	$\lambda = 2.$	$\lambda = 2.5$
	Algorithm $\tau = 0.05$ $\gamma = 500$	$\lambda = 0.5$	$\lambda = 0.5$	$\lambda = 0.5$
B-PnP	Initialization $\eta = 1$ $\gamma = 50$	$\lambda = 1.5$	$\lambda = 2.$	$\lambda = 2.5$
	Algorithm $\eta = 0.05$ $\gamma = 500$	$\lambda = 0.025$	$\lambda = 0.025$	$\lambda = 0.025$

Table 6.2: B-RED and B-PnP hyperparameters

Quantitative performance and comparison with existing methods. We verify the efficiency of both algorithms over a variety of blur kernels (real-world camera shake, uniform and Gaussian). We present in Figure 6.6 the four blur kernels used for evaluation. We provide in Table 6.3 a quantitative comparison between our 2 algorithms B-RED and B-PnP and three other methods:

(a) (b) (c) (d)

Figure 6.6: The 4 blur kernels used for deblurring evaluation. (a) and (b) are real-world camera shake kernels from Levin et al. (2009). (c) is a 9×9 uniform kernel. (d) is a 25×25 Gaussian kernel with standard deviation 1.6.

(i) PnP-PGD corresponds to the plug-and-play proximal gradient descent algorithm $x_{k+1} = D_\sigma \circ (\mathrm{Id} - \tau \nabla f)$ with D_σ the DRUNet denoiser (same architecture than B-RED and B-PnP) trained to denoiser Gaussian noise.

(ii) PnP-BPG corresponds to the B-PnP algorithm $x^{k+1} = D_\sigma \circ \nabla h^*(\nabla h - \tau \nabla f)(x_k)$ with again the DRUNet denoiser D_σ trained for Gaussian noise.

(iii) ALM Unfolded (Sanghvi et al., 2022) uses the Augmented Lagrangian Method for deriving a 3-operator splitting algorithm that is then trained specifically in an unfolded fashion for image deblurring with a variety of blurs and noise levels α. The publicly available model being trained on grayscale images, for restoring our color images, we treat each color channel independently.

For both (i) and (ii) the parameters σ and τ are optimized for each noise level α. Note that contrary to the proposed B-PnP and B-RED algorithms, the three compared methods do not have any convergence guarantees. We observe that our algorithms perform best when the Poisson noise is not too intense ($\alpha = 40$ and $\alpha = 60$) but that PSNR performance decreases for intense noise ($\alpha = 20$). We assume that this is due to the fact that the denoising prior trained on Inverse Gamma noise is not powerful enough for such a strong noise. As a future direction, we plan on investigating how to increase the regularization capacity of the Inverse Gamma noise denoiser to better handle intense noise.

α	20	40	60
PnP-PGD	23.81	24.41	24.45
PnP-BPG	**23.85**	24.26	24.71
ALM Unfolded (Sanghvi et al., 2022)	23.39	23.91	24.22
B-RED	23.58	**24.54**	**24.90**
B-PnP	23.29	**24.54**	24.80

Table 6.3: PSNR (dB) of Poisson deblurring methods on the CBSD68 dataset. PSNR averaged over the 4 blur kernels represented in Figure 6.6 for each noise levels α.

6.5 Conclusion

In this chapter, we derive a complete extension of the plug-and-play framework in the general Bregman paradigm for non-smooth image inverse problems. In Table 6.4, we provide an overview of the different notions generalized from the Euclidean to the Bregman geometry.

Given a convex potential h adapted to the geometry of the problem, we propose a new deep denoiser, parameterized by h, which provably writes as the Bregman proximal

operator of a nonconvex potential. We argue that this denoiser should be trained on a particular noise model, called Bregman noise, that also depends on h. By plugging this denoiser in the BPG algorithm, we propose two new plug-and-play algorithms, called B-PnP and B-RED, and show that both algorithms converge to stationary points of explicit nonconvex functionals. We apply this framework to Poisson image inverse problems. Experiments on image deblurring illustrate numerically the convergence and the efficiency of the approach.

The central significance of our work stems from its theoretical study, but we recognize certain limits within our experimental results. First, when applied to deblurring with Poisson noise, our proposed algorithms do not outperform existing methods in terms of PSNR. Second, while we prove that B-RED is convergent without restriction, the convergence of B-PnP depends on a specific condition of positive definiteness of the denoiser. Despite being confirmed with experiments and having robust theoretical foundations, this assumption could potentially not be verified when applied to non-natural images that significantly differ from those in the training dataset.

Bregman potential	Euclidian $\frac{1}{2}\|\|x\|\|^2$	$h : \mathbb{R}^n \to \mathbb{R}$ strictly convex, $\mathcal{C}^2$
Smoothness	∇f L-Lipschitz	$Lh - f$ convex
Distance	$\frac{1}{2}\|\|x-y\|\|^2$	Bregman divergence $D_h(x,y)$
$Prox_f^h$	$\arg\min_x f(x) + \frac{1}{2}\|\|x-y\|\|^2$	$\arg\min_x f(x) + D_h(x,y)$
GD	$\mathrm{Id} - \tau\nabla F$	$\nabla h^*(\nabla h - \tau\nabla F)$
PGD	$\mathrm{Prox}_{\tau g} \circ (\mathrm{Id} - \tau\nabla f)$	$\mathrm{Prox}^h_{\tau g} \circ \nabla h^*(\nabla h - \tau\nabla f)$
Plug-and-Play noise	Gaussian	$\exp(-\gamma D_h(x,y) + \rho(x))$
MAP	$\mathrm{Prox}_{-\sigma^2 \log p}(y)$	$\mathrm{Prox}^h_{-\frac{1}{\gamma}\log p}(y)$
MMSE	$y + \sigma^2 \nabla \log p_\sigma(y)$	$y + \frac{1}{\gamma}\nabla^2 h(y)^{-1} \cdot \nabla \log p_\gamma(y)$
GS Denoiser	$D_\sigma(y) = y - \nabla g_\sigma(y)$	$D_\gamma(y) = y - \nabla^2 h(y)^{-1} \cdot \nabla g_\gamma(y)$
It is a Prox *if ...*	D_σ injective	D_γ injective
RED-GD	$\mathrm{Id} - \tau\nabla(\lambda f + (\mathrm{Id} - D_\sigma))$	$\nabla h^*(\nabla h - \tau\nabla(\lambda f + (\mathrm{Id} - D_\sigma)))$
PnP-PGD	$D_\sigma \circ (\mathrm{Id} - \lambda\nabla f)$	$D_\gamma \circ \nabla h^*(\nabla h - \lambda\nabla f)$

Table 6.4: Equivalences between the different notions used for analyzing plug-and-play in the standard Euclidean geometry (Chapters 4 and 5) and their Bregman generalization proposed in this chapter. "GD" stands for Gradient Descent and "PGD" for Proximal Gradient Descent.

Chapter 7

Conclusion and Perspectives

7.1 Summary of the contributions

In this manuscript, we analyzed the convergence of plug-and-play algorithms for solving image inverse problems. After introducing the PnP and RED schemes, we addressed the convergence of both families of algorithms when used with a particular denoiser, called Gradient-Step (GS) denoiser. This denoiser writes as a gradient-descent step on a smooth and nonconvex potential parameterized by a deep neural network. We prove that if the GS denoiser has positive definite Jacobian, then it is a proximity operator. The convergence of PnP and RED algorithms with GS denoiser then follows from the established convergence of first-order optimization algorithms (PGD, DRS or ADMM) in the nonconvex (or weakly convex) setting. Despite the constraints on the denoiser, our RED and PnP algorithms reach state-of-the-art performance, among plug-and-play methods, in various ill-posed inverse problems. These algorithms are however not applicable when the noise affecting the measurements is Poissonian. In the last chapter of the manuscript, we proposed a generalization of our previous study in a different geometry defined by a Bregman divergence, which proves useful for solving Poisson inverse problems. We defined a Bregman extension to the Gradient-Step denoiser, called Bregman Score denoiser, which can again be shown to be a (Bregman) proximity operator under positive definiteness of its Jacobian. This denoiser should be trained on a Bregman noise model, which generalizes Gaussian noise in this new geometry. From the Bregman Proximal Gradient (BPG) scheme, we proposed the Bregman PnP and RED algorithms. When used with the Bregman Score denoiser, both schemes are shown to converge towards stationary points of an explicit nonconvex functional.

7.2 Other scientific productions

During my PhD, I engaged in various collaborative projects whose specific contributions were not detailed in this manuscript. Here is a brief summary of those contributions.

An analysis of generative methods for multiple image inpainting (Ballester et al., 2022). In this book chapter, we analyze in details the recent learning-based methods for diverse image inpainting. The goal of these methods is to provide a set of distinct and coherent solutions for a given damaged image. They capitalize on the probabilistic nature of certain generative models to sample various solutions that coherently restore the missing content. We present a quantitative and qualitative comparisons of a

variety of methods regarding both the quality and the diversity of the set of inpainted solutions. The analysis is performed on different datasets and different type of masks. Our analysis allows us to identify the most successful generative strategies in both inpainting quality and inpainting diversity.

Self-consistent velocity matching of probability flows (Li et al., 2023). This paper resulted from a three-month research visit to MIT with Justin Salomon's group. In this work, we introduce a new framework that does not require discretization for solving a wide range of mass-conserving partial differential equations. This includes for example the time-dependent Fokker-Planck equation and the Wasserstein gradient flow. We give a more in-depth discussion of this contribution in the upcoming perspective section.

DeepInverse (`https://deepinv.github.io/deepinv/`). In collaboration with J. Tachella, D. Chen and M. Terris, we developed DeepInverse, a Pytorch library for imaging with deep learning. The goal of the library is to simplify the implementation of deep learning based methods for imaging inverse problems, by combining popular and state-of-the-art learning-based reconstruction approaches in a common and simplified framework. The objective was to keep a very general and simple structure to make it easy for the user to implement its own regularization, data-fidelity term or optimization algorithm on top of the library.

For variational image restoration, we implemented first a collection of optimization algorithms (PGD, DRS, ADMM, Primal-Dual ...) for minimizing general objectives of the form $\lambda f + g$. Then, we proposed two base classes for representing the data fidelity term f and the prior term g. Both classes come with methods for computing the gradient and the Prox of these terms. The Prior class can be used to implement explicit user-defined priors but also implicit priors, like PnP and RED. In particular, we implemented in the library all the PnP and RED algorithms presented in Section 2.4.2 as well as the Gradient Step (Chapter 4) and Proximal (Chapter 5) denoisers. All the visual results are convergence curves shown in the manuscript can be directly obtained with the library.

