

# TPE : UNLocBox

Outil de Modélisation et Résolution d'Optimisation Convexe

Université de Yaoundé I  
Faculté des Sciences - Département d'Informatique

Octobre 2025

*L'optimisation convexe comme pilier de l'ingénierie et du Machine Learning*

## Membres du Groupe


**AZEUFACK NGNINWO THIERRY** Matricule : 12U0012

**NDEKEBAI MEYIE MICHAEL** Matricule : 21T2707

**DJAMPA MBIANGANG PLATINY CABREL** Matricule : 21T2437

**MOUGOU OWOUNDI BRICE WILLIAM** Matricule : 0000

# Plan de la Présentation

- 
- 1 Introduction à l'Optimisation Convexe
  - 2 Méthodes Proximales et Splitting
  - 3 UNLocBox : La Boîte à Outils
  - 4 Cas d'Usage et Implémentation
  - 5 Analyse et Comparaison
  - 6 Conclusion

# Qu'est-ce que l'Optimisation Convexe ?

## Définition

Minimiser une fonction objectif **convexe** sous des contraintes **convexes**.

### Avantage clé :

- Tout minimum local est un minimum global
- Garantie de convergence
- Algorithmes efficaces

### Applications :

- Machine Learning
- Traitement du signal
- Optimisation de portefeuille
- Vision par ordinateur

# Le Problème Type (Formulation)

## Formulation Générale

UNLocBox résout des problèmes de la forme :

$$\min_{x \in \mathbb{R}^N} \sum_{k=1}^K f_k(x)$$

## Caractéristiques :

- Chaque  $f_k$  est une fonction **convexe**
- Les fonctions peuvent être **lisses** et ou **non-lisses** (non différentiables)
- Combinaison de termes de fidélité aux données et de régularisation

## Exemple : Régression LASSO

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

# L'Approche UNLocBox : Proximal Splitting

## Principe des méthodes de Splitting

- **Diviser** un problème complexe en sous-problèmes plus simples
- Résoudre chaque sous-problème **séparément**
- **Combiner** les solutions

## L'Opérateur Proximal

Élément central des méthodes de splitting :

$$\text{prox}_f(y) = \arg \min_x \left\{ f(x) + \frac{1}{2} \|x - y\|^2 \right\}$$

Interprétation : Compromis entre minimiser  $f$  et rester proche de  $y$

# UNLocBox : Une Boîte à Outils Matlab

## Caractéristiques principales :

- Bibliothèque **open-source**
- Environnement : **Matlab/Octave**
- Framework **simple et modulaire**
- Implémentation d'algorithmes de pointe

## Objectif

Fournir un outil accessible pour résoudre des problèmes d'optimisation convexe non-lisse

**UNLocBox** = *UNconstrained LOw-complexity convex optimization BOX*

# Architecture : Les 3 Piliers de l'UNLocBox

**1. Solvers  
(Algorithmes)**

**2. Prox  
(Opérateurs  
Proximaux)**

**3. Grad  
(Gradients)**

*Framework Modulaire UNLocBox*

**Principe :** Assembler ces composants comme des *briques LEGO* pour construire votre solution d'optimisation



# Pilier 1 : Les Solveurs (Algorithmes)

## Algorithmes d'optimisation implémentés :

- **FB** (Forward-Backward)
- **FISTA** (Fast ISTA)
- **Douglas-Rachford**
- **ADMM**
- **Chambolle-Pock**
- **PPXA**
- **SDMM**
- **Generalized Forward-Backward**

## Comment choisir ?

Le choix du solveur dépend de :

- La structure du problème (nombre de fonctions  $f_k$ )
- Les propriétés des fonctions (lisse/non-lisse)
- La présence de contraintes

## Pilier 2 : Les Opérateurs Proximaux (prox)

### Rôle Central

$$\text{Résoudre : } \min_y \left\{ f(y) + \frac{1}{2} \|y - x\|^2 \right\}$$

### Fonctions gérées (exemples) :

- **Norme  $\ell_1$**  :  $\text{prox}_{\|\cdot\|_1}$  (soft-thresholding)  
*Usage : Parcimonie, sélection de variables*
- **Indicateur de convexe** : Projection sur un ensemble  
*Usage : Contraintes*
- **Norme  $\ell_2$**  : Projection sur une boule  
*Usage : Régularisation*
- **Norme nucléaire** : Pour matrices de rang faible  
*Usage : Complétion de matrices*

