

Perchais Clément

Maxime Dumont

\* Critique de choix du pas de descente ?

\* Manque l'itération de descente de gradient

## Optimisation et Problèmes inverses

### Introduction

Dans le cadre du module traitement d'image nous nous sommes intéressés aux problèmes inverses. On définit ce problème numérique comme trouver une solution  $z$  tel à partir de donnée d'observation. On dit qu'un problème est correctement posé s'il possède les propriétés suivantes :

- Pour toutes données ou conditions initiales il existe bien une solution noté  $z$
- Cette solution est unique

+ stabilité

Cette problématique est présente dans de nombreux domaines scientifiques par exemple la chimie, la physique et le traitement d'image. Une autre difficulté du problème est la quantité d'information que nous possédons pour adapter notre solution. Il existe donc déjà un problème pour la conception d'une solution générale à tous nos problèmes.

Nous posons le problème de la solution inverse de la manière suivante, supposant que nous avons un problème linéaire :

$$z = H \cdot \bar{x} + n$$

Avec  $z \in \mathbb{R}^m$  observation,  $\bar{x} \in \mathbb{R}^N$  la vérité terrain donc notre inconnue,  $H \in \mathbb{R}^{m \times N}$  et enfin  $n \in \mathbb{R}^m$  un bruit additif. Ainsi on cherche  $\hat{x}$  le plus proche possible de notre inconnu à partir des informations d'observations. En inversant le problème direct on a alors :

$$\hat{x} = \bar{x} + H^{-1} n$$

Notre objectif est donc maintenant d'optimiser notre solution pour cela nous allons tenter de minimiser l'écart entre  $\hat{x}$  et  $\bar{x}$  ce qui revient à minimiser la **fonction de coût** noté  $f$  tel que  $f : \mathbb{R} \rightarrow \mathbb{R}$  trouver  $\hat{x}$  tel que  $\forall x \in D f(\bar{x}) \leq f(x)$ . Ceci revient à résoudre

$$\hat{x} \in \operatorname{argmin} f(x)$$

Comme nous l'avons déjà évoqué il n'existe pas de cas général pour cela nous allons donc approcher un cas général avec différentes méthodes.

Dans un premier temps nous aborderons donc la méthode dites de descente de gradient, puis nous développerons le modèle des moindres carrés ainsi que celui de Tikhonov en une et deux dimensions. Dans l'ensemble de ce TP nous nous intéresserons à la théorie de ces modèles mais principalement à l'influence des différents paramètres.

⚠ Ne pas confondre :

① moindres carrés, Tikhonov, Polts, etc → fonctions coût

② descente de gradient : méthode (numérique) pour minimiser une fonction de coût

## Sommaire

- I- Prise en main de la méthode
- II- Résolution du modèle des moindres carrés
  - 1) Théorie
  - 2) Implémentation Algorithme
  - 3) Influence des paramètres *TB!*
- III- Résolution du modèle de Tikhonov 1D
  - 1) Théorie
  - 2) Implémentation Algorithme
  - 3) Influence des paramètres
- IV- Résolution du modèle de Tikhonov 2D
  - 1) Théorie
  - 2) Implémentation Algorithme
  - 3) Influence des paramètres

## Contexte

Comme nous avons pu le voir en introduction nous cherchons à optimiser un problème inverse à l'aide de différentes méthodes pour minimiser notre fonction de coût.

On regroupe l'ensemble de ces méthodes sous le nom de méthode de « **descente** ».

Le principe de ces méthodes est la suivante, nous partons d'un élément  $u_0$  donné à partir duquel nous construisons une suite  $(u_1 \dots u_n)$  destinée à converger vers la solution de notre problème.

On détermine le passage du rang  $n$  à  $n+1$  par une direction de descente notée  $d$  à laquelle on rajoute une condition de précision notée  $\epsilon$  lorsque qu'on approche notre solution.

Par définition  $d$  est une direction de descente de  $f$  en  $x \in \mathbb{R}$   $f'(x) \cdot d < 0$  ainsi une telle direction convient bien à notre optimisation car il suffit de se déplacer le long de  $d$  pour que  $f$  décroisse et donc qu'on obtienne

$$\hat{x} \in \operatorname{argmin} f(x)$$

Nous allons dans la suite de ce cours implémenter nos algorithmes reposant sur aspects importants:

- La nature de notre direction  $d$  qui donne le nom à notre algorithme
- Définition du pas entre chaque itération permettant d'optimiser le résultat

Nous allons donc maintenant aborder notre première méthode de descente qu'est la descente de gradient.