Moreover, all the blocks are created with PyTorch modules such that any parameter of the algorithm can be easily trained in an unfolded fashion, be it the regularization, the data-fidelity term or the regularization parameters.

7.3 Research perspectives

This thesis paves the way for numerous research avenues. Within this section, we explore different research directions that could be pursued based on the findings presented in this manuscript.

7.3.1 PnP convergence with Proximal denoiser

How to enforce the Lipschitz condition? In Chapter 5, in order to apply Proposition 25 to the Gradient Step denoiser $D_\sigma = \mathrm{Id} - \nabla g_\sigma$ and make it a proximity operator, we required the deep potential g_σ to have contractive gradient. This condition was softly enforced during the training of g_σ by regularizing the training loss with a term penalizing the spectral norm of the Hessian of g_σ (see (5.98)). With only a soft regularization of the training loss, there is no theoretical guarantee for the 1-Lipschitz condition to be true. This is a limitation of this work. Indeed, even though we empirically verified that ∇g_σ

is contractive on natural noisy images, the condition may not be verified for particular images that do no look like the ones from the training dataset. For instance, when applied to medical, astronomical or microscopic images, it should be necessary to retrain the denoiser on images of the same kind.

In order to hardly enforce this condition, one needs to choose an architecture accordingly. Instead of the Lipschitz continuity of ∇g_σ, a sufficient condition for Proposition 25 to remain true is the convexity of $h_\sigma = \frac{1}{2}||x||^2 - g_\sigma$. Therefore, a possible solution to hardly enforce D_σ to be a Prox is to parameterize h_σ with an Input Convex Neural Network (ICNN) (Amos et al., 2017) or with other types of convex parameterizations. For instance, Goujon et al. (2023) propose convex potentials of the form

$$h(x) = \sum_{i=1}^{p} \psi_i(w_i^T x) \tag{7.1}$$

where ψ_i are convex profile functions and w_i are learnable weights. Instead of parameterizing ψ_i directly, with $\phi_i = \psi_i'$, then

$$\nabla h(x) = W^T \phi(Wx) \tag{7.2}$$

and ∇h writes as a bias-free 1-layer CNN with activation function ϕ. They propose to parameterize ϕ with trainable non-decreasing and non-expansive linear splines (which make ψ_i splines of order 2) In practice, this potential h is used as a regularizer in a variational formulation of the form (2.2). In our context, we could directly parameterize the denoiser $D_\sigma = \nabla h_\sigma$ with such a 1-layer (σ dependent) neural network. As for ICNN, the capacity of this model is strongly limited and is likely to provide inferior performance.

ProxPnP limits on the regularization parameter λ In Chapter 5, we address with PnP-αPGD (5.93) the limitation of the regularization parameter due to the use of the Proximal Denoiser $D_\sigma = \text{Prox}_{\phi_\sigma}$ with PnP-PGD (5.18). More precisely, contrary to PnP-PGD, which restrict the sum of L_f the Lipschitz constant of ∇f and M the weak convexity constant of ϕ_γ, for PnP-αPGD, the condition for convergence (Corollary 5) is on the product between these two terms. Thus, for $M \to 0$ we are free to choose λ as small as desired. We thus proposed a way to control the weak convexity constant M of ϕ_γ with a γ-relaxed version of the Gradient Step denoiser (Section 5.1.3).

However, the nonconvex convergence theory for the PnP-αPGD is not fully established. In particular, we proved in Corollary 5 the decrease of a Liapunov function, convergence of the residual $||x_{k+1} - x_k|| \to 0$ and convergence of a subsequence towards a stationary point of the objective. However, we still did not manage to prove, like for PnP-PGD, single-point convergence of the whole sequence using the Kurdyka-Łojasiewicz property. As explained in Remark 20, we would need to adapt the abstract convergence result (presented in Theorem 4) from Attouch et al. (2013).

Moreover, the γ-relaxation of the Gradient Step denoiser adds again an extra hyperparameter that needs to be tuned. It means that PnP-αPGD has in total four hyperparameters to tune: σ, λ, γ and α. Even though we proposed default values in Chapter 5, this can be challenging and time-consuming for the user.

Different from αPGD, another possible modification of the PGD algorithm is the Primal-Dual Davis-Yin (PDDY) algorithm (Salim et al., 2022). It is a Primal Dual version of the Davis-Yin splitting method (Davis and Yin, 2017) for minimizing the sum of three

functions. It is shown by Condat and Richtárik (2022) (Algorithm 3 with $\mathcal{R}^t = \mathrm{Id}$, $\gamma = \alpha\tau$ and $1 + \omega = \frac{1}{\alpha}$) that, when one of the three functions is 0 , the latter can be written as

$$\begin{aligned} \hat{x}_{k+1} &= x_k - \alpha\tau(\nabla\lambda f(x_k) + u_k) \\ d_{k+1} &= \hat{x}_{k+1} - \mathrm{Prox}_{\tau g}(\hat{x}_{k+1} + \tau u_k) \\ u_{k+1} &= u_k + \frac{\alpha}{\tau} d_{k+1} \\ x_{k+1} &= \hat{x}_{k+1} - \alpha d_{k+1}. \end{aligned} \tag{7.3}$$

Note that for $\alpha = 1$, it falls back to the standard PGD algorithm. For **convex** f and g, adapting the convergence of the PDDY algorithm (Condat and Richtárik, 2022, Theorem 2), we get that for $\alpha\tau\lambda L_f < 2$ and $\alpha \in (0, 1]$, (x_k) converges towards a minimizer of $\lambda f + g$. Therefore, with $\tau = 1$, *convex* convergence is proven provided $\alpha\lambda L_f < 2$. The constraint on λ can be counter-balanced by α which can be as small as desired. However, this is proven only for convex f and g. No convergence results for the PDDY algorithm in the nonconvex setting has been proposed so far. This is a research perspective that would be interesting to follow.

7.3.2 Continuous-time optimization

Gradient Flow (GF) interpretation Given the regularization g_σ (resp. ϕ_σ), our RED (resp. PnP) algorithms are exact optimization algorithms for finding critical points of $F = \lambda f + g_\sigma$ (resp. $F = \lambda f + \phi_\sigma$). We assume for simplicity that all the functions are differentiable. Let g account for g_σ or ϕ_σ. We assume that we have, thanks to the denoising operation, tractable ∇g or Prox_g. These optimization algorithms are different first-order discretizations of the gradient flow

$$\frac{dx(t)}{dt} = -\nabla(\lambda f + g)(x(t)). \tag{7.4}$$

Indeed, forward Euler discretization of (7.4) gives the Gradient Descent algorithm applied to the full objective $F = \lambda f + g$ while Backward Euler discretization corresponds to the proximal point algorithm on F. A Gradient Flow interpretation of the Proximal Gradient Algorithm can be obtained by mixing forward and backward discretizations on the right-hand side of (7.4). It is less straightforward but also possible to derive a flow interpretation of the DRS algorithm (França et al., 2021).

First, it is interesting to directly analyze convergence in the continuous-time setting, independently to the choice of discretization. As explained by Bach (2021), all the traditional annoying issues regarding the choice of step-size, with line-search, constant, decreasing or with a weird schedule are unnecessary. Moreover, the use of differential calculus makes proving properties really simple. For instance, without further Lipschitz continuity assumption, it is possible to show that the objective function F decreases along the flow:

$$\frac{dF(x(t))}{dt} = \nabla F(x(t))^T \frac{dx(t)}{dt} = -\left|\left|\nabla F(x(t))\right|\right|^2 \leq 0. \tag{7.5}$$

Single-point convergence of $x(t)$ for nonconvex and lower-bounded F can also directly be shown on the GF if F verifies the Kurdyka–Łojasiewicz property, introduced in Section 3.1.2.

Second, one can derive **acceleration** schemes by higher-order discretizations *in space* of the gradient flow ODE, with for instance Heun or Runge-Kutta methods. It is also

possible to accelerate by replacing the Gradient Flow ODE by second-order ODEs *in time*. For instance, Nesterov's Accelerated Gradient Method corresponds to the forward Euler discretization of (Su et al., 2014)

$$\frac{d^2x(t)}{dt^2} + \frac{3}{t}\frac{df(x(t))}{dt} = -\nabla f(x(t)) \tag{7.6}$$

It would be interesting to study whether the proposed αPGD algorithm (5.55) from Chapter 5, which was inspired from acceleration methods of PGD, corresponds to a discretization of the Gradient Flow or of any higher-order ODE.

Finally, such analysis could be also generalized to the Bregman paradigm: the Bregman proximity operator (6.2) or the BPG algorithm (6.81) correspond to discretizations of the following Gradient Flow on the Hessian manifold (Gunasekar et al., 2021):

$$\frac{dx(t)}{dt} = -(\nabla^2 h(x(t)))^{-1} \cdot \nabla F(x(t)) \tag{7.7}$$

Non-constant σ parameter Another benefit of the continuous perspective is to use a time-varying σ parameter. In our manuscript, we interpreted PnP and RED algorithms as first-order optimization algorithms for minimizing a fixed objective $\lambda f + g_\sigma$. As the λ parameter, the σ parameter is then a fixed regularization parameter which cannot be iteration-dependent. However, it has been observed in practice (Zhang et al., 2021) that the performance of plug-and-play algorithms can be improved by starting with large σ and annealing σ along the iterations. It is also common practice for score-based generative models (Song and Ermon, 2019; Song et al., 2020c). Recall that with Denoising Score Matching we have, for all σ seen during training, $g_\sigma \approx -\log p_\sigma$ with $p_\sigma = p_X * G_\sigma$ a Gaussian smoothed version of the true image prior p_X. It verifies a weak convergence $p_\sigma \to p_X$ when $\sigma \to 0$. Decreasing σ along the iterations is a coarse-to-fine approach to progressively restore the details in the estimated image. Although we cannot interpret PnP/RED algorithms with decreasing σ as minimizers of a fixed objective, we can still interpret them as discretizations of an ODE of the form

$$\frac{dx(t)}{dt} = -\nabla \left(\lambda f + g_{\sigma(t)}\right)(x(t)) \tag{7.8}$$

where $\sigma = \sigma(t)$ is time-dependent. This ODE is not a gradient flow anymore. It takes the form of the conditional *diffusion ODE* introduced in Song et al. (2020c). Using the fact that $g_\sigma \to g \approx -\log p_X$ when $\sigma \to 0$, an interesting question is to study whether a discretization of this ODE with $\sigma(t) \to 0$ would converge to a critical point of $\lambda f + g$.

