# INTRODUCTION

Après avoir étudié dans le chapitre 1 l’étude du problème de Tikhonov :

Dans ce chapitre nous allons plutôt étudiez ce problème dans le cas de fonctions non différentiables ou encore avec un terme de régulation R(x) non différentiable.

Pour simplifier et sachant que l’on étudie le terme de régulation on va mettre H = id pour appliquer seulement un bruit

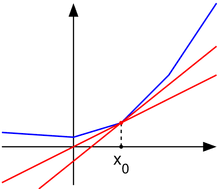
Cela se résume au problème d’optimisation suivant :

Nous allons tout d’abord nous concentrer sur le terme de régulation donc au problème d’optimisation :

# I théorie des concepts

## Sous différentielles

Nous allons utiliser une généralisation des différentielles pour les fonctions continues non différentiables. C’est pourquoi on introduit la notion de sous différentielle :



Visuellement les sous différentielles sous équivalente au gradient pour les fonctions différentiables. Cependant pour les fonctions non différenciables il y a quelque point où il y a une infinité de tangentes et la sous-différentielle représente cet ensemble.

On peut avec cette notion réaliser un algorithme de sous gradient qui choisit arbitrairement une différentielle parmi l’ensemble des sous différentielles.

Bien que cet algorithme possède l’avantage de pouvoir passer les points non dérivables. Cet algorithme possède tout de même des limites semblables à l’utilisation de descente de gradient (choix de lambda…) mais il y a de plus un risque de non convergence dû au à la non continuité de la dérivée.

## Enveloppe de Moreau

De plus il existe aussi la notion d’enveloppe de moreau qui peut être calculée de la manière suivante :

Celle-ci possède plusieurs propriétés utiles au contexte de minimisation, en effet celle-ci possède le même minimum que f et est ce qui permet par exemple d’appliquer l’algorithme de descente de gradient vu dans le 1er chapitre. De plus cette enveloppe est une fonction propre et est convexe.

## Opérateur proximal

Le calcul de cette enveloppe de moreau se calcule à l’aide de l’opérateur proximal :

Le calcul de ce proximal se fait à la main pour éviter la complexité d’une résolution numérique.

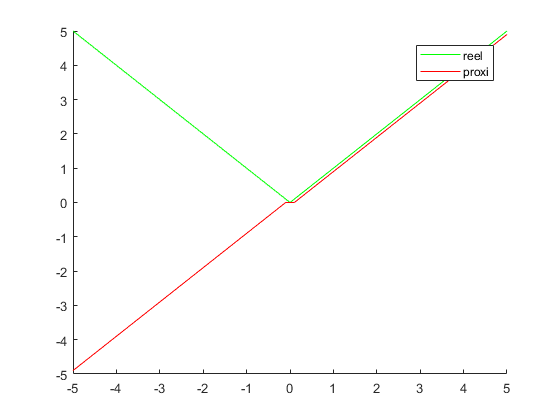
De plus on peut retrouvez l’enveloppe de moreau à l’aide de la formule suivante :