## I- Prise en main de la méthode

### 1) Théorie

Dans un premier temps nous allons utiliser la méthode dite de descente de gradient pour une fonction de coût telle que :

$$\begin{aligned} f: \mathbb{R} &\rightarrow \mathbb{R}^2 \\ x &\rightarrow x^2 \end{aligned}$$

On se place donc dans un contexte simple en une dimension sans présence de contrainte tel qu'un bruit, on cherche ainsi à trouver  $x \in \mathbb{R}$  tel que  $x \in \operatorname{argmin} x^2$ .

On définit la direction de la manière suivante  $d_k = -\lambda \nabla f(x_k)$   $\lambda \in \mathbb{R}$  on appelle aussi cette direction, direction de la plus profonde descente. Ce terme s'explique par le fait que si le gradient est non nul alors la direction est parallèle à la solution du problème. On note aussi que  $d$  existe si  $x_k$  n'est pas un point stationnaire.

### 2) Implémentation Algorithmique

Dans un premier nous initialisons nos paramètres que sont :

- L'influence de descente du gradient  $\lambda$
- Le nombre d'itération
- Notre élément de départ notre  $x_0$
- Notre taux d'erreur entre la solution et la valeur attendue

Puis on incrémente notre élément de départ par notre direction  $d_k$ , que l'on superpose à notre courbe

Code :

```
for i=1:1:nb_iteration
    % définition de la fonction
    f= X0.^2;
    % calcul de la dérivée partielle
    df= 2*X0;
    % calcul descente
    deltax=-lambda*df;
    X0=X0+deltax;
    figure(1)
    plot( X0,f, '*')
    hold on;
end
f= x.^2;
figure(1)
plot(x,f);
title('Descente de gradient de x^2');
```

### 3) Influence des paramètres

Dans un premier temps nous allons étudier le nombre d'itération <sup>5</sup> on s'attend à obtenir un résultat satisfaisant à partir d'un certain nombre d'itération.

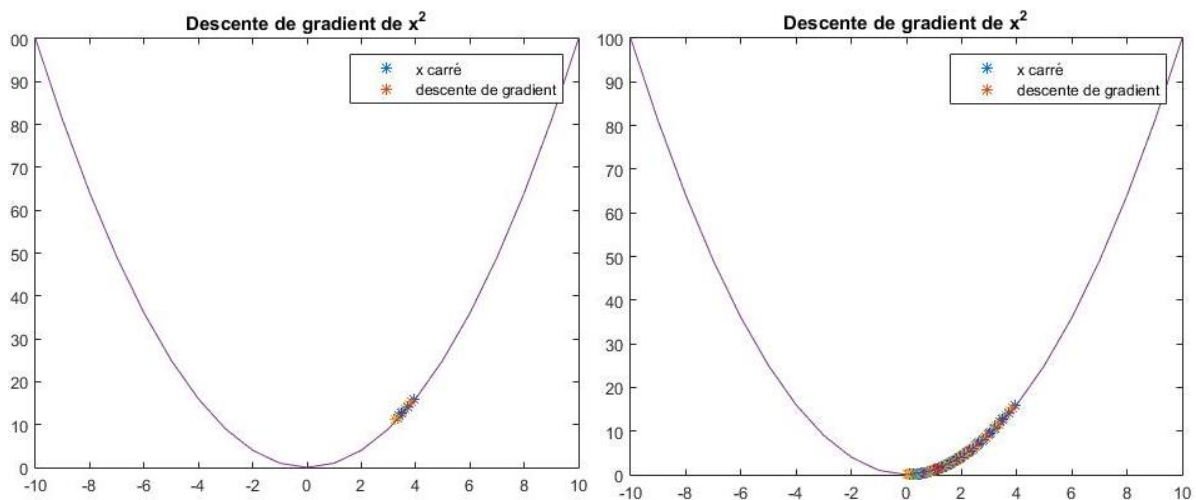


Figure 1 : Influence du nombre d'itération pour 10 et 300 itérations

D'après ces résultats on se fixera donc un nombre d'itération suffisamment grand pour obtenir des résultats concluant ici de 300.