7.3.3 Convergent regularization method

A recent line of works (Ebner and Haltmeier, 2022; Hauptmann et al., 2023) studies conditions under which condition PnP is a convergent regularization method. In regularization theory, a convergent regularization method (Clason, 2020) is a well-posed and stable algorithm that converges to a solution of the noiseless operator equation. More formally, consider an observation $y^\delta = Ax^* + \xi$ with a noise bounded by δ i.e. $||\xi|| \leq \delta$. Then we define the family $(\mathcal{R}_\lambda)$ of reconstruction mappings given by the solution of the variational formulation

$$\mathcal{R}_\lambda(y^\delta) \in \arg\min_x \frac{1}{2}\left|\left|Ax - y^\delta\right|\right|^2 + g_\lambda(x) \tag{7.9}$$

where we choose for simplicity a L^2 data-fidelity term, g is a regularization term and $\lambda = \lambda(\delta, y^\delta)$ a regularization parameter that depends on the noise level δ and the input y^δ. In classical regularization theory, $g_\lambda = \lambda g$ and $(\mathcal{R}_\lambda)$ is a convergent regularization method if there exists a parameter choice $\delta \to \lambda(\delta, y^\delta)$ such that $\lambda(\delta, y^\delta) \to 0$ and such that the reconstructions $\mathcal{R}_{\lambda(\delta,y^\delta)}(y^\delta)$ converge (point-wise) to the pseudo-inverse solution $x^\dagger = A^\dagger y^\delta$ when the noise vanishes *i.e.* $\delta \to 0$. This is however too restrictive for image variational regularization techniques. Moreover, with this notion, we cannot define σ as a regularization parameter. Instead, Hauptmann et al. (2023) consider a more general notion of convergent variational regularization method. It is verified if there exists a parameter choice $\delta \to \lambda(\delta, y^\delta)$ such that, when $\delta \to 0$, $\lambda \to \lambda_0$ and $\mathcal{R}_{\lambda(\delta,y^\delta)}(y^\delta) \to x^\dagger$ where $x^\dagger$ is a *g_{λ_0}-minimizing solution*, *i.e.* $g_{\lambda_0}(x^\dagger) = \min_{x\in\mathbb{R}^n} \{g_{\lambda_0}(x) : Ax = y\}$

In the context of our work, for instance for GS-RED (Chapter 4), the regularization term is λg_σ with two regularization parameters σ and λ. We proved that the GS-RED algorithms converge to a solution $\mathcal{R}(y^\delta, \sigma, \lambda)$ which is a critical point of $f + \lambda g_\sigma$. We can now ask if the regularization converges in the sense of regularization theory, *i.e.* if there is an appropriate choice of σ and λ w.r.t. δ such that $\sigma \to \sigma_0$, $\lambda \to \lambda_0$ and $\mathcal{R}(y^\delta, \sigma, \lambda)$ converges to a g_{σ_0}-minimizing solution when $\delta \to 0$.

The first difficulty for showing such a result is that the dependence of g_σ w.r.t σ is hard to characterize explicitly. Hauptmann et al. (2023) empirically showed the convergence of our proposed gradient-step regularization λg_σ w.r.t λ and keeping σ fixed. They plot in their Figure 2 (page 18), for a deblurring problem, the distance $\left|\left|\mathcal{R}(y^\delta, \lambda(\delta)) - x^\dagger\right|\right|$ between the solution $\mathcal{R}(y^\delta, \lambda(\delta))$ of the algorithm GSRED-PGD run with $\lambda(\delta) = c\delta + \lambda_0$ and the solution $x^\dagger$ of the algorithm run with $\lambda = \lambda_0$. We observe on this plot that $\left|\left|\mathcal{R}(y^\delta, \lambda(\delta)) - x^\dagger\right|\right| \to 0$ when $\delta \to 0$.

Proving formally this convergence w.r.t σ is an open research perspective. Convergence w.r.t. λ is easier to deal with. It is proven in (Scherzer et al., 2009, Proposition 3.32) for a strictly convex and coercive objective. However, in our case, the objective is not convex. Following the same proof as (Scherzer et al., 2009, Theorem 3.26), we can however still show the following weak convergence result:

Proposition 30. *Given an observation y, assume $g : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ proper, lsc, lower-bounded and coercive. Let $\lambda : [0, +\infty] \to [0, +\infty]$ such that*

$$\lambda(\delta) \to 0 \text{ and } \frac{\delta^2}{\lambda(\delta)} \to 0 \text{ as } \delta \to 0. \tag{7.10}$$

Let $\delta_k \to 0$ and a sequence of observations $y_k = y^{\delta_k}$ such that $||y - y_k|| \leq \delta_k$. Set $\lambda_k = \lambda(\delta_k)$ and $F_k(x) = \frac{1}{2}||Ax - y_k||^2 + \lambda_k g(x)$. Let (x_k) a sequence of minimizers defined by

$$x_k \in \arg\min_x F_k(x). \tag{7.11}$$

Then (x_k) has a converging subsequence. Moreover, any converging subsequence of (x_k) converges towards $x^\dagger$ a g-minimizing solution and $g(x_k) \to g(x^\dagger)$. If the g-minimizing solution is unique, then $x_k \to x^\dagger$.

Although interesting, the main limitation of the above result is that with condition (7.11), we need to find global minimizers $x_k \in \arg\min_x F_k(x)$. For nonconvex g, this is out of reach, and we can only expect critical points of this objective. Following the proof, we can however replace (7.11) by the weaker condition

$$\forall k,\ F_k(x_k) \leq F_k(x^\dagger) \text{ for any } x^\dagger \text{ } g\text{-minimizing solution.} \tag{7.12}$$

But it is still hard to verify this condition in practice. Moreover, this result does not solve the convergence w.r.t. the σ regularization parameter.

7.3.4 Bayesian Plug-and-Play

Posterior sampling The goal of this thesis is to study the convergence of algorithms for finding a point solution of the MAP problem $x^{MAP} \in \arg\max p_{Y|X}(x|y)$. We introduced an explicit approximation of the smoothed prior $g_\sigma \approx -\log p_X * G_\sigma$ and estimated the maximum of the corresponding posterior

$$p_\sigma(x|y) \propto p(y|x)\hat{p}_\sigma(x) \propto \exp\left(-(\lambda f(x) + g_\sigma(x))\right) \tag{7.13}$$

Instead of just looking for its mode, it would be interesting to say more on this approximation of the posterior. $p_\sigma(x|y)$ is intractable due to the intractable normalization constant. It is however reasonable to sample from it. Having access to an algorithm that samples from this posterior first provides an approximation of the posterior mean $\mathbb{E}[x|y]$ by averaging all the samples. It also permits to estimate the uncertainty on the result by calculating the pixel-wise variance between all generated samples.

Sampling can be reformulated as an optimization problem over the space of distributions. Assuming that the target distribution $\pi = p_\sigma(.|y)$ is regular enough, it is a solution of the problem

$$\arg\min_{\mu} D(\mu, \pi) \tag{7.14}$$

for some dissimilarity functional D. Similarly to the Euclidean case, this minimization problem can be solved using a Gradient Flow differential equation, called Wasserstein Gradient Flow (WGF)

$$\frac{\partial \mu_t}{\partial t} = \operatorname{div}\left(\mu_t \nabla_{W_2} D(\mu_t, \pi)\right) \tag{7.15}$$

There are many possibilities for the choice of D, the most popular being the Kullback-Leibler divergence, for which the WGF corresponds to the Fokker-Planck equation

$$\frac{\partial \mu_t}{\partial t} = -\operatorname{div}(\mu_t \nabla \log \pi) + \Delta \mu_t \tag{7.16}$$

$$= \operatorname{div}\left(\mu_t \nabla(\lambda f + g_\sigma)\right) + \Delta \mu_t \tag{7.17}$$

Similar to the Euclidean case, in order to solve the Fokker-Planck equation, it is possible to discretize the WGF with forward or backward schemes. These methods are limited because they require explicit access to the distribution μ_k (see the next paragraph). Instead, one can use the connection between WGF and Stochastic Differential Equations (SDE): a stochastic process X_t governed by the following SDE

$$dX_t = \nabla \log \pi(X_t)dt + \sqrt{2\tau}dW_t \text{ s.t } X_0 \sim \mu_0 \tag{7.18}$$

with W_t a Brownian motion, has marginal μ_t following the WGF (7.16). The *Unadjusted Langevin Algorithm (ULA)* creates samples by forward discretization of this SDE. It writes, for $Z_k \sim \mathcal{N}(0, \mathrm{Id})$,

$$X_{k+1} = X_k + \tau \nabla \log \pi(X_k) + \sqrt{2\tau} Z_k \tag{7.19}$$

$$= X_k - \tau(\lambda \nabla f(X_k) + \nabla g(X_k)) + \sqrt{2\tau} Z_k. \tag{7.20}$$

Thus, sampling from the posterior using ULA is done by iterating a step of gradient descent on the variational objective, followed by adding some random Gaussian noise. Although ergoticity of the ULA Markov Chain can be shown under some assumptions, at each step, the chain accumulates some error and ultimately does not target the distribution π. ULA can nevertheless be corrected at every iteration by a Metropolis-Hastings accept-reject step, leading to the *Metropolis-Adjusted Langevin Algorithm* (MALA) algorithm (Roberts and Tweedie, 1996). At every iteration (7.19), the proposal X_{k+1} is accepted or rejected according to a test involving the value of $\frac{\pi(X_{k+1})}{\pi(X_k)}$. The correction step ensures that the MALA Markov chain has the correct stationary distribution.

We can create a RED-ULA scheme following the same arguments as for the derivation of the RED-GD algorithm in Section 2.4.2. In particular, given a generic denoiser D_σ that approximates the true MMSE denoiser, with $\nabla(-\log p_\sigma) \longleftarrow \mathrm{Id} - D_\sigma$, we get the RED-ULA algorithm

$$X_{k+1} = X_k - \tau(\lambda \nabla f(X_k) + (X_k - D_\sigma(X_k)) + \sqrt{2\tau} Z_k. \tag{7.21}$$

The convergence of this algorithm is thoroughly studied by Laumont et al. (2021). However, using a generic denoiser, there is no explicit target distribution π. Therefore, the Metropolis-Hastings corrective step cannot be performed. Using the Gradient-Step (GS) denoiser $D_\sigma = \mathrm{Id} - \nabla g_\sigma$ with a nonconvex deep potential g_σ, we recover an explicit objective $-\log \pi = \lambda f + g_\sigma$ and the MALA algorithm could be used to sample exactly from this target distribution. Moreover, the convergence of the RED-ULA (respectively RED-MALA) algorithm with GS denoiser would directly follow from the established convergence of ULA (respectively MALA) for non log-concave distributions (Durmus and Moulines, 2017; Raginsky et al., 2017; Mangoubi and Vishnoi, 2019).

