



UNLocBoX

NOMS ET MATRICULE DES MEMBRES :

BELL ARSÈNE KEVIN	22T2960
MELONG LETHYCIA	22W2147
DASSI MANDJO LEA JUSTINE	22W2164

Sous la supervision de Pr. MELATAGIA Paulin

Année Académique : 2025-2026

Structure de la présentation

- ➊ Introduction
- ➋ Les Méthodes de Splitting Proximal
- ➌ Architecture et Utilisation de la Boîte à Outils
- ➍ Avantages et Limites
- ➎ Conclusion

Qu'est-ce que l'Optimisation Convexe ?

- L'Optimisation Convexe est une branche de l'optimisation mathématique qui s'intéresse à la minimisation (ou maximisation) d'une fonction objective, convexe définie sur un ensemble convexe, avec des contraintes.

La facilité de résolution de ces problèmes est garantie par la propriété fondamentale : tout minimum local est équivalent à un minimum global.

- Elle est cruciale dans de nombreux domaines : **Apprentissage Automatique, Traitement des Images, Finance, etc.**
- Pour les problèmes modernes (big data, images haute résolution), s'imposent l'utilisation des méthodes spécifiques qui exploitent la structure convexe du problème, comme les méthodes de splitting.

Présentation d'UNLocBoX

- UNLocBoX (Matlab Convex Optimization Toolbox) est une boîte à outils MATLAB conçue pour résoudre les problèmes d'optimisation convexe de la forme additive :
$$x^* = \underset{x \in \mathbb{R}^N}{\operatorname{argmin}} \sum_{n=1}^K f_n(x)$$
 où les fonctions f_n sont des fonctions convexes.
- Elle implémente principalement des méthodes de splitting proximal (ou séparation proximale). Ces méthodes permettent de traiter séparément les fonctions f_n , ce qui est extrêmement efficace lorsque l'opérateur proximal de chaque fonction est facile à calculer.
- UNLocBoX se veut modulaire et proche de la formulation mathématique, permettant à l'utilisateur novice ou expert de combiner facilement des opérateurs proximaux existants pour prototyper rapidement de nouveaux algorithmes.

Concept de l'Opérateur Proximal

- L'Opérateur Proximal est l'outil central de cette approche. Pour une fonction convexe f , son opérateur proximal, noté $\text{prox}_f(y)$, est défini par :

$$\text{prox}_f(x) = \underset{y}{\operatorname{argmin}} \left(f(y) + \frac{1}{2} \|x - y\|_2^2 \right)$$

C'est la solution d'un sous-problème de minimisation simple. Il permet de trouver le point y qui minimise à la fois la fonction f et la distance au point donné x ; et ainsi de gérer la non-différentiabilité de f de manière efficace.

- L'avantage est Pour de nombreuses fonctions courantes (normes ℓ_1 , fonctions indicatrices de contraintes), l'opérateur proximal a une solution analytique simple et très rapide à calculer.

Principe du Splitting Proximal

UNLocBoX est conçu pour résoudre les problèmes d'optimisation de la forme :

$$\min_{x \in \mathbb{R}^N} f_1(x) + f_2(x)$$

Le principe de splitting (séparation) permet de contourner la difficulté de minimiser la somme ($f_1 + f_2$) en exploitant la facilité de minimiser chaque partie séparément via son opérateur proximal.

Au lieu de s'attaquer au problème global, l'algorithme alterne entre l'application de l'opérateur proximal pour f_1 et l'opérateur proximal (ou un pas de gradient) pour f_2 .

Ainsi, la résolution d'une fonction complexe se fait en la décomposant en de sous-fonctions plus simples, que l'on résout successivement et de manière répétée.

Les Composantes d'UNLocBoX

UNLocBoX est structurée autour de trois types de fonctions qui permettent de séparer la logique de l'algorithme de la définition du problème :

- Les Solvers (Solveurs) : C'est le cœur de la boîte à outils. Ce sont les fonctions qui implémentent les boucles itératives des algorithmes de splitting proximal (Douglas-Rachford, Forward-Backward, etc.). Ils gèrent la convergence et les critères d'arrêt.
- Les Opérateurs Proximaux : Une bibliothèque exhaustive de fonctions qui calculent l'opérateur $\text{prox}_f(y)$ pour les fonctions convexes les plus courantes (norme ℓ_1 , norme ℓ_2 , Total Variation (TV), etc.). Ces fonctions sont l'interface entre le solveur et la nature spécifique des termes du problème.
- Les Fichiers de Démonstration : Des exemples pratiques qui montrent comment combiner les solveurs et les opérateurs proximaux pour résoudre des problèmes classiques (débruitage, deconvolution, inpainting).
- Les fonctions utilitaires.