Intéressons-nous maintenant à notre facteur de descente de gradient  $\lambda \in \mathbb{R}$ . Dans un premier temps nous allons partir de  $\lambda = 1$  puis nous diminuons sa valeur.

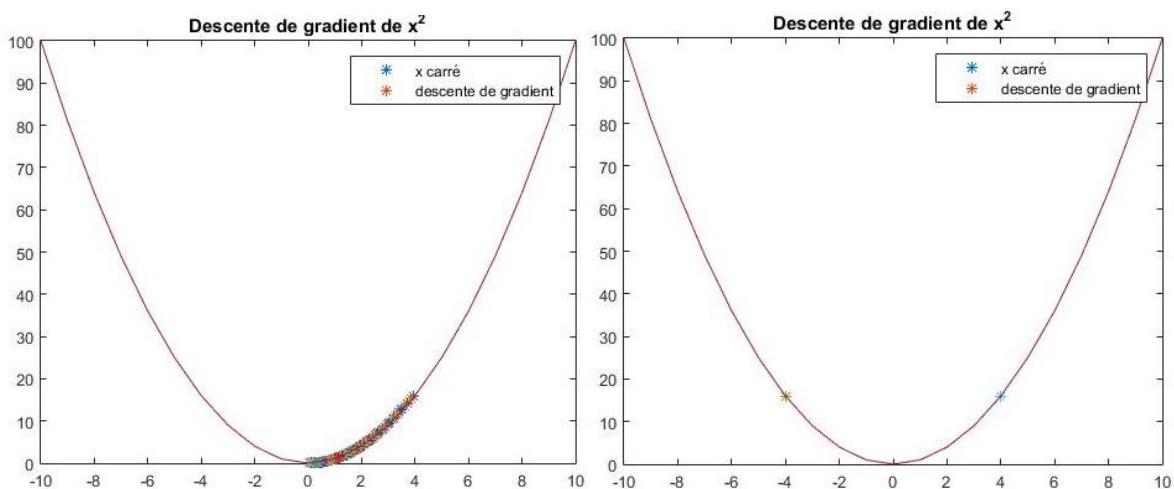
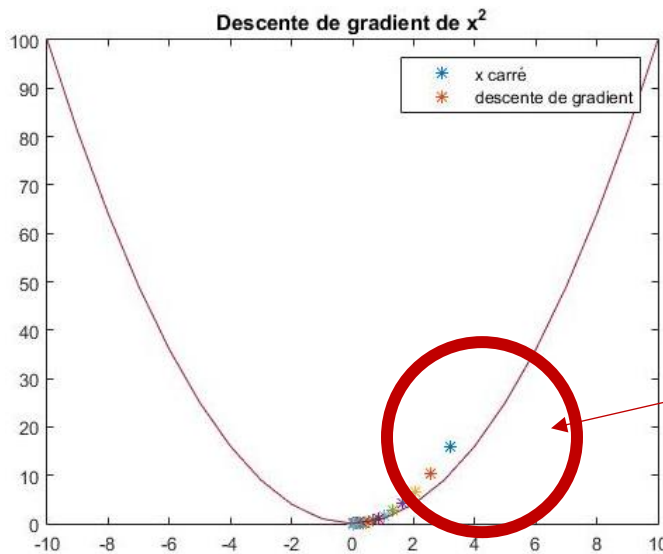


Figure 2 : Influence du facteur de descente  $\lambda$  pour 0.01 et 1

Comme on peut le constater dès que  $\lambda \geq 1$  on observe alors une divergence de la descente de gradient et un changement de descente. Plus notre  $\lambda$  est grande plus cette divergence est prononcée. On rajoute aussi un aspect de l'influence de  $\lambda$ , ce dernier doit être très proche de 0 car

*Revoir d'interprétation*

sinon notre descente est très approximative sur les premières itérations comme le montre l'exemple suivant :



Zone d'approximation dû à une valeur de lambda non concluante

Nou, je pense qu'il y a un problème de code.  $\lambda = 0.1$  ne doit pas faire ça

Figure 3 : Influence du facteur de descente  $\lambda$  pour 0.1

Ainsi en conclusion de l'influence de ce facteur on se doit de faire un compromis entre une valeur suffisamment petite pour éviter une divergence mais aussi obtenir une très petite approximation.

