

# Optimisation et problèmes inverses :

## TP1 - Descente de gradient

4 ETI – CPE Lyon

Travaux Pratiques Optimisation

2020-2021

**Noms, Prénoms : Dorian FABREGUE, Axel BRUYERE**

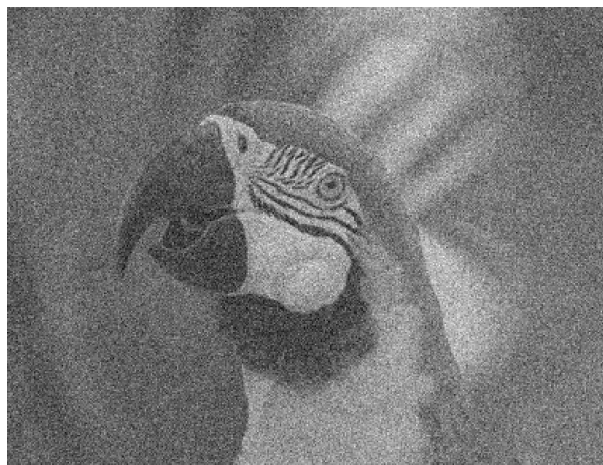
**Spécialité : IMI**

**Date : 07/03/2021**

- Objectifs** –
- Compréhension de l'algorithme de descente de gradient
  - Compréhension de l'apport de la méthode numérique par rapport à une résolution directe
  - Compréhension des limites de l'algorithme
  - Compréhension du modèle des moindres carrés
  - Compréhension du modèle de Tikhonov
  - Compréhension des limites des modèles
  - Implémentation de l'algorithme pour la minimisation des modèles des moindres carrés et de Tikhonov en 1D et en 2D

## I Contexte

On se concentrera dans ce TP sur des exemples en débruitage et déconvolution 1D (signal) et 2D (image). Pour autant, nous rappelons que l'optimisation est un domaine transversal, et permet, en fonction du choix de la fonction de coût, de traiter de nombreuses applications. En particulier, on veut dans ce TP débruiter l'image ci-dessous :



## II Prise en main de la méthode

Pour étudier le principe de la méthode de gradient, on se propose d'appliquer l'algorithme pour minimiser la fonction  $f(x) = x^2$ , i.e. résoudre le problème de minimisation 1D :

$$\hat{x} = \operatorname{argmin}_{x \in \mathbb{R}^N} \|x\|_2^2 \quad (1)$$

Nous allons pour cela mettre en place un algorithme de descente de gradient.

### II.1 Algorithme de descente de gradient

Le but d'un algorithme de descente est de construire une suite  $(x_n)_n$  qui converge vers la valeur  $\hat{x}$  souhaitée. Dans le cas de la descente de gradient, et grâce aux formules de développement de Taylor, chaque itération est de la forme :  $x_{k+1} = x_k - t_k \nabla f(x_k)$  avec  $t_k$  le pas de descente.

Cependant la descente de gradient doit vérifier plusieurs critères afin de garantir la convergence de la suite  $(x_k)_k$  vers  $\hat{x}$  :

- $f : \mathbb{R}^N \rightarrow ]-\infty, +\infty]$  convexe, propre et semi-continue inférieurement, de gradient  $\nu$ -Lipschitz,  $\nu \in ]0, +\infty[$ , tq  $\operatorname{argmin} f \neq \emptyset$
- $t_k$  une suite dans  $[\underline{t}, \bar{t}]$  tq  $0 \leq \underline{t} \leq \bar{t} \leq \frac{2}{\nu}$
- $x_0 \in \mathbb{R}^N$ .

On sait que la fonction norme  $f : x \rightarrow \|x\|_2^2$  vérifie facilement ces critères. Il nous reste à déterminer un pas  $t_k$  et de choisir un  $x_0 \in \mathbb{R}^N$ .