We can also generalize the PGD algorithm for sampling with the Proximal Gradient Langevin Algorithm (PGLA) algorithm (Wibisono, 2018; Salim et al., 2020) which corresponds to a "Forward-Backward" discretization of the SDE (7.18)

$$X_{k+1} = \mathrm{Prox}_{\tau g_\sigma}(X_k - \tau\lambda \nabla f(X_k) + \sqrt{2\tau} Z_k). \tag{7.22}$$

Same as ULA, this Markov chain is biased and can be corrected via Metropolis-Hastings.

Following the same derivation as PnP algorithms in Section 2.4.2, given a generic denoiser D_σ assumed to approximate the MAP denoiser, with $\mathrm{Prox}_{g_\sigma} \longleftarrow \mathrm{Id} - D_\sigma$ we get the PnP-PGLA algorithm

$$X_{k+1} = D_\sigma(X_k - \tau\lambda \nabla f(X_k) + \sqrt{2\tau} Z_k). \tag{7.23}$$

To the best of our knowledge, this scheme has not been studied in the literature for plug-and-play sampling. Once again, using the proximal denoiser introduced in Chapter 5 $D_\sigma = \mathrm{Prox}_{\phi_\sigma}$, the Metropolis-Hastings correction can be performed and convergence results for PnP-PGLA towards the objective distribution $\exp(-(\lambda f + \phi_\sigma))$ could be established.

Discretization-free sampling The main difficulties for solving the Wasserstein Gradient Flow (7.16) are related to time discretizations of the flow. Using forward Euler on the WGF requires closed-form access to the density at each time-step, which is generally not available. Backward discretization corresponds to the so-called JKO scheme (Jordan et al., 1998). It has been proposed in different works (Fan et al., 2021; Mokrov et al., 2021) to approximate each JKO step μ_k by $\Phi_\theta^k \# \mu_0$ with a neural network Φ_θ^k optimized at each

step. However, this is a very time-consuming strategy as one needs to run a full training at each time step, and an unpredictable accumulation of errors may occur.

In contract, we proposed in (Li et al., 2023) a discretization-free framework to solve a wide class of *mass-conserving PDEs* that take the form

$$\frac{\partial \mu_t}{\partial t}(x) = -\operatorname{div}\left(\mu_t f_t(x, \mu_t)\right) \tag{7.24}$$

The WGF (7.16) corresponds to the special case $f_t(x, \mu) = -\nabla_{W_2} D(\mu_t, \pi)(x)$. The idea is to use the fact that (given an initial condition μ_0), μ_t is uniquely determined by its velocity field v_t, and conversely v_t is uniquely determined by μ_t. Both quantities are linked via the continuity equation

$$\frac{\partial \mu_t}{\partial t} = -\operatorname{div}\left(\mu_t v_t\right). \tag{7.25}$$

Therefore, if a probability flow μ_t with velocity field v_t satisfies the following fixed-point equation

$$v_t(x) = f_t(x, \mu_t) \tag{7.26}$$

then the PDE (7.24) is solved. In (Li et al., 2023), we parameterize the velocity field v_t^θ with a time-dependent neural network (for instance with a neural ODE (Chen et al., 2018)) and propose an efficient iterative procedure to minimize

$$\min_\theta \int_0^T \mathbb{E}_{X \sim \mu_t^\theta} \left[\left|\left| v_t^\theta(X) - f_t(X, \mu_t^\theta) \right|\right|^2 \right] dt. \tag{7.27}$$

Our algorithm achieves comparable or better performance in various test cases (Gaussian mixture sampling, Ornstein-Uhlenbeck processes, porous medium equation) compared to JKO methods while using a lower computational budget and without discretizing time. We did not experiment with image data but we plan to extend our model for such high-dimensional modalities.

Uncertainty estimation Beyond single-point MAP estimation, with the frameworks presented in this manuscript, we are unable to estimate the uncertainty in the proposed solutions. Quantification of the uncertainty is important in many applications such as medical imaging, where humans should make decisions based on the proposed solution of the inverse problem. Given an explicit regularization g (for example g_σ from Chapter 4 or ϕ_σ from Chapter 5), in this manuscript we proposed estimations of

$$\hat{x}^{MAP} = \arg\min_x \lambda f(x) + g(x) = \arg\max_x \log p(x|y) \tag{7.28}$$

A first method to estimate uncertainty is to generate many samples $\hat{x}_i$ from the above posterior (see the previous paragraphs for more details on posterior sampling algorithms) and then to visualize the image of standard deviation between samples. As an example, we show Figure 7.1, standard deviation images obtained by Laumont et al. (2021) after sampling their plug-and-play regularized posterior (for the inpainting problem) with the RED-ULA (7.21) scheme. The main limitation of this method is that it requires to generate thousands of samples, which is extremely time-consuming.

It has also been proposed to quantify posterior uncertainty by estimating the *highest posterior density (HPD)* region. The latter is the region of the parameter space where lies

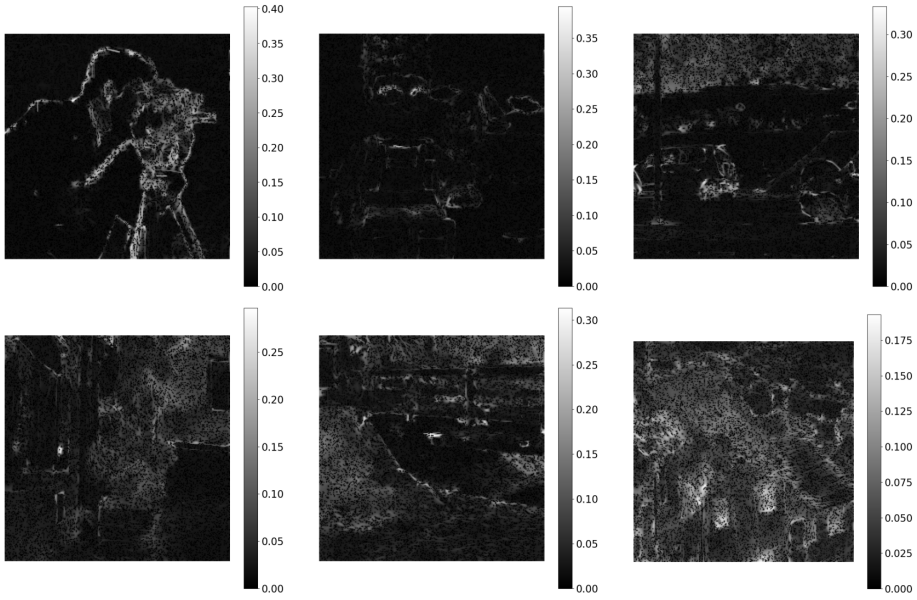


Figure 7.1: Images taken from (Laumont et al., 2021) of standard deviation between samples of the inpainting plug-and-play posterior, obtained with the RED-ULA (7.21) scheme. Uncertainty is located around edges and in textured areas.

most of the posterior probability mass (Pereyra, 2017). For $\alpha \in (0,1)$, a HPD region C_α is defined as

$$C_\alpha = \{x : \lambda f(x) + g(x) \leq \gamma_\alpha\} \text{ where } \gamma_\alpha \text{ is chosen such that } \int_{x \in C_\alpha} p(x|y) = 1 - \alpha \tag{7.29}$$

For **log-concave posterior** (i.e. $f + g$ convex), Pereyra (2017) leverages probability concentration inequalities to approximate C_α by

$$\{x : \lambda f(x) + g(x) \leq \lambda f(x^{MAP}) + g(x^{MAP}) + n(\sqrt{\frac{1}{n}\log(\frac{3}{\alpha})} + 1)\} \tag{7.30}$$

This result can then be used to perform hypothesis tests on questionable structures identified in the MAP estimate (Repetti et al., 2019; Tang and Repetti, 2023). Computing the MAP being much cheaper than estimating the posterior distribution, this method is much more time-efficient than sampling approaches.

However, a limitation of this method is that it requires the minimized objective $\lambda f + g$ to be convex, which is not true in our framework. It could be interesting to investigate if a related result could be derived for nonconvex objectives, for instance by assuming local convexity around critical points.

7.3.5 Connection with Optimal Transport

In Chapter 5, we proposed a denoiser $D_\sigma = \mathrm{Id} - \nabla g_\sigma = \nabla h_\sigma$ which is the gradient of a convex function h_σ. Our denoiser is parameterized with g_σ a neural network. As explained in Section 2.4.1, the theoretical MMSE denoiser also writes, with Tweedie formula (2.74) as $D_\sigma = \mathrm{Id} - \nabla g_\sigma = \nabla h_\sigma$ where $g_\sigma = -\sigma^2 \log p_\sigma$ and where $h_\sigma = \frac{1}{2}||x||^2 - g_\sigma$ is convex. Operators that write as the gradient of a convex function play an important role in optimal

transport. We now recall the optimal transport problem with the simple L^2 cost in $\mathbb{R}^n$. Given two probability measures (μ, ν) on $\mathbb{R}^n$, and for the L^2 cost function,

- the Monge problem consists in finding a map $T : \mathbb{R}^n \to \mathbb{R}^n$ (the Monge map) that realizes
$$\inf_T \left\{ \int_{\mathbb{R}^n} \frac{1}{2} ||x - T(x)||^2 \, d\mu(x) : T\#\mu = \nu \right\} \tag{7.31}$$
- the Kantorovich problem is a relaxation of the Monge problem, with existence guarantee, which consists in finding an optimal transport plan that realizes
$$\min_\gamma \left\{ \int_{\mathbb{R}^n \times \mathbb{R}^n} \frac{1}{2} ||x - y||^2 : \gamma \in \Pi(\mu, \nu) \right\} \tag{7.32}$$
where $\Pi(\mu, \nu)$ denotes the probability measures on $\mathbb{R}^n \times \mathbb{R}^n$ with marginals μ and ν. With a dual formulation, one can show that it is equal to the maximum of the following problem
$$\max_{\psi \ c\text{-convex}} \left\{ \int \psi d\mu + \int \psi^c d\nu \right\} \tag{7.33}$$
where $f^c(y) = \inf_x c(x, y) - f(x)$ denotes the c-transform of f. A c-convex function is a function which writes as f^c for some function f. ψ is called the Kantorovich potential. When the cost is $c(x, y) = \frac{1}{2} ||x - y||^2$, $f^c(x) = \frac{1}{2} ||x||^2 - f^*(x)$ where f^* is the convex conjugate.