Implémentation d'un Problème Simple

Exemple : Régularisation ℓ_1

$$\min_x \frac{1}{2} \|y - Ax\|_2^2 + \lambda \|x\|_1$$

Définition des Fonctions :

- $f_1(x) = \lambda \|x\|_1$ (Le terme de régularisation non-lisse)
f1.name = 'proxL1'; // Nom de l'opérateur proximal
f1.lambda = lambda; // Paramètre de régularisation
- $f_2(x) = \frac{1}{2} \|y - Ax\|_2^2$ (Le terme d'erreur quadratique lisse)
f2.name = 'gradL2'; // Ici, on utilise le gradient de la fonction
f2.A = A;
f2.y = y;
f2.beta = 1; // Le pas de gradient

Résolution : L'appel au solveur est direct et reflète l'assemblage des composantes

```
param.maxit = 100;  
sol = solve_forward_backward(f1, f2, x0, param);
```


Gestion des Contraintes

UNLocBoX ne gère pas directement les contraintes fortes (du type $x \in \mathcal{C}$), mais utilise la fonction indicatrice pour les intégrer.

Principe : Une contrainte $x \in \mathcal{C}$ est transformée en une fonction convexe, $i_{\mathcal{C}}(x)$, définie par :

$$i_{\mathcal{C}}(x) = \begin{cases} 0 & \text{si } x \in \mathcal{C} \\ +\infty & \text{sinon} \end{cases}$$

L'opérateur proximal de cette fonction indicatrice est simplement la projection orthogonale sur l'ensemble \mathcal{C} (par exemple, la projection sur un ensemble de valeurs positives).

Solveurs

Les solveurs sont appelés avec la séquence d'arguments suivante :

- Le Point Initial (\mathbf{x}_0).
- Les Fonctions à Minimiser (fournies dans un tableau de cellules, ex. $\{f_1, f_2, f_3\}$).
- La Structure de Paramètres (param).

Le Solveur Automatique est `solvep()`. Elle sélectionne et exécute automatiquement l'algorithme de splitting le plus adapté en fonction du nombre et du type des fonctions fournies. Exemple : `sol = solvep(\mathbf{x}_0 , $\{f_1, f_2, f_3\}$, param);`
Pour un contrôle total, l'utilisateur peut appeler directement un solveur spécifique (ex. forward-backward).

Choix du solveur

Groupe	Caractéristique de la minimisation	Efficacité	Exemples
Spécifiques	deux fonctions ($f_1 + f_2$)	Plus Élevée	Douglas-Rachford, Forward-Backward, ADMM,
Généraux	plus de deux fonctions ($\sum f_i(x)$)	Moins Élevée	GFB, PPXA, SDMM

Le pas de temps est crucial pour l'équilibre entre vitesse et précision.

- Automatique : UNLocBoX le calcule automatiquement si le problème contient au moins une fonction différentiable (lisse), en utilisant sa constante de Lipschitz.
- Manuel : Si toutes les fonctions sont non-différentiables, l'utilisateur doit le définir via **param.gamma**. Une valeur plus faible de γ augmente la précision au détriment de la vitesse de convergence.

Avantages d'UNLocBoX

Table: Avantages Majeurs d'UNLocBoX

Catégorie	Description Détaillée	Impact
Performance et Scalabilité	Basé sur les méthodes de <i>splitting</i> proximal, évitant les inversions matricielles coûteuses. Gère efficacement la non-différentiabilité (ex: ℓ_1).	Efficacité supérieure pour les problèmes de grande dimensionnalité (<i>imagerie, big data</i>).
Modularité et Flexibilité	Séparation claire entre les Solveurs (logique itérative) et les Opérateurs Proximaux (fonctions spécifiques).	Permet le prototypage rapide de nouveaux algorithmes par simple assemblage.
Facilité d'Utilisation	Fonction <code>solvep()</code> pour la sélection automatique du solveur. Calcul automatique du pas de temps optimal pour les fonctions lisses.	Simplifie la traduction théorie-code et assure une convergence stable.
Problèmes Traités	Optimisé pour la forme $\min f_1(x) + f_2(x)$, idéale pour la régularisation (<i>parcimonie ℓ_1, TV</i>).	Excellent outil pour le traitement du signal et la reconstruction d'images .

Limites d'UNLocBoX

Table: Limites et Restrictions d'UNLocBoX

Restriction	Description Détaillée	Conséquence
Restriction de Convexité	Strictement conçu pour l'optimisation convexe .	Aucune garantie de convergence ou de solution pour les problèmes non-convexes .
Formulation (Additivité)	Optimisé pour la forme additive ($\sum f_i$).	Moins adapté aux objectifs ou contraintes avec des termes de couplage complexes non-séparables.
Faiblesse en Modélisation	Ne fournit pas de langage de modélisation de haut niveau (comme CVX) ni de vérification automatique de convexité.	L'utilisateur doit valider mathématiquement la convexité de sa formulation.

Conclusion

UNLocBoX est une boîte à outils incontournable en MATLAB pour l'optimisation convexe à grande échelle, en particulier dans le domaine du traitement du signal et de l'imagerie. Sa force réside dans son implémentation performante et modulaire des méthodes de splitting proximal, offrant un équilibre optimal entre flexibilité algorithmique et efficacité de calcul. Il permet aux chercheurs de traduire rapidement des concepts théoriques avancés en solutions pratiques et performantes.

Merci pour votre attention !