On peut aussi rajouter les paramètres de points de départ et du taux d'erreur pour trouver la solution finale. Concernant le point de départ son influence se limite au nombre d'itération qui doit être suffisant pour obtenir la solution finale et donc dépend forcément de l'influence de la descente de gradient. Ainsi la descente de gradient est le paramètre le plus influant permettant d'optimiser notre résultat.

## II- Résolution du modèle des moindres carrés

### 1) Théorie

Pour cette méthode nous ~~repren~~ le principe de la partie précédente, cette fois ci nous modifions notre fonction cout de la manière suivante :

$$f(x) = \mathcal{L}(Hx; z) + \lambda R(x)$$

Avec  $\mathcal{L}$  l'attache aux données et  $R$  est une régularisation. Cette forme s'approche d'un modèle que l'on rencontre dans la manipulation d'image.

Pour commencer nous allons prendre uniquement en compte l'attache aux données, donc notre fonction de cout est :

$$f(x) = \mathcal{L}(Hx; z)$$

Ainsi on cherche à minimiser le système suivant :

$$\operatorname{argmin}(\|Hx - z\|_2^2)$$

On peut noter qu'ici on reformule un problème inverse comme un problème d'optimisation.

On part du principe au début de prendre H comme une matrice identité on s'attend alors à retrouver le même signal estimé que le signal observé

Ainsi cette fois ci notre direction est égale à :  $d_k = -\lambda \nabla f(x_k)$  tel que

$$\nabla f(x_k) = \nabla \|Hx - z\|_2^2 = 2H^T(Hx - z)$$

## 2) Implémentation de l'Algorithme

On implémente la méthode de la manière suivante, on modifie cette fois la direction.

Code :

```
xBAR=sin(-10:0.1:10);
xBAR=xBAR';
H=matH_1D(length(xBAR),5,'gaussian',20);
z=H*xBAR + 0.01*rand(201,1);
figure(2)
plot(-10:0.1:10,z,'b');
X0=zeros(length(xBAR),1);
dExpression_analytique=2*H'*(H'*X0-z);
Expression_analytique=norm(H*X0-z,2)^2;
lambda=1;
for i=1:1:1000
    deltax=-lambda*dExpression_analytique;
    X0=X0+deltax;
    Expression_analytique=norm(H*X0-z,2);
    dExpression_analytique=2*H'*(H'*X0-z);
end
hold on;
title('Descente de gradient moindre carré');
plot(-10:0.1:10,X0,'r');
legend('sinus','moindres carrés');
```

On note que l'on prend pour expression xbarre un sinus variant de -10 à 10 .

Dans le cas de H comme matrice identité on s'attend à obtenir le même signal estimé que le signal observé ce qui nous intéresse ici est de « flouté » notre signal, pour cela on admet H étant inversible.

## 3) Influence des paramètres et résultats

Dans un premier temps nous allons comparer nos deux méthodes pour un même signal 1D, un sinus par exemple :