Brenier theorem (Brenier, 1991) states that if μ is absolutely continuous, Monge and Kantorovich formulations are equivalent, *i.e.* the optimal transport plan is unique and writes as $(\mathrm{Id} \times T)\#\mu$ with T the Monge map. Moreover, it states that the unique Monge map writes as $T = \nabla h$ the gradient of a convex function. Conversely, any map of the form $T = \nabla h$ for a convex potential h is an optimal transport plan from μ to $T\#\mu$.

The proof of Brenier theorem is based on the equality between (7.32) and (7.33). Indeed, by definition of the c-transform,

$$\forall x, y \quad \psi(x) + \psi^c(y) \leq c(x, y) \tag{7.34}$$

and by equality between (7.33) and (7.32), for $(x_0, y_0) \in \mathit{support}(\gamma)$

$$\psi(x_0) + \psi^c(y_0) = c(x_0, y_0) \tag{7.35}$$

Therefore $x \to \psi(x) - c(x, y_0)$ is minimal at $x = x_0$. For the L^2 cost $c(x, y) = \frac{1}{2} ||x - y||^2$, the optimality equation is $\nabla\psi(x_0) - (x_0 - y_0) = 0$ or

$$y_0 = \nabla \left(\frac{1}{2} ||x_0||^2 - \psi(x_0) \right) = \nabla h(x_0) \tag{7.36}$$

and $h = \frac{1}{2} ||x||^2 - \psi(x)$ is convex by definition of the c-transform for the L^2 cost.

Therefore the MMSE denoiser (and GS denoiser) $D_\sigma = \nabla h_\sigma$ defines an optimal transport map, for the L^2 cost, between $p_\sigma = p_X * G_\sigma$ the distribution of images noised with Gaussian noise of standard deviation σ and $D_\sigma\#p_\sigma$. An open question is then to relate the distributions $D_\sigma\#p_\sigma$ and p_X.

Another interesting perspective is to study the link between Brenier theorem and Theorem 3 from Gribonval and Nikolova (2020) which characterizes proximity operators as gradient of convex functions, *i.e.* to relate optimal transport and proximity operators. From Theorem 3, there is $\phi : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ such that $\forall x \in \mathbb{R}^n$ the Brenier map $T(x) = \nabla h(x) \in \text{Prox}_\phi(x)$. The proof of Theorem 3 from Gribonval and Nikolova (2020) is based on the construction of ϕ (or h) on $\text{Im}(T)$ as

$$\phi(T(x)) = \langle x, T(x)\rangle - \frac{1}{2}\left|\left|T(x)\right|\right|^2 - h(x) \tag{7.37}$$

Then the idea of the proof is to show equivalence between

$$h(y) - h(x) \geq \langle y - x, T(x)\rangle \ \forall y \tag{7.38}$$

which means $T(x) \in \partial h(x)$ (and $\forall x, \partial h(x) \neq \emptyset$ i.e. h is convex), and

$$\frac{1}{2}\left|\left|y - T(x)\right|\right|^2 + \phi(T(x)) \leq \frac{1}{2}\left|\left|y - x\right|\right|^2 + \phi(x) \ \forall y \tag{7.39}$$

which means $T(x) \in \text{Prox}_\phi(x)$.

Note that using Proposition 5,

$$\langle x, T(x)\rangle = \langle x, \nabla h(x)\rangle = h(x) + h^*(T(x)) \tag{7.40}$$

we can rewrite $\phi(T(x))$ as

$$\phi(T(x)) = h^*(T(x)) - \frac{1}{2}\left|\left|T(x)\right|\right|^2 \tag{7.41}$$

Therefore, on $\text{Im}(T)$, ϕ writes as

$$\phi(x) = h^*(x) - \frac{1}{2}\left|\left|x\right|\right|^2 \tag{7.42}$$

We thus have $\phi = -\psi^c$ with ψ the Kantorovich potential solution of (7.33). Actually, the relation $T = \text{Prox}_{-\psi^c}$ could have already been shown using the equality between (7.33) and (7.32). Indeed, after (7.35), we can also observe that $y \to -\psi^c(y) + c(x_0, y)$ is minimal at $y = y_0$. For the L^2 cost $c(x, y) = \frac{1}{2}\left|\left|x - y\right|\right|^2$ it corresponds to

$$y_0 = T(x_0) \in \text{Prox}_{-\psi^c}(x_0). \tag{7.43}$$

Similarly, using for the cost a Bregman divergence, we could show in this way a characterization of Bregman proximity operators reminiscent of Proposition 27.

7.3.6 Bregman Plug-and-Play

Does the Bregman MMSE denoiser have positive-definite Jacobian? As reported in Lemma (2), it is shown in Gribonval (2011) that the MMSE denoiser with Gaussian noise has positive definite Jacobian. Moreover, in Chapter 6, we proved in Lemma 10 that, in the particular case of h being Burg's entropy, the MMSE denoiser with Bregman noise model (6.5) has also positive definite Jacobian. We assume that this remains true for any Bregman potential h but we did not have time to prove it. It would be interesting to generalize this result because it would back up the assumption from Proposition 27 on the fact that the introduced Bregman Score denoiser (which is trained to approximate the MMSE denoiser) is positive definite.

Another Bregman version of the PGD algorithm In our analysis Chapter 6, we departed from the BPG algorithm (6.81) which extend the PGD scheme in the Bregman paradigm. The BPG update writes as the composition of a Bregman proximity operator with a step of mirror descent. However, the proposed Bregman Score denoiser (6.38) does not write as a *mirror descent* step but as a *natural gradient descent* step $\mathcal{B}_\gamma(y) = y - (\nabla^2 h(y))^{-1} \cdot \nabla g_\gamma(y)$. As explained in Gunasekar et al. (2021), both mirror descent and natural gradient descent schemes are different forward discretizations of the same Riemannian Gradient Flow. Instead of departing from the BPG algorithm, it could then make more sense to depart from an algorithm of the form

$$x_{k+1} = \text{Prox}^h_{\tau g} \circ \left(x_k - (\nabla^2 h(x_k))^{-1} \nabla f(x_k)\right) \tag{7.44}$$

The Bregman Score denoiser would then naturally replace the gradient step in this scheme. We would first need to study if this algorithm actually converges to a critical point of $f + g$ under a NoLip condition ($Lh - f$ convex) on f.

Learned Bregman potential In Chapter 6, we introduced the Bregman Score denoiser (6.38) to regularize inverse problems with a data-fidelity term f. The denoiser depends on a convex potential h previously chosen such that the NoLip property is verified, *i.e.* $Lh - f$ is convex for some $L > 0$. For instance, for Poisson inverse problems, f is a Kullback-Leibler divergence and h Burg's entropy allows $Lh - f$ for $L \geq ||y||_1$. However, as explained in Section 6.4.2, this bound on L is too restrictive in practice when dealing with large images. Also, for other non-smooth data-fidelity terms, finding such h by hand could be difficult.

We may wonder if one could train a Bregman potential h_θ parameterized by a scalar neural network, along with the Bregman Denoiser. Given a data-fidelity term f and $L > 0$, h_θ needs (a) to be strictly convex and (b) to verify $Lh_\theta - f$ convex. The first constraint could be hardly enforced, for example by parameterizing h_θ by an Input Convex Neural Network (ICNN) (Amos et al., 2017). The second constraint is harder to deal with. In Chapter 5, convexity of h_σ is softly enforced by regularizing the training loss with a penalization on the spectral norm of the Hessian of $x \to \frac{1}{2}||x||^2 - h_\sigma(x)$, estimated via power iterations. The same strategy could be adopted here for the potential $Lh - f$.

Bregman diffusion models To go further, it would be interesting to study if it could be beneficial to generalize *Gaussian diffusion models* to *Bregman diffusion models*. Similar to RED, diffusion models (or *score-based generative models*) (Song et al., 2020c; Karras et al., 2022) sample from a target distribution p_X using the Denoising Score Matching result (or Tweedie formula) to replace the score $\nabla \log(p * G_\sigma)$ by a Gaussian noise denoiser in a Langevin-based algorithm. It actually comes back to RED-ULA (7.21) with well-chosen varying stepsizes τ_k and decaying noise levels $\sigma_k \to 0$. In particular, these parameters are chosen such that, departing from white Gaussian noise $x_0 \sim \mathcal{N}(0, \sigma^2 \,\text{Id})$, at iteration k, we have $x_k \sim p_X * \mathcal{N}(0, \sigma_k^2 \,\text{Id})$. This ensures that for $k \to \infty$, $x_\infty \sim p_X$. This property is based on the fact that the Langevin scheme adds fresh Gaussian noise at every iteration and that the addition of independent Gaussians is Gaussian. A Bregman diffusion model would first require extending the Langevin diffusion SDE (7.18) to the Bregman geometry, where the Gaussian Brownian motion is replaced by its equivalent with the Bregman noise model (6.5). Then, using the generalization of the Denoising Score Matching result that we established in Proposition 26, the equivalent Bregman score-based generative algorithm would follow.

7.3.7 Monotone operator perspective

In Section 3.2, we proved convergence of the optimization algorithms GD, PGD or DRS applied to **convex** functions by viewing them as fixed-points algorithms. We used the fact that the zeros of the subdifferential of a convex function corresponds to its minimizers. These optimization problems can be generalized by replacing the subdifferential of a convex function by a more general *maximally monotone operator*. We refer to (Bauschke and Combettes, 2011a, Chapter 20) for definitions and details on monotone operators.