### II.2 Mise en pratique

Nous codons cette suite  $(x_k)_k$  sous Matlab, sachant que celle-ci doit normalement converger vers  $\hat{x} = \operatorname{argmin}_{x \in \mathbb{R}^N} \|x\|_2^2 = 0$ . Les valeurs de la dimension de  $X$ , du pas d'itération et de la précision exigée sont ici choisies de manière arbitraire.

```
df = @(x) 2*norm(x);
N=5;
X=randn([N 1]);

%Suite x_k
k = 0; pas = 0.4; aim = 1.0e-06; max_it = 10000;

while (abs(df(X))>aim && k<max_it)
    X= X - 2*pas*X;
    k = k + 1;
end
```

Nous avons obtenu les résultats suivants :

$X = [1.06 \times 10^{-7}; 7.44 \times 10^{-08}; -3.10 \times 10^{-08}; 3.00 \times 10^{-08}; -8.06 \times 10^{-08}] \approx \hat{x}$   
 $k = 10$  (compteur)

Un problème de cet algorithme réside dans un choix correct du pas de descente : si celui-ci est trop faible, le nombre d'itérations nécessaires sera trop important, et à l'inverse, si celui-ci est trop grand nous sortons des critères de convergence définis précédemment.

FIGURE 1 – figure

### II.3 Choix du pas de descente optimal

Précédemment, nous avons défini un pas de descente constant  $\forall k, t_k = 0,4$ . Nous allons chercher ici à déterminer une règle de calcul d'un  $t_k$  optimal proche de  $\frac{2}{\nu}$ .

En premier lieu, on peut penser à réaliser une recherche linéaire du  $t_k$  à chaque itération jusqu'à obtenir sa valeur optimale. On sait que la recherche linéaire est une méthode coûteuse, surtout quand celle-ci est répétée à chaque itération.

Afin de réduire ce coût, nous utiliserons la règle d'Armijo :  $\forall k$ , on choisit  $t_k$  et on teste si  $f(x_k - t_k \nabla f(x_k)) < f(x_k)$ , tant que ce n'est pas le cas, on choisit  $t_k \frac{t_k}{c}$

### III Introduction aux différentes fonctions de coût

Nous avons vu en cours que les fonctions de coût en image sont de la forme

$$f(x) = L(Hx; z) + \lambda R(x) \quad (2)$$

où  $L$  est l'attache aux données (choisie en fonction du type de dégradation) et  $R$  est une régularisation (choisie en fonction du type de résultat souhaité).

Nous allons ici étudier l'influence de chaque termes.

#### III.1 Modèle des moindres carrés

On se concentre dans un premier temps sur une fonction de coût sans régularisation (i.e.  $f(x) = L(Hx; z)$ ). Dans le cas d'une dégradation par un flou  $H$  et un bruit gaussien, le problème de minimisation est donc réduit au modèle des moindres carrés, défini par :

$$\hat{x} = \underset{x \in \mathbb{R}^N}{\operatorname{argmin}} ||Hx - z||_2^2 \quad (3)$$

On pourra se limiter dans un premier temps au choix  $H = Id$ . Une fois la méthode implémentée, on pourra introduire une dégradation  $H$  de type flou et/ou décimation.

#### III.2 Modèle de Tikhonov 1D

On souhaite étudier l'influence de la régularisation  $R$ . L'un des modèles les plus connus (utilisé par exemple pour la reconstruction tomographique) en traitement d'image est le modèle de Tikhonov, défini de la manière suivante :

$$f(x) = L(Hx; z) + \lambda ||\Gamma x||_2^2 \quad (4)$$

où  $\Gamma$  modélise un opérateur de notre choix (identité, gradient, Laplacien...).

#### III.3 Modèle de Tikhonov 2D

---

[code](#)

---

□

---

[résultats](#)

---

FIGURE 2 – figure

□

#### III.4 Analyse des résultats

# IV    Résolution du modèle de Tikhonov 1D

Insérer texte

## IV.1    Programmation et résultats

Insérer texte

\_\_\_\_\_ **code** \_\_\_\_\_



\_\_\_\_\_ **résultats** \_\_\_\_\_

FIGURE 3 – figure



## IV.2    Analyse des résultats

## V Résolution du modèle de Tikhonov 2D

Insérer texte

### V.1 Programmation et résultats

Insérer texte

---

**code**

---



---

**résultats**

---

FIGURE 4 – figure



### V.2 Analyse des résultats