Revoir

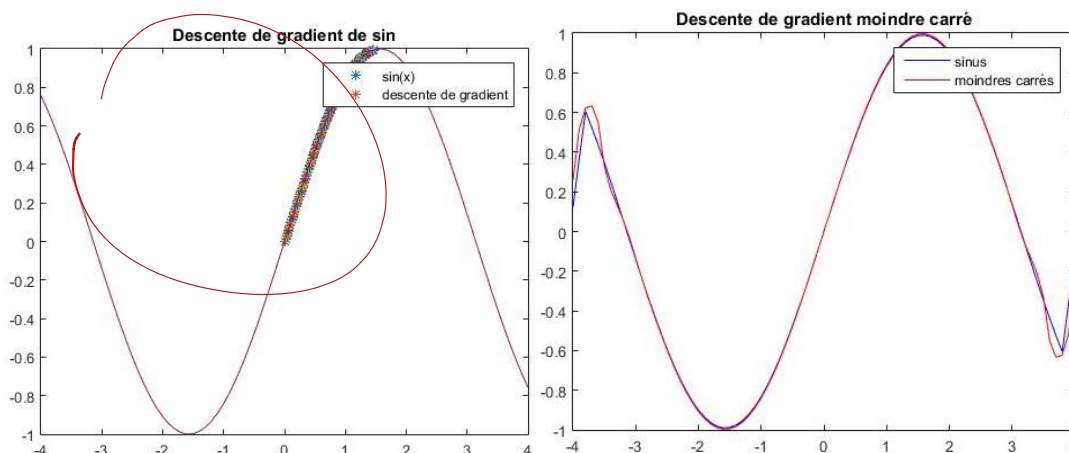


Figure 4 : Comparaison des modèles de descente de gradient et moindres carrés

Pour ces deux résultats nous avons utilisés les mêmes paramètres pour pouvoir effectuer une comparaison :  $X_0 = -1$ . On note la limite de la première méthode par rapport à la valeur du  $\lambda$  mais

Dans un premier temps nous allons mettre en évidence l'influence des conditions de l'observation

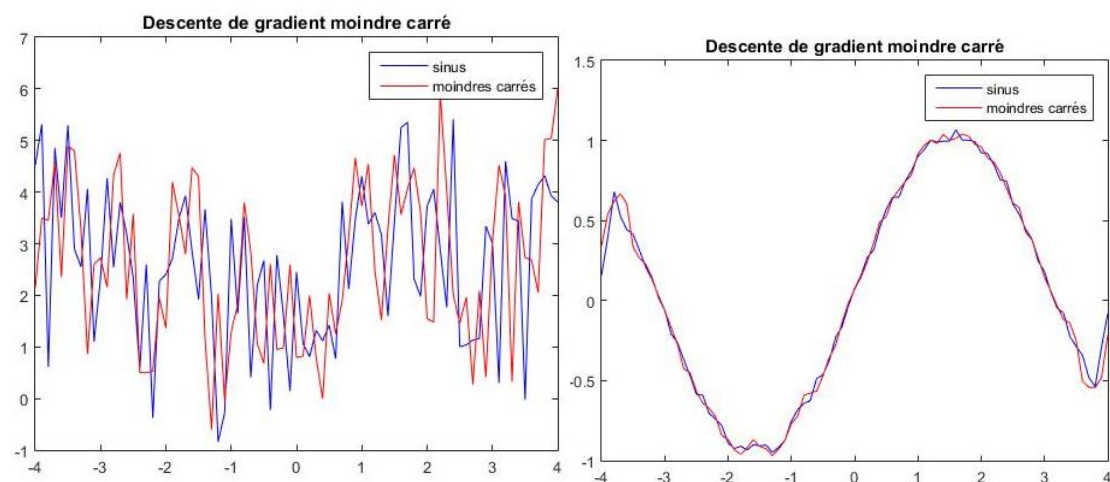


Figure 5 : Influence du bruit pour une amplitude 5 et 0.1

On observe donc bien que le résultat de la méthode des moindres carrés dépend bien des conditions d'observation, mais ce résultat était prévisible.

Enfin intéressons-nous à la notion de conditionnement de la matrice  $H$ . Pour cela nous allons rajouter un flou de type gaussien. Cette opération modifie la propriété d'invisibilité de notre matrice  $H$ . Ainsi notre résultat  $\hat{x} = \bar{x}$  devient  $\hat{x} = H^{-1}z$ , c'est ce que l'on peut remarquer ci-dessous :