Monotone operators form a general class of operators that include the subdifferentials of convex functions but also other kinds of operators, for instance, positive semidefinite matrices. In this more general formalism, the problem

$$\arg\min_x f(x) \Leftrightarrow 0 \in \partial f(x) \tag{7.45}$$

for proper, lsc, convex f generalizes to the inclusion problem $0 \in A(x)$ with A a maximally monotone operator. In this formalism, the proximity operator of f generalizes to the *resolvent* of A

$$J_A : A \to (A + \mathrm{Id})^{-1} \tag{7.46}$$

which is defined everywhere for maximally monotone operators. Also, gradients of differentiable L-smooth convex functions generalize to $\beta = \frac{1}{L}$*-cocoercive operators*. Hence, all the considered first-order optimization algorithms can be generalized for solving monotone inclusion problems. For instance, the PGD algorithm naturally extends to

$$x_{k+1} = J_{\tau A} \circ (\mathrm{Id} - \tau B)(x_k) \tag{7.47}$$

for finding the zeros of $A + B$ where A is maximally monotone and B is $\frac{1}{L}$-cocoercive. It is also theoretically interesting to work with monotone operators because it considerably simplifies the proofs. All the proofs of convergence of convex optimization algorithms in Section 3.2 were derived from Krasnonel'skii-Mann iterations (Theorem 5). We just needed to verify that the proximal map and the gradient descent map of a convex function are averaged operators. The proofs in the general monotone operator formalism follow identically, using that

- An operator is firmly nonexpansive if and only if it is the resolvent of a maximally monotone operator (Bauschke and Combettes, 2011a, Proposition 28.3).
- An operator is firmly nonexpansive if and only it is 1-cocoercive.

Besides, the two above properties generalize the characterization from Moreau (Theorem 2) of proximity operators of convex functions as nonexpansive gradients of convex functions: An operator is 1-cocoercive if and only it is the resolvent of a maximally monotone operator.

However, in Section 3.2, for studying the convergence of optimization in the **nonconvex** setting, we could not use this fixed-point strategy, and we had to build instead more complicated proofs for each algorithm. Indeed, the Prox and gradient descent maps of nonconvex functions are not averaged.

Even though it seems difficult to extend to the fully nonconvex setting, Bauschke et al. (2021) propose, with ρ-monotone operators, an extension of monotone operators for including **weakly convex** functions. Given the fact that both introduced regularizers g_σ from Chapter 4 (as a Lipschitz function) and ϕ_σ in Chapter 5 are weakly convex, such a

theory could permit to get convergence of our RED and PnP algorithms with simplified proofs.

For $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$, proper lsc, and ∂f the limiting subdifferential of f (defined Section 2.2.1). Then, for $\rho \in \mathbb{R}$, f is ρ-convex if and only if ∂f is ρ-maximally monotone.

In parallel, Bauschke et al. (2021) also generalize θ-averaged operators (Definition 9) for $\theta > 1$: for $\theta \in [0, +\infty)$, T is θ-conically nonexpansive if there exists R nonexpansive such that $T = \theta R + (1 - \theta)\,\mathrm{Id}$. In particular, θ-conically nonexpansive operators are not necessarily nonexpansive.

Then, the link between monotonicity and averagedness is generalized in this new framework:

- An operator $T : \mathbb{R}^n \to \mathbb{R}^n$ is θ-conically nonexpansive if and only if $\mathrm{Id} - T = J_A$ where A is a maximally ρ-monotone operator and $\rho = \frac{1}{2\theta} - 1$. (Bauschke et al., 2021, combinaison of Corollary 3.8, Proposition 2.11 and Lemma 2.8).
- An operator $T : \mathbb{R}^n \to \mathbb{R}^n$ is θ-conically nonexpansive if and only if $\mathrm{Id} - T$ is $\frac{1}{2\theta}$-cocoercive (Bauschke et al., 2021, Corollary 3.5).

Furthermore, the two above properties generalize the characterization from Gribonval and Nikolova (2020) (Theorem 3) of proximity operators of weakly convex functions as gradients of smooth convex functions. We summarize this result in the following Proposition.

Proposition 31. *Let $T : \mathbb{R}^n \to \mathbb{R}^n$ and $L > 0$. Then the following are equivalent*

(i) T is $\frac{1}{L}$-cocoercive.

(ii) There is $A : \mathbb{R}^n \to \mathbb{R}^n$ such that $T = J_A$ and A is $\left(\frac{1}{L} - 1\right)$-maximally ρ-monotone.

It would be interesting to analyze if there is a further generalization of monotone operators that would correspond to general nonconvex functions, in order to fully generalize Theorem 3.

Finally, Bartz et al. (2022) propose to extend the fixed-point convergence of PGD or DRS algorithms for the inclusion problems $0 \in A(x) + B(x)$ where A and B are ρ-monotone. However, they require the sum $A + B$ to be monotone. In the optimization case, this means that the overall objective remains convex, which is not the case in our framework. It would be interesting to investigate the convergence with general ρ-monotone objective with $\rho < 0$.

Bibliography

Adler, J. and Öktem, O. (2018). Learned primal-dual reconstruction. *IEEE transactions on medical imaging*, 37(6):1322–1332.

Agustsson, E. and Timofte, R. (2017). Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 126–135.

Aharon, M., Elad, M., and Bruckstein, A. (2006). K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322.

Ahmad, R., Bouman, C. A., Buzzard, G. T., Chan, S., Liu, S., Reehorst, E. T., and Schniter, P. (2020). Plug-and-play methods for magnetic resonance imaging: Using denoisers for image recovery. *IEEE signal processing magazine*, 37(1):105–116.

Al-Shabili, A. H., Xu, X., Selesnick, I., and Kamilov, U. S. (2022). Bregman plug-and-play priors. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 241–245. IEEE.

Amari, S.-I. (1998). Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276.

Amos, B., Xu, L., and Kolter, J. Z. (2017). Input convex neural networks. In *International Conference on Machine Learning*, pages 146–155. PMLR.

Anil, C., Lucas, J., and Grosse, R. (2019). Sorting out lipschitz function approximation. In *International Conference on Machine Learning*, pages 291–301. PMLR.

Attouch, H., Bolte, J., Redont, P., and Soubeyran, A. (2010). Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the kurdyka-łojasiewicz inequality. *Mathematics of operations research*, 35(2):438–457.

Attouch, H., Bolte, J., and Svaiter, B. F. (2013). Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized gauss–seidel methods. *Math. Programming*, 137(1-2):91–129.

Bach, F. (2021). Effortless optimization through gradient flows. `https://francisbach.com/gradient-flows/`.

Ballester, C., Bugeau, A., Hurault, S., Parisotto, S., and Vitoria, P. (2022). An analysis of generative methods for multiple-image inpainting.

Banerjee, A., Merugu, S., Dhillon, I. S., Ghosh, J., and Lafferty, J. (2005). Clustering with bregman divergences. *Journal of machine learning research*, 6(10).

Bartz, S., Dao, M. N., and Phan, H. M. (2022). Conical averagedness and convergence analysis of fixed point algorithms. *Journal of Global Optimization*, 82(2):351–373.

Bauschke, H. H., Bolte, J., and Teboulle, M. (2017). A descent lemma beyond lipschitz gradient continuity: first-order methods revisited and applications. *Mathematics of Operations Research*, 42(2):330–348.

Bauschke, H. H. and Combettes, P. L. (2011a). *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer.

Bauschke, H. H. and Combettes, P. L. (2011b). *Convex analysis and monotone operator theory in Hilbert spaces*, volume 408. Springer.

Bauschke, H. H., Moursi, W. M., and Wang, X. (2021). Generalized monotone operators and their averaged resolvents. *Mathematical Programming*, 189:55–74.

Beck, A. (2017). *First-order methods in optimization*. SIAM.

Beck, A. and Teboulle, M. (2009a). Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE Transactions on Image Processing*, 18(11):2419–2434.

Beck, A. and Teboulle, M. (2009b). Gradient-based algorithms with applications to signal recovery. *Convex optimization in signal processing and communications*, pages 42–88.

Bertero, M., Boccacci, P., Desiderà, G., and Vicidomini, G. (2009). Image deblurring with poisson data: from cells to galaxies. *Inverse Problems*, 25(12):123006.

Bigdeli, S. A. and Zwicker, M. (2017). Image restoration using autoencoding priors. *arXiv preprint arXiv:1703.09964*.

Bohra, P., Goujon, A., Perdios, D., Emery, S., and Unser, M. (2021). Learning lipschitz-controlled activation functions in neural networks for plug-and-play image reconstruction methods. In *NeurIPS Workshop on Deep Learning and Inverse Problems*.

Bolte, J., Daniilidis, A., and Lewis, A. (2007). The łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. *SIAM Journal on Optimization*, 17(4):1205–1223.

Bolte, J., Sabach, S., and Teboulle, M. (2014). Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1-2):459–494.

Bolte, J., Sabach, S., Teboulle, M., and Vaisbourd, Y. (2018). First order methods beyond convexity and lipschitz gradient continuity with applications to quadratic inverse problems. *SIAM Journal on Optimization*, 28(3):2131–2151.

Bora, A., Jalal, A., Price, E., and Dimakis, A. G. (2017). Compressed sensing using generative models. *arXiv preprint arXiv:1703.03208*.

Boulanger, J., Pustelnik, N., Condat, L., Sengmanivong, L., and Piolot, T. (2018). Nonsmooth convex optimization for structured illumination microscopy image reconstruction. *Inverse problems*, 34(9):095004.

Brenier, Y. (1991). Polar factorization and monotone rearrangement of vector-valued functions. *Communications on pure and applied mathematics*, 44(4):375–417.

Buzzard, G. T., Chan, S. H., Sreehari, S., and Bouman, C. A. (2018). Plug-and-play unplugged: Optimization-free reconstruction using consensus equilibrium. *SIAM Journal Imaging Sci.*, 11(3):2001–2020.

Calatroni, L. and Chambolle, A. (2019). Backtracking strategies for accelerated descent methods with smooth composite objectives. *SIAM Journal Optimization*, 29(3):1772–1798.

Censor, Y. and Zenios, S. A. (1992). Proximal minimization algorithm with d-functions. *Journal of Optimization Theory and Applications*, 73(3):451–464.

Chambolle, A. and Pock, T. (2011). A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Math. Imaging and Vision*, 40(1):120–145.

Chambolle, A. and Pock, T. (2016). On the ergodic convergence rates of a first-order primal–dual algorithm. *Mathematical Programming*, 159(1):253–287.

Chan, S., Wang, X., and Elgendy, O. (2016). Plug-and-play ADMM for image restoration: fixed-point convergence and applications. *IEEE Transactions on Computational Imaging*, 3(1):84–98.

Chen, G. and Teboulle, M. (1993). Convergence analysis of a proximal-like minimization algorithm using bregman functions. *SIAM Journal on Optimization*, 3(3):538–543.

Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. (2018). Neural ordinary differential equations. *Advances in neural information processing systems*, 31.

Chen, S. S., Donoho, D. L., and Saunders, M. A. (2001). Atomic decomposition by basis pursuit. *SIAM review*, 43(1):129–159.