## Exemple de fonctions

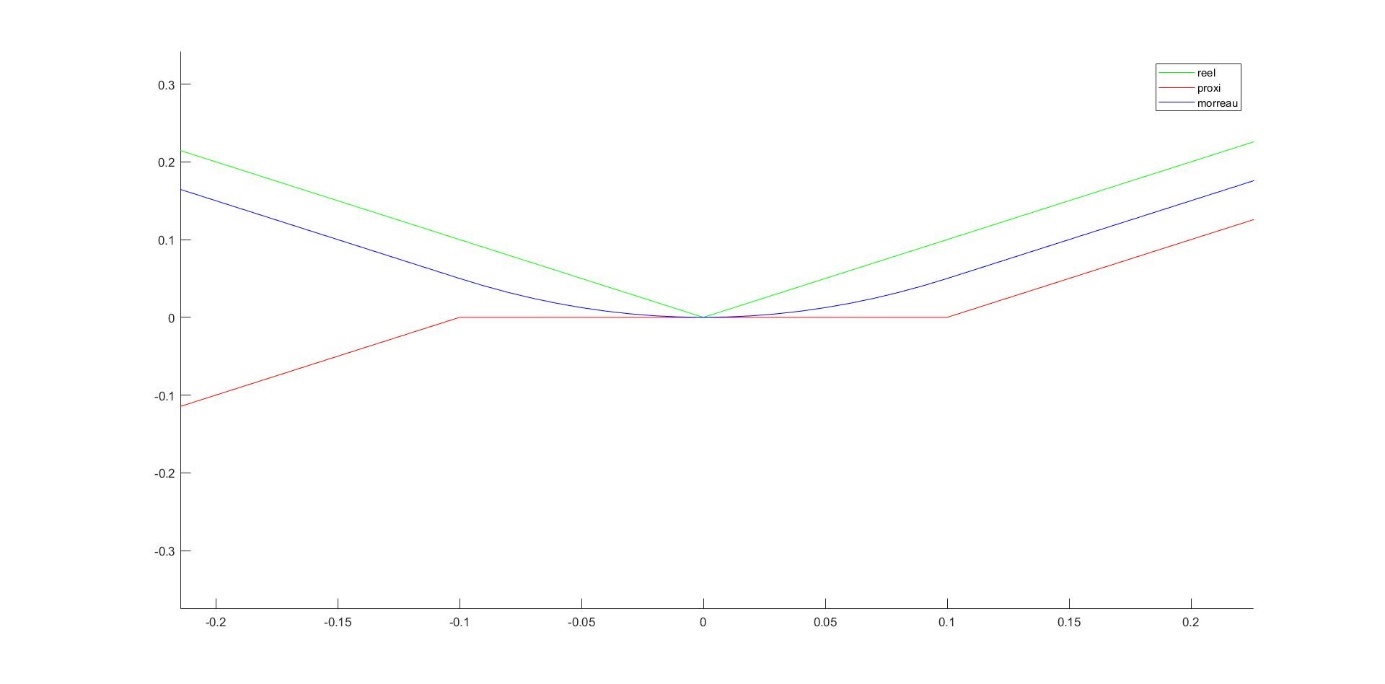
Pour la fonction abs (x) on trouve le proximal égale à

Pour f(x) = |x|

X



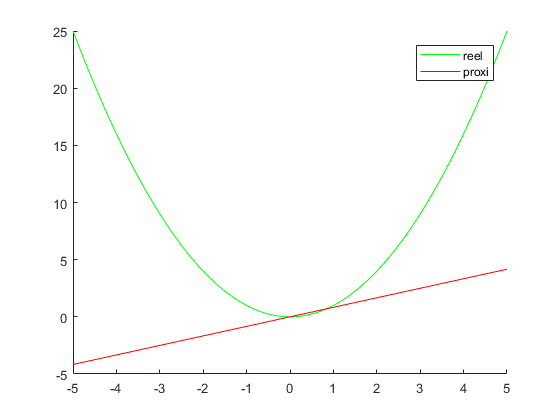
Ainsi que l’enveloppe de moreau suivante :



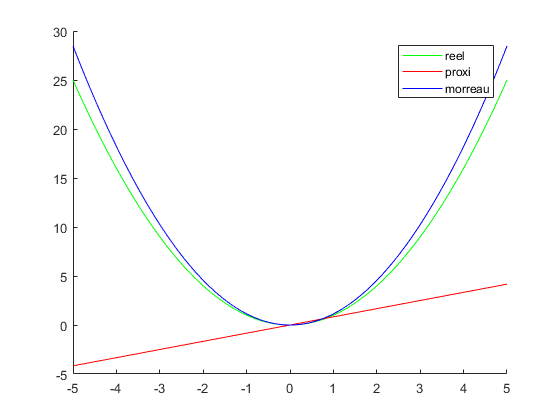
On peut donc remarquer que l’enveloppe de moreau possède bien le même minimum que la fonction f et elle semble .

Pour f(x) = x² on trouve le proximal suivants :

X



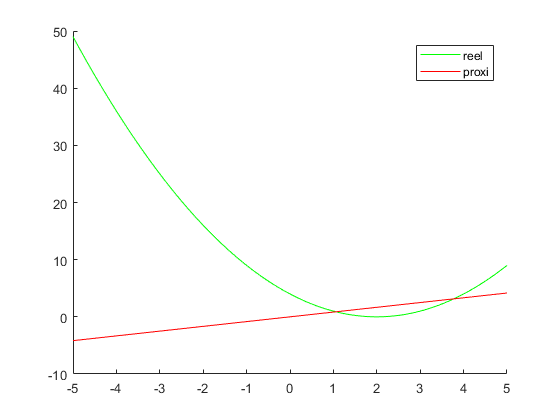
Et l’enveloppe suivante :



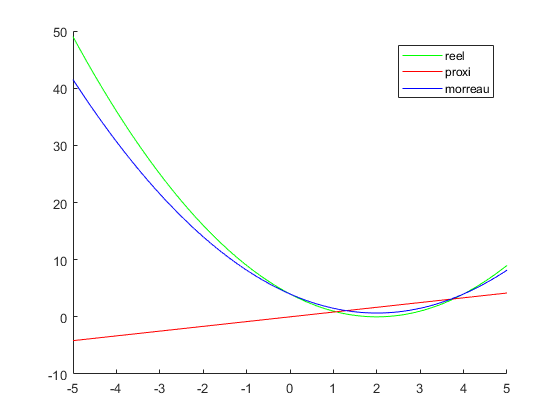
Ici bien que la fonction soit dérivable on peut voir que l’enveloppe et le proximal peuvent toujours être calculés et sont cohérent car l’enveloppe possède le même minimum.