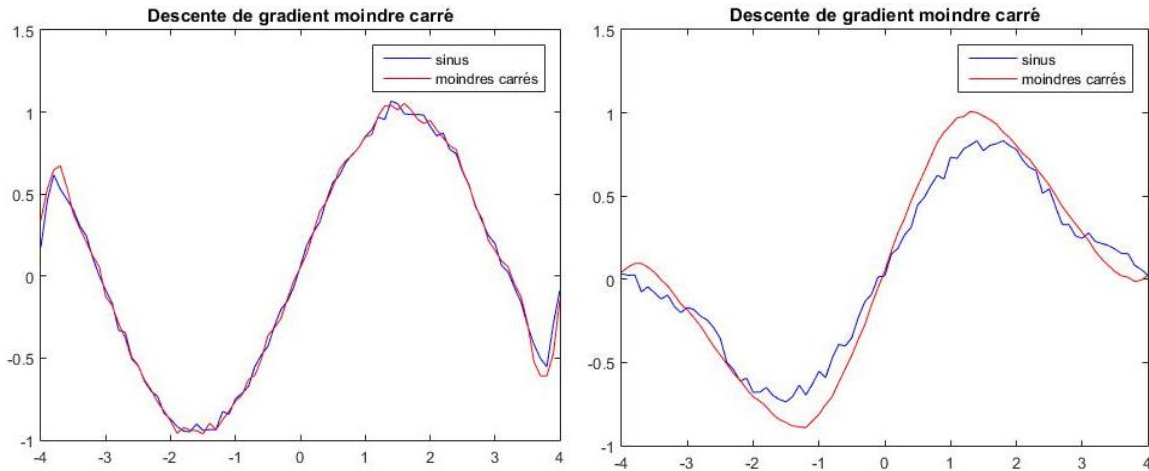


Figure 6 : Influence du flou pour un écart type de 5 et 25

On note donc que la solution analytique ce n'est pas satisfaisante et que la descente de gradient n'approche pas la solution analytique. En visualisant notre fonction cout on s'aperçoit que l'on n'obtient pas les résultats attendus. On note un problème de surapprentissage lorsque notre matrice  $H$  est mal dimensionnée donc la régularité du résultat très dépendant du noyau  $H$ .

Ainsi il est obligatoire de réguler notre matrice  $H$ . Nous allons maintenant aborder la régularisation de Tikhonov en 1 D puis en 2 D pour répondre aux limites des moindres carrés et obtenir  $\bar{x}$  et non plus  $H^{-1}z$ .

### III- Résolution du modèle de Tikhonov 1D

#### 1) Théorie

Comme nous l'avons évoqué nous allons rajouter la notion de régularisation et transformer notre problème inverse en un problème d'optimisation de la forme suivante :

$$f(x) = \mathcal{L}(Hx; z) + \lambda R(x)$$

On utilisera un modèle connu sous le nom de modèle de Tikhonov défini de la manière suivante :

$$\operatorname{argmin} ( \|Hx - z\|_2^2 + \lambda \|\Gamma x\|_2^2 )$$

On a donc une régularisation de la forme  $\|\Gamma x\|_2^2$  avec  $\Gamma$  un opérateur de notre choix : identité , gradient , La placcien...

On s'attend donc théoriquement avec ce modèle à obtenir un résultat non plus égal à  $H^{-1}z$  mais bien  $\bar{x}$ .

Pour l'étude de ce modèle nous avons repris l'étude de notre signal 1D qu'est le sinus. Nous verrons dans cette partie l'influence de l'opérateur de régularisation ainsi que l'influence du paramètre de régularisation  $\lambda$ .



## 2) Implémentation Algorithmme

Code :

```
%% Tikhonov 1D

xBAR=sin(-10:0.1:10);xBAR=xBAR';
H=math_1D(length(xBAR),5,'gaussian',10);
z=H*xBAR + 0.01*rand(201,1);
figure(3)
plot(-10:0.1:10,z,'b');
X0=zeros(length(xBAR),1);
lambdaT=1;
Expression_analytique=2*H'*(H'*X0-z) +
lambdaT*del2(norm(gradient(X0),2).^2);
lambda=1;

for i=1:1:1000
    deltax=-lambda*Expression_analytique;
    X0=X0+deltax;
    Expression_analytique=2*H'*(H'*X0-z) +
lambdaT*gradient(norm(gradient(X0),2).^2);
end
hold on;
title(' Tikhonov - Laplacien');
plot(-10:0.1:10,X0,'r')
legend('sinus','Thikonov');
```

*Explication*

### 3) Influence des paramètres

Comme nous l'avons déjà évoqué ce modèle de régularisation repose principalement sur deux paramètres que sont

- Le paramètre de régularisation  $\lambda$
- L'opérateur régularisant  $\Gamma$

Ainsi nous allons étudier ces deux paramètres.

#### Choix de l'opérateur régularisant $\Gamma$

Nous allons essayer différents opérateurs partant du principe que pour un problème inverse donné il existe un opérateur nous permettant d'optimiser notre résultat. Nous allons discuter dans le cas de notre signal 1D du choix de l'opérateur.