Cisse, M., Bojanowski, P., Grave, E., Dauphin, Y., and Usunier, N. (2017). Parseval networks: Improving robustness to adversarial examples. In *International conference on machine learning*, pages 854–863. PMLR.

Clason, C. (2020). Regularization of inverse problems. *arXiv preprint arXiv:2001.00617*.

Cohen, R., Blau, Y., Freedman, D., and Rivlin, E. (2021). It has potential: Gradient-driven denoisers for convergent solutions to inverse problems. In *Neural Information Processing Systems*, volume 34.

Combettes, P. L. (2018). Monotone operator theory in convex optimization. *Mathematical Programming*, 170:177–206.

Condat, L. and Richtárik, P. (2022). Randprox: Primal-dual optimization algorithms with randomized proximal updates. *arXiv preprint arXiv:2207.12891*.

Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K. (2007). Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095.

Daras, G., Dean, J., Jalal, A., and Dimakis, A. (2021). Intermediate layer optimization for inverse problems using deep generative models. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 2421–2432. PMLR.

Davis, D. and Yin, W. (2017). A three-operator splitting scheme and its optimization applications. *Set-valued and variational analysis*, 25:829–858.

Douglas, J. and Rachford, H. H. (1956). On the numerical solution of heat conduction problems in two and three space variables. *Transactions of the American mathematical Society*, 82(2):421–439.

Durmus, A. and Moulines, E. (2017). Nonasymptotic convergence analysis for the unadjusted langevin algorithm.

Ebner, A. and Haltmeier, M. (2022). Plug-and-play image reconstruction is a convergent regularization method. *arXiv preprint arXiv:2212.06881*.

Eckstein, J. (1993). Nonlinear proximal point algorithms using bregman functions, with applications to convex programming. *Mathematics of Operations Research*, 18(1):202–226.

Efron, B. (2011). Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614.

Fan, J., Zhang, Q., Taghvaei, A., and Chen, Y. (2021). Variational wasserstein gradient flow. *arXiv preprint arXiv:2112.02424*.

Figueiredo, M. A. and Bioucas-Dias, J. M. (2009). Deconvolution of poissonian images using variable splitting and augmented lagrangian optimization. In *2009 IEEE/SP 15th Workshop on Statistical Signal Processing*, pages 733–736. IEEE.

Figueiredo, M. A. and Bioucas-Dias, J. M. (2010). Restoration of poissonian images using alternating direction optimization. *IEEE transactions on Image Processing*, 19(12):3133–3145.

França, G., Robinson, D. P., and Vidal, R. (2021). Gradient flows and proximal splitting methods: A unified view on accelerated and stochastic optimization. *Physical Review E*, 103(5):053304.

Gabay, D. (1983). Chapter ix applications of the method of multipliers to variational inequalities. In *Studies in mathematics and its applications*, volume 15, pages 299–331. Elsevier.

Gavaskar, R. G., Athalye, C. D., and Chaudhury, K. N. (2021). On plug-and-play regularization using linear denoisers. *IEEE Transactions on Image Processing*.

Golub, G. H. and Van der Vorst, H. A. (2000). Eigenvalue computation in the 20th century. *Journal of Computational and Applied Mathematics*, 123(1-2):35–65.

Gonzalez, M., Almansa, A., Delbracio, M., Musé, P., and Tan, P. (2019). Solving inverse problems by joint posterior maximization with a vae prior. *arXiv preprint arXiv:1911.06379*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.

Goujon, A., Neumayer, S., Bohra, P., Ducotterd, S., and Unser, M. (2023). A neural-network-based convex regularizer for inverse problems. *IEEE Transactions on Computational Imaging*.

Gribonval, R. (2011). Should penalized least squares regression be interpreted as maximum a posteriori estimation? *IEEE Transactions on Signal Processing*, 59(5):2405–2410.

Gribonval, R. and Nikolova, M. (2020). A characterization of proximity operators. *Journal of Mathematical Imaging and Vision*, 62(6):773–789.

Gribonval, R. and Nikolova, M. (2021). On bayesian estimation and proximity operators. *Applied and Computational Harmonic Analysis*, 50:49–72.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. (2017). Improved training of wasserstein gans. *Advances in neural information processing systems*, 30.

Gunasekar, S., Woodworth, B., and Srebro, N. (2021). Mirrorless mirror descent: A natural derivation of mirror descent. In *International Conference on Artificial Intelligence and Statistics*, pages 2305–2313. PMLR.

Hasannasab, M., Hertrich, J., Neumayer, S., Plonka, G., Setzer, S., and Steidl, G. (2020). Parseval proximal neural networks. *Journal of Fourier Analysis and Applications*, 26:1–31.

Hauptmann, A., Mukherjee, S., Schönlieb, C.-B., and Sherry, F. (2023). Convergent regularization in inverse problems and linear plug-and-play denoisers. *arXiv preprint arXiv:2307.09441*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Helminger, L., Bernasconi, M., Djelouah, A., Gross, M., and Schroers, C. (2020). Blind image restoration with flow based priors. *arXiv preprint arXiv:2009.04583*.

Hertrich, J., Neumayer, S., and Steidl, G. (2021). Convolutional proximal neural networks and plug-and-play algorithms. *Linear Algebra and its Applications*, 631:203–234.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *arXiv preprint arXiv:2006.11239*.

Huang, L., Liu, X., Lang, B., Yu, A., Wang, Y., and Li, B. (2018). Orthogonal weight normalization: Solution to optimization over multiple dependent stiefel manifolds in deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Hurault, S., Chambolle, A., Leclaire, A., and Papadakis, N. (2023a). A relaxed proximal gradient descent algorithm for convergent plug-and-play with proximal denoiser. *arXiv preprint arXiv:2301.13731*.

Hurault, S., Ehret, T., and Arias, P. (2018). Epll: an image denoising method using a gaussian mixture model learned on a large set of patches. *Image Processing On Line*, 8:465–489.

Hurault, S., Leclaire, A., and Papadakis, N. (2022a). Gradient step denoiser for convergent plug-and-play. In *International Conference on Learning Representations*.

Hurault, S., Leclaire, A., and Papadakis, N. (2022b). Proximal denoiser for convergent plug-and-play optimization with nonconvex regularization. In *International Conference on Machine Learning*.

Hurault, S., Leclaire, A., and Papadakis, N. (2023b). Convergent bregman plug-and-play image restoration for poisson inverse problems. *Advances in Neural Information Processing Systems*.

Hyvärinen, A. and Dayan, P. (2005). Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4).

Jordan, R., Kinderlehrer, D., and Otto, F. (1998). The variational formulation of the fokker–planck equation. *SIAM journal on mathematical analysis*, 29(1):1–17.

Kamilov, U. S., Mansour, H., and Wohlberg, B. (2017). A plug-and-play priors approach for solving nonlinear imaging inverse problems. *IEEE Signal Processing Letters*, 24(12):1872–1876.

Karras, T., Aittala, M., Aila, T., and Laine, S. (2022). Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35:26565–26577.

Kim, K. and Ye, J. C. (2021). Noise2score: tweedie's approach to self-supervised image denoising without clean images. *Advances in Neural Information Processing Systems*, 34:864–874.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Krantz, S. G. and Parks, H. R. (2002). *A primer of real analytic functions*. Springer Science & Business Media.

Kurdyka, K. (1998). On gradients of functions definable in o-minimal structures. In *Annales de l'institut Fourier*, volume 48, pages 769–783.

Lan, G. and Zhou, Y. (2018). An optimal randomized incremental gradient method. *Mathematical programming*, 171(1):167–215.

Laumont, R., De Bortoli, V., Almansa, A., Delon, J., Durmus, A., and Pereyra, M. (2021). Bayesian imaging using plug & play priors: when langevin meets tweedie. *arXiv preprint arXiv:2103.04715*.

Levin, A., Weiss, Y., Durand, F., and Freeman, W. T. (2009). Understanding and evaluating blind deconvolution algorithms. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1964–1971.

Li, G. and Pong, T. K. (2016). Douglas–rachford splitting for nonconvex optimization with application to nonconvex feasibility problems. *Math. Progr.*, 159:371–401.

Li, H. and Lin, Z. (2015). Accelerated proximal gradient methods for nonconvex programming. *Advances in neural information processing systems*, 28:379–387.

Li, L., Hurault, S., and Solomon, J. (2023). Self-consistent velocity matching of probability flows. *Advances in Neural Information Processing Systems*.

Lim, B., Son, S., Kim, H., Nah, S., and Mu Lee, K. (2017). Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144.

Ma, K., Duanmu, Z., Wu, Q., Wang, Z., Yong, H., Li, H., and Zhang, L. (2017). Waterloo Exploration Database: New challenges for image quality assessment models. *IEEE Transactions on Image Processing*, 26(2):1004–1016.

Mairal, J., Bach, F., Ponce, J., and Sapiro, G. (2009). Online dictionary learning for sparse coding. In *Proceedings of the 26th annual international conference on machine learning*, pages 689–696.

Mairal, J., Ponce, J., Sapiro, G., Zisserman, A., and Bach, F. (2008). Supervised dictionary learning. *Advances in neural information processing systems*, 21.

Mallat, S. (2009). *A Wavelet Tour of Signal Processing, The Sparse Way*. Academic Press, Elsevier, 3rd edition edition.

Mallat, S. G. and Zhang, Z. (1993). Matching pursuits with time-frequency dictionaries. *IEEE Transactions on signal processing*, 41(12):3397–3415.

Mangoubi, O. and Vishnoi, N. K. (2019). Nonconvex sampling with the metropolis-adjusted langevin algorithm. In *Conference on Learning Theory*, pages 2259–2293. PMLR.

Martin, D., Fowlkes, C., Tal, D., and Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423.

Meinhardt, T., Moller, M., Hazirbas, C., and Cremers, D. (2017). Learning proximal operators: Using denoising networks for regularizing inverse imaging problems. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1781–1790.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. (2018). Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*.

Mokrov, P., Korotin, A., Li, L., Genevay, A., Solomon, J. M., and Burnaev, E. (2021). Large-scale wasserstein gradient flows. *Advances in Neural Information Processing Systems*, 34:15243–15256.

Moreau, J.-J. (1965). Proximité et dualité dans un espace hilbertien. *Bulletin de la Société Mathématique de France*, 93:273–299.

Möllenhoff, T., Strekalovskiy, E., Moeller, M., and Cremers, D. (2015). The primal-dual hybrid gradient method for semiconvex splittings. *SIAM Journal Imaging Sci.*, 8(2):827–857.