## Pilier 3 : Les Opérateurs de Gradient (grad)

**Rôle** : Calculer le gradient des fonctions **lisses**  $f(x)$

### Fonction Lisse vs Non-Lisse

- **Lisse** : Différentiable partout  $\rightarrow$  utiliser le gradient
- **Non-lisse** : Non différentiable  $\rightarrow$  utiliser le prox

### Exemple : Moindres Carrés

Pour  $f(x) = \frac{1}{2} \|Ax - b\|_2^2$  :

- Fonction lisse
- Gradient :  $\nabla f(x) = A^T (Ax - b)$
- Implémentation facile dans UNLocBox

## Cas d'Usage 1 : Denoising (Débruitage)

**Problème** : Restaurer un signal bruité

### Formulation Mathématique

$$\min_x \underbrace{\frac{1}{2} \|x - y\|_2^2}_{\text{Fidélité aux données}} + \underbrace{\lambda \|Dx\|_1}_{\text{Régularisation TV}}$$

**Modélisation UNLocBox :**

- $f_1(x) = \frac{1}{2} \|x - y\|_2^2 \rightarrow$  Gradient step (Lisse)
- $f_2(x) = \lambda \|Dx\|_1 \rightarrow$  Prox  $\ell_1$  (Non-lisse)

où :

- $y$  : signal observé (bruité)
- $D$  : opérateur de différences (capture les variations)
- $\lambda$  : paramètre de régularisation

# Mise en Œuvre Pratique (Code)

## Listing 1 – Exemple : Débruitage avec UNLocBox

```
% Parametres
param.verbose = 1;
param.maxit = 100;

% Fonction 1 : Attache aux donnees (lisse)
f1.eval = @(x) 0.5*norm(x - y)^2;
f1.grad = @(x) x - y;
f1.beta = 1;

% Fonction 2 : Regularisation L1 (non-lisse)
f2.eval = @(x) lambda*norm(D*x, 1);
f2.prox = @(x, T) prox_l1(D*x, lambda*T);

% Resolution avec Forward-Backward
sol = forward_backward(y, f1, f2, param);
```

# Avantages d'UNLocBox

## Simplicité

- Modélisation par **briques**
- Syntaxe claire
- Documentation complète

## Modularité

- Ajout facile de nouveaux solveurs
- Bibliothèque extensible
- Réutilisation de code

## Performance

- Algorithmes rapides
- Convergence garantie
- Adapté aux grands problèmes

## Flexibilité

- Problèmes variés
- Fonctions personnalisées
- Contrôle fin des paramètres

# Comparaison : UNLocBox vs CVX / Optimization Toolbox

Critère	UNLocBox	CVX / OptimTB
<b>Approche</b>	Méthodes proximales	Méthodes intérieures (CVX)
<b>Modélisation</b>	Par composants (briques)	Déclarative (DCP)
<b>Non-lissité</b>	Très adapté	Limité
<b>Grands problèmes</b>	Excellente scalabilité	Limité
<b>Vitesse</b>	Très rapide	Plus lent
<b>Facilité</b>	Courbe d'apprentissage	Plus intuitif (CVX)

## Verdict

UNLocBox est idéal pour les problèmes de grande taille avec régularisation non-lisse (ex :  $\ell_1$ , TV)

# Limites et Perspectives d'Évolution

## Limites Actuelles

- **Convexité stricte** : Non adapté aux problèmes non-convexes
- **Courbe d'apprentissage** : Nécessite de comprendre les méthodes proximales
- **Matlab/Octave uniquement** : Pas de version Python native
- **Documentation** : Certains exemples avancés manquent

## Perspectives

- Implémentation de nouveaux algorithmes (variance-reduced, stochastiques)
- **Optimisation GPU** pour accélérer les calculs
- Extension à certains problèmes non-convexes



# Conclusion

## Points clés à retenir :

- UNLocBox est un outil **puissant** pour l'optimisation convexe non-lisse
- Architecture **modulaire** basée sur 3 piliers (solvers, prox, grad)
- Particulièrement adapté aux problèmes avec régularisation  $\ell_1$ , **TV**
- Framework **simple** mais nécessite une compréhension des méthodes proximales

**UNLocBox : Une référence pour**  
l'optimisation convexe moderne

# Questions & Discussion

?

## Vos Questions

Discussion et échanges

*N'hésitez pas à poser vos questions*