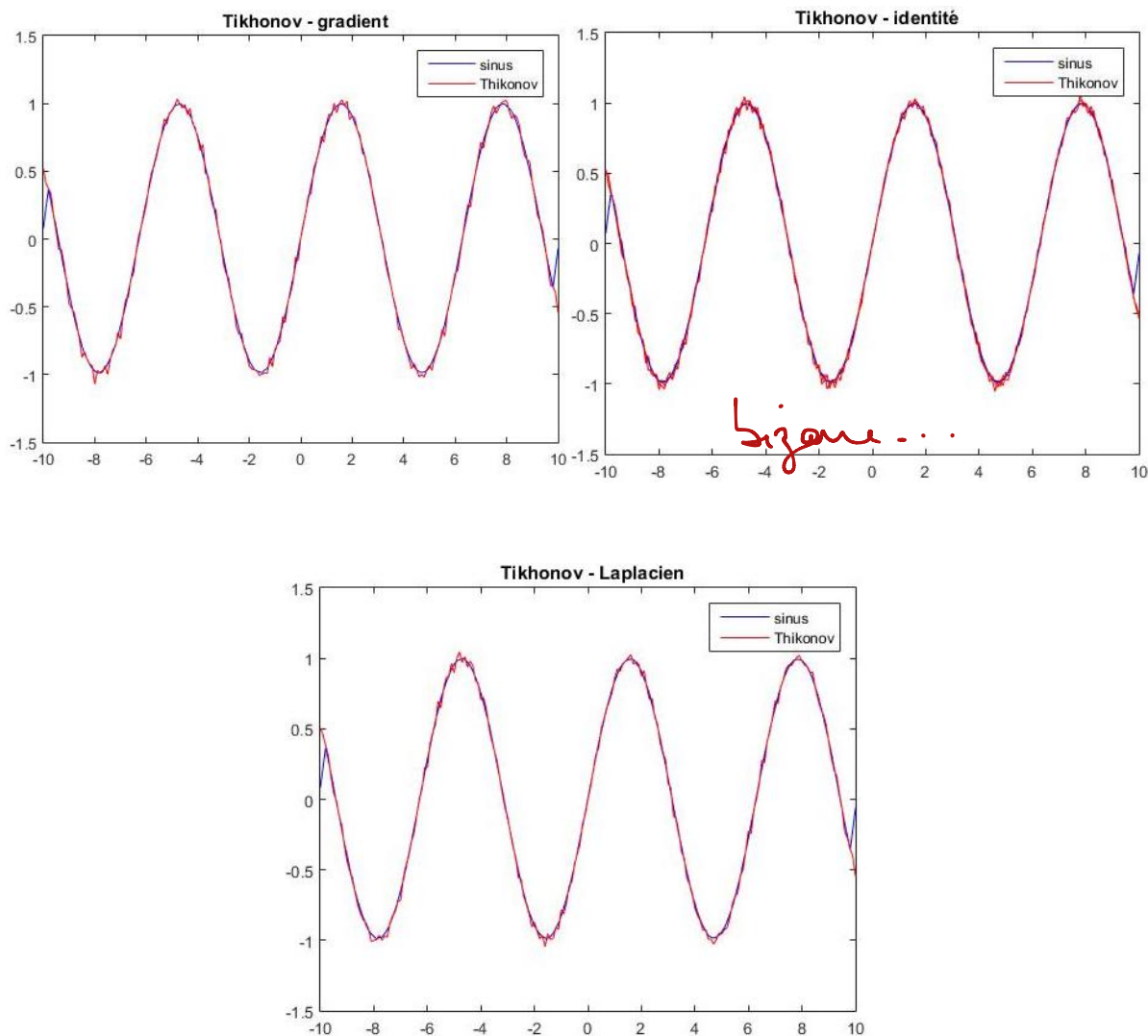


Figure 7 : Influence de l'opérateur de régulation

Comme on le constate les résultats diffèrent pour différents opérateurs et pour une même valeur de  $\lambda=1$ . On note que l'opérateur gradient nous permet d'approcher idéalement la solution analytique.

Pourquoi ?

### Choix du paramètre de régularisation $\lambda$

Une difficulté des méthodes de régularisation réside dans le choix de la valeur du paramètre  $\lambda$ . Il existe une grande variété de méthodes permettant d'optimiser ce choix reposant sur une étude attentive du problème posé.