Nair, P. and Chaudhury, K. N. (2022). On the construction of averaged deep denoisers for image regularization. *arXiv preprint arXiv:2207.07321*.

Nair, P., Gavaskar, R. G., and Chaudhury, K. N. (2021). Fixed-point and objective convergence of plug-and-play algorithms. *IEEE Transactions on Computational Imaging*.

Nesterov, Y. (2013). Gradient methods for minimizing composite functions. *Mathematical programming*, 140(1):125–161.

Neumayer, S., Goujon, A., Bohra, P., and Unser, M. (2023). Approximation of lipschitz functions using deep spline neural networks. *SIAM Journal on Mathematics of Data Science*, 5(2):306–322.

Ng, M. K., Weiss, P., and Yuan, X. (2010). Solving constrained total-variation image restoration and reconstruction problems via alternating direction methods. *SIAM journal on Scientific Computing*, 32(5):2710–2736.

Ochs, P., Chen, Y., Brox, T., and Pock, T. (2014). ipiano: Inertial proximal algorithm for nonconvex optimization. *SIAM Journal Imaging Sci.*, 7(2):1388–1419.

Olshausen, B. A. and Field, D. J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325.

Pereyra, M. (2017). Maximum-a-posteriori estimation with bayesian confidence regions. *SIAM Journal on Imaging Sciences*, 10(1):285–302.

Perrin, M. S. (2020). *Semialgebraic Geometry*. PhD thesis, University of Sydney.

Pesquet, J.-C., Repetti, A., Terris, M., and Wiaux, Y. (2021). Learning maximally monotone operators for image recovery. *SIAM Journal Imaging Sci.*, 14(3):1206–1237.

Raginsky, M., Rakhlin, A., and Telgarsky, M. (2017). Non-convex learning via stochastic gradient langevin dynamics: a nonasymptotic analysis. In *Conference on Learning Theory*, pages 1674–1703. PMLR.

Reehorst, E. T. and Schniter, P. (2018). Regularization by denoising: Clarifications and new interpretations. *IEEE Transactions on Computational Imaging*, 5(1):52–67.

Repetti, A., Pereyra, M., and Wiaux, Y. (2019). Scalable bayesian uncertainty quantification in imaging inverse problems via convex optimization. *SIAM Journal on Imaging Sciences*, 12(1):87–118.

Rezende, D. and Mohamed, S. (2015). Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR.

Roberts, G. O. and Tweedie, R. L. (1996). Exponential convergence of langevin distributions and their discrete approximations. *Bernoulli*, pages 341–363.

Rockafellar, R. T. (1997). *Convex analysis*, volume 11. Princeton university press.

Rockafellar, R. T. and Wets, R. J.-B. (2009). *Variational analysis*, volume 317. Springer Science & Business Media.

Romano, Y., Elad, M., and Milanfar, P. (2017). The little engine that could: Regularization by denoising (red). *SIAM Journal Imaging Sci.*, 10(4):1804–1844.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer.

Rudin, L. I., Osher, S., and Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Phys. D*, 60:259–268.

Russell Luke, D., Thao, N. H., and Tam, M. K. (2018). Quantitative convergence analysis of iterated expansive, set-valued mappings. *Mathematics of Operations Research*, 43(4):1143–1176.

Ryu, E., Liu, J., Wang, S., Chen, X., Wang, Z., and Yin, W. (2019). Plug-and-play methods provably converge with properly trained denoisers. In *International Conference on Machine Learning*, pages 5546–5557.

Ryu, E. K. and Boyd, S. (2016). Primer on monotone operator methods. *Appl. Comput. Math*, 15(1):3–43.

Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D. J., and Norouzi, M. (2022). Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4713–4726.

Salim, A., Condat, L., Mishchenko, K., and Richtárik, P. (2022). Dualize, split, randomize: Toward fast nonsmooth optimization algorithms. *Journal of Optimization Theory and Applications*, 195(1):102–130.

Salim, A., Korba, A., and Luise, G. (2020). The wasserstein proximal gradient algorithm. *Advances in Neural Information Processing Systems*, 33:12356–12366.

Salimans, T. and Ho, J. (2021). Should EBMs model the energy or the score? In *Energy Based Models Workshop-ICLR 2021*.

Sanghvi, Y., Gnanasambandam, A., and Chan, S. H. (2022). Photon limited non-blind deblurring using algorithm unrolling. *IEEE Transactions on Computational Imaging*, 8:851–864.

Scheinberg, K., Goldfarb, D., and Bai, X. (2014). Fast first-order methods for composite convex optimization with backtracking. *Foundations of Computational Mathematics*, 14(3):389–417.

Scherzer, O., Grasmair, M., Grossauer, H., Haltmeier, M., and Lenzen, F. (2009). *Variational methods in imaging*, volume 167. Springer.

Setzer, S., Steidl, G., and Teuber, T. (2010). Deblurring poissonian images by split bregman techniques. *Journal of Visual Communication and Image Representation*, 21(3):193–199.

Shiota, M. (2012). *Geometry of subanalytic and semialgebraic sets*, volume 150. Springer Science & Business Media.

Song, J., Meng, C., and Ermon, S. (2020a). Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Song, Y. and Ermon, S. (2019). Generative modeling by estimating gradients of the data distribution. In *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems*.

Song, Y., Garg, S., Shi, J., and Ermon, S. (2020b). Sliced score matching: A scalable approach to density and score estimation. In *Uncertainty in Artificial Intelligence*, pages 574–584. PMLR.

Song, Y. and Kingma, D. P. (2021). How to train your energy-based models. *arXiv e-prints*, pages arXiv–2101.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2020c). Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

Sreehari, S., Venkatakrishnan, S. V., Wohlberg, B., Buzzard, G. T., Drummy, L. F., Simmons, J. P., and Bouman, C. A. (2016). Plug-and-play priors for bright field electron tomography and sparse interpolation. *IEEE Transactions on Computational Imaging*, 2(4):408–423.

Su, W., Boyd, S., and Candes, E. (2014). A differential equation for modeling nesterov's accelerated gradient method: theory and insights. *Advances in neural information processing systems*, 27.

Sun, T., Barrio, R., Cheng, L., and Jiang, H. (2018). Precompact convergence of the nonconvex primal–dual hybrid gradient algorithm. *Journal of Comp. Appl. Math.*, 330:15–27.

Sun, Y., Liu, J., and Kamilov, U. S. (2019a). Block coordinate regularization by denoising. *arXiv preprint arXiv:1905.05113*.

Sun, Y., Wohlberg, B., and Kamilov, U. S. (2019b). An online plug-and-play algorithm for regularized image reconstruction. *IEEE Transactions on Computational Imaging*, 5(3):395–408.

Sun, Y., Wu, Z., Xu, X., Wohlberg, B., and Kamilov, U. S. (2021). Scalable plug-and-play admm with convergence guarantees. *IEEE Transactions Computational Imaging*, 7:849–863.

Tang, M. and Repetti, A. (2023). A data-driven approach for bayesian uncertainty quantification in imaging. *arXiv preprint arXiv:2304.11200*.

Terris, M., Repetti, A., Pesquet, J.-C., and Wiaux, Y. (2020). Building firmly nonexpansive convolutional neural networks. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 8658–8662.

Themelis, A. and Patrinos, P. (2020). Douglas–rachford splitting and ADMM for nonconvex optimization: Tight convergence results. *SIAM Journal Optimization*, 30(1):149–181.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288.

Tikhonov, A. N. (1963). On the solution of ill-posed problems and the method of regularization. In *Doklady akademii nauk*, volume 151, pages 501–504. Russian Academy of Sciences.

Tseng, P. (2008). On accelerated proximal gradient methods for convex-concave optimization. *submitted to SIAM Journal Optimization*, 2(3).

Venkatakrishnan, S. V., Bouman, C. A., and Wohlberg, B. (2013). Plug-and-play priors for model based reconstruction. In *IEEE Glob. Conf. Signal Inf. Process.*, pages 945–948.

Vidal, A. F., De Bortoli, V., Pereyra, M., and Durmus, A. (2020). Maximum likelihood estimation of regularization parameters in high-dimensional inverse problems: An empirical bayesian approach part i: Methodology and experiments. *SIAM Journal on Imaging Sciences*, 13(4):1945–1989.

Vincent, P. (2011). A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674.

Wei, K., Aviles-Rivero, A., Liang, J., Fu, Y., Huang, H., and Schönlieb, C.-B. (2020). Tfpnp: Tuning-free plug-and-play proximal algorithm with applications to inverse imaging problems. *arXiv preprint arXiv:2012.05703*.

Whang, J., Lei, Q., and Dimakis, A. (2020). Compressed sensing with invertible generative models and dependent noise. In *NeurIPS 2020 Workshop on Deep Learning and Inverse Problems*.

Wibisono, A. (2018). Sampling as optimization in the space of measures: The langevin dynamics as a composite optimization problem. In *Conference on Learning Theory*, pages 2093–3027. PMLR.

Yuan, X., Liu, Y., Suo, J., and Dai, Q. (2020). Plug-and-play algorithms for large-scale snapshot compressive imaging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zeng, J., Lau, T. T.-K., Lin, S., and Yao, Y. (2019). Global convergence of block coordinate descent in deep learning. In *International conference on machine learning*, pages 7313–7323. PMLR.

Zhang, K., Gool, L. V., and Timofte, R. (2020). Deep unfolding network for image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3217–3226.

Zhang, K., Li, Y., Zuo, W., Zhang, L., Van Gool, L., and Timofte, R. (2021). Plug-and-play image restoration with deep denoiser prior. *IEEE Transactions on Image Processing*.

Zhang, K., Zuo, W., Chen, Y., Meng, D., and Zhang, L. (2017a). Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155.

Zhang, K., Zuo, W., Gu, S., and Zhang, L. (2017b). Learning deep CNN denoiser prior for image restoration. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3929–3938.

Zhang, K., Zuo, W., and Zhang, L. (2018). Ffdnet: Toward a fast and flexible solution for CNN-based image denoising. *IEEE Transactions on Image Processing*, 27(9):4608–4622.

Zhao, N., Wei, Q., Basarab, A., Dobigeon, N., Kouamé, D., and Tourneret, J.-Y. (2016). Fast single image super-resolution using a new analytical solution for $\ell_2 - \ell_2$ problems. *IEEE Transactions on Image Processing*, 25(8):3683–3697.

Zoran, D. and Weiss, Y. (2011). From learning models of natural image patches to whole image restoration. In *IEEE ICCV*, pages 479–486.