Pour f(x) = (x-z)²

X



Nous trouvons le même proximal que pour x² ce qui montre que le proximal n’est pas sensible à la translation.

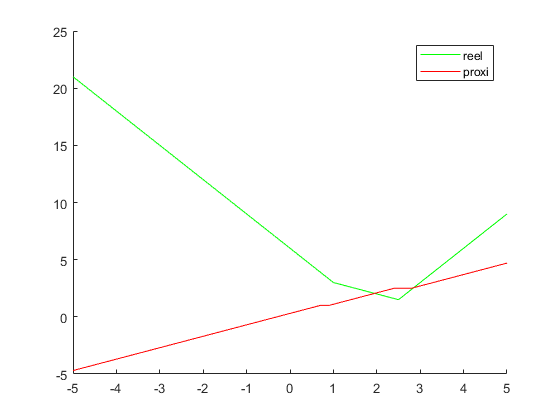


Cependant on peut noter que l’enveloppe de moreau est bien sensible à la translation car elle a été translatée tout comme la fonction f.

On peut aussi noter qu’il semble que le minimal de l’enveloppe de moreau diffère de celui de f mais cela est dû à une trop forte approximation. En effet ici gamma = 0.1 mais si on réduit ce même gamma on obtient une enveloppe très proche de f et donc avec le même minimal.

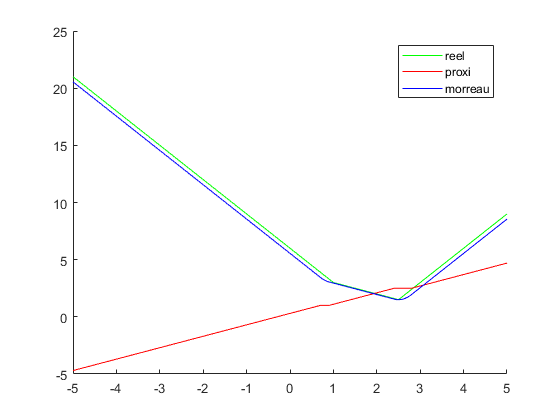
Maintenant nous allons tester le proximal et l’enveloppe de moreau sur une fonction moins classique comme abs(x-1)+abs(2\*x-5)

On trouve son proximal égale à :



On peut remarquer que la fonction possède deux points de non dérivabilité.

Ainsi on obtient l’enveloppe de moreau suivante pour gamma = 0.1.

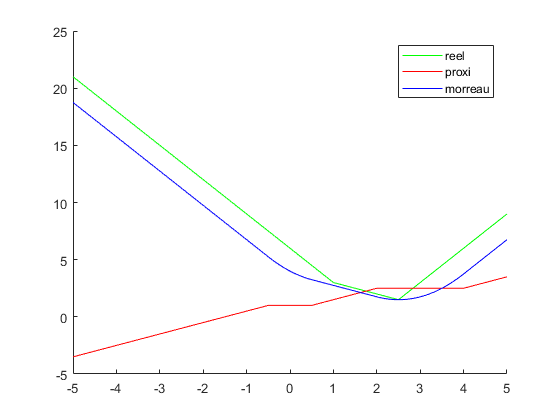


On peut voir ici la force des propriétés de l’enveloppe de moreau car bien qu’il y ait deux points de discontinuité on observe bien que l’enveloppe de moreau approche la fonction f d’origine tout en la rendant

En effet à l’aide de l’enveloppe de moreau on pourra très bien appliquer l’algorithme de descente de gradient sur celle-ci et trouver le minimum de f même si elle n’est pas différentiable. Cependant on verra par la suite qu’il y de meilleurs algorithmes qui utilisent l’opérateur proximal.

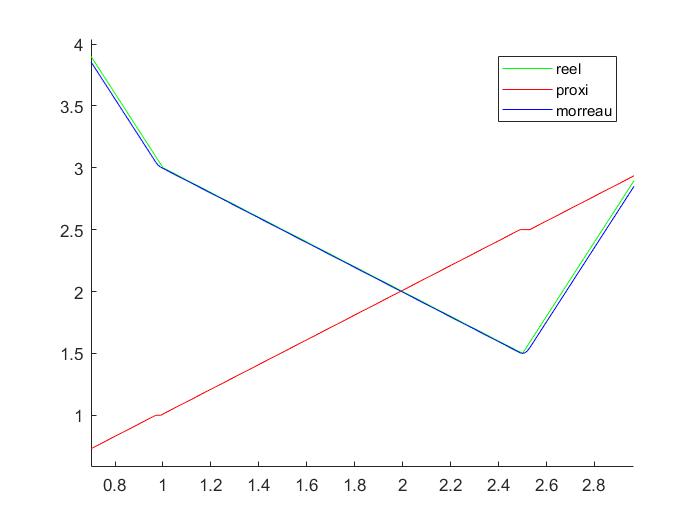
Nous allons voir ensuite l’influence du paramètre gamma sur l’opérateur proximal et l’enveloppe de moreau.

En effet pour gamma = 0.5



On remarque que les plateaux du proximal sont plus large ce qui a pour conséquence de rallonger le temps pour adoucir la pente aux points de non dérivabilité.

Pour gamma = 0.01 on a :



On remarque cette fois ci à l’opposé de précédemment que le temps de variation de pente est beaucoup plus rapide car gamma a fortement réduit. Cela a pour conséquence de rapprocher l’enveloppe de moreau à la fonction d’origine mais les variations de pentes sont plus importantes.

## Fonction conjuguée

La fonction conjuguée est définie par :

Cette fonction conjuguée est aussi appelée transformée de Legendre-Fenchel

Par exemple dans le cas de fonction particulière on a :

Pour

Pour

Pour

Cette fonction conjuguée a une utilité pour la résolution du problème dit primal :

En effet on peut le transformer en problème dit dual :

De plus la résolution du problème primal revient à la résolution du problème dual.

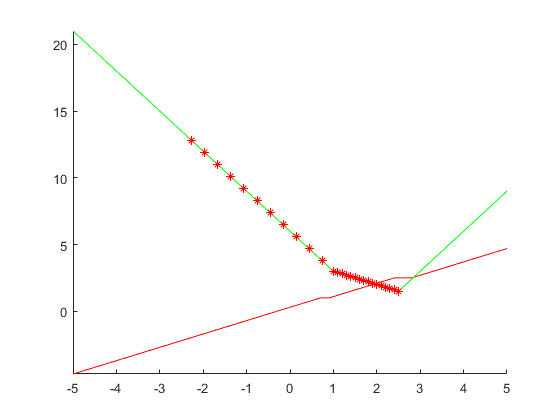
On pourra résoudre ce problème en utilisant aussi :

# II Algorithmes

## Algorithme du point proximal

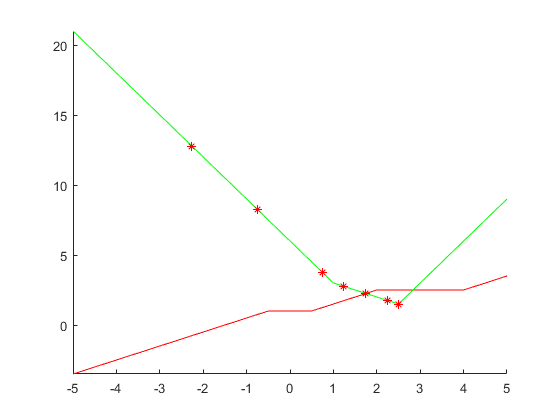
L’algorithme du point proximal a pour principe d’utiliser l’opérateur proximal tel que :

On peut remarquer qu’il n’est pas nécessaire de connaitre l’expression de l’enveloppe de moreau et seule la connaissance de l’opérateur proximal est suffisante.



On peut remarquer cet algorithme avance par pas constant jusqu’à arriver à la valeur du plateau de l’opérateur proximal ce qui a pour conséquence soit de changer le pas de descente soit de converger sur la même valeur qui se trouve être le minimal.

En effet l’algorithme converge vers un point fixe de l’opérateur proximal et ce point fixe se trouve être exactement le minimum de la fonction.



Pour gamma = 0.5 on remarque que la convergence de l’algorithme est encore plus rapide. En effet si gamma augmente on a déjà remarqué que le plateau s’agrandisse et permet que prendre un pas plus grand.

L’avantage de cet algorithme est d’être beaucoup plus rapide que l’algorithme de descente de gradient qui ne fait que tendre vers une solution alors que cet algorithme trouve directement une solution exacte.

## Algorithme de gradient proximal