L'idée consiste à mettre de côté l'une des données du problème et à considérer que la valeur optimale de  $\lambda$  conduit à une bonne approximation de cette valeur. Le  $\lambda$  choisit sera alors celui qui optimise cette approximation.

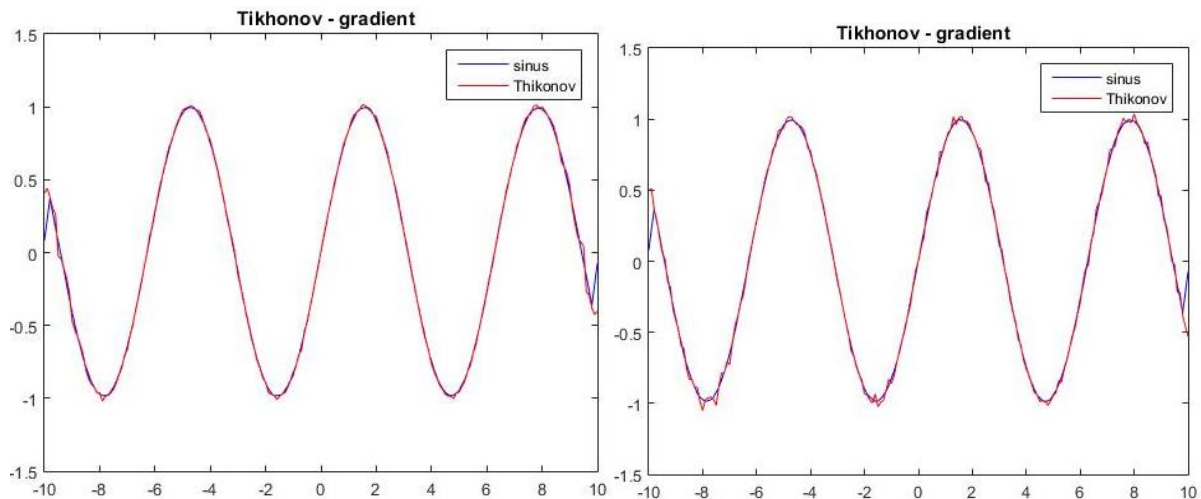


Figure 8 : Influence du paramètre de régularisation  $\lambda = 0.03$  et  $\lambda = 0.8$

On note que plus notre  $\lambda$  doit être positif inférieur à 1 pour éviter la divergence comme pour les cas précédents. On peut rajouter que sa valeur doit être un compromis pour tendre vers 0 de l'ordre de  $10^{-2}$  mais doit être suffisamment grand donc entre 0.1 et  $10^{-2}$ .



## Notes

Si  $H$  est identité on retrouve exactement le même signal estimé que le signal observé, si  $H$  est un flou par exemple alors sous réserve d'inversibilité on veut  $\hat{x} = H^{-1}z$  or si  $z = H \bar{x}$  alors  $\hat{x} = \bar{x}$  mais ce cas est très générale ce qui n'est pas un cas réel. Si non inversible alors amplification du bruit  $n$ .

$H$  mal conditionnée  $\Rightarrow$  solution analytique non satisfaisante la descente de gradient n'approche pas la solution analytique.

Visualiser la fonction cout ! car si la fonction cout n'est pas linéaire alors on peut avoir une remontée de la distance entre le résultat attendu et observé. Attention entre ce qu'on veut et ce que la méthode nous donne, besoin d'une régularisation (Tikonov) pour bien obtenir  $\bar{x}$  et non plus  $H^{-1}z$

Tikonov n'est pas forcément la solution au sur apprentissage.

CCI générale quels sont les avantages de la descente de gradient face à la solution analytique.

Notes et Remarques :

Bien apprendre les notations :  $h$  = flou ,  $z$  vérité terrain  $\hat{x}$  est l'approximation idéal . Bien séparé le problème d'optimisation (argmin) du problème inverse

Optimisation = existence unicité stabilité

Note pour les moindres carrés : problème de surapprentissage quand un  $H$  mal dimensionné donc la régularité du résultat très dépendant du noyau  $H$ .

Conclusion sur les moindres carrés : marche mal et problème de conditionnement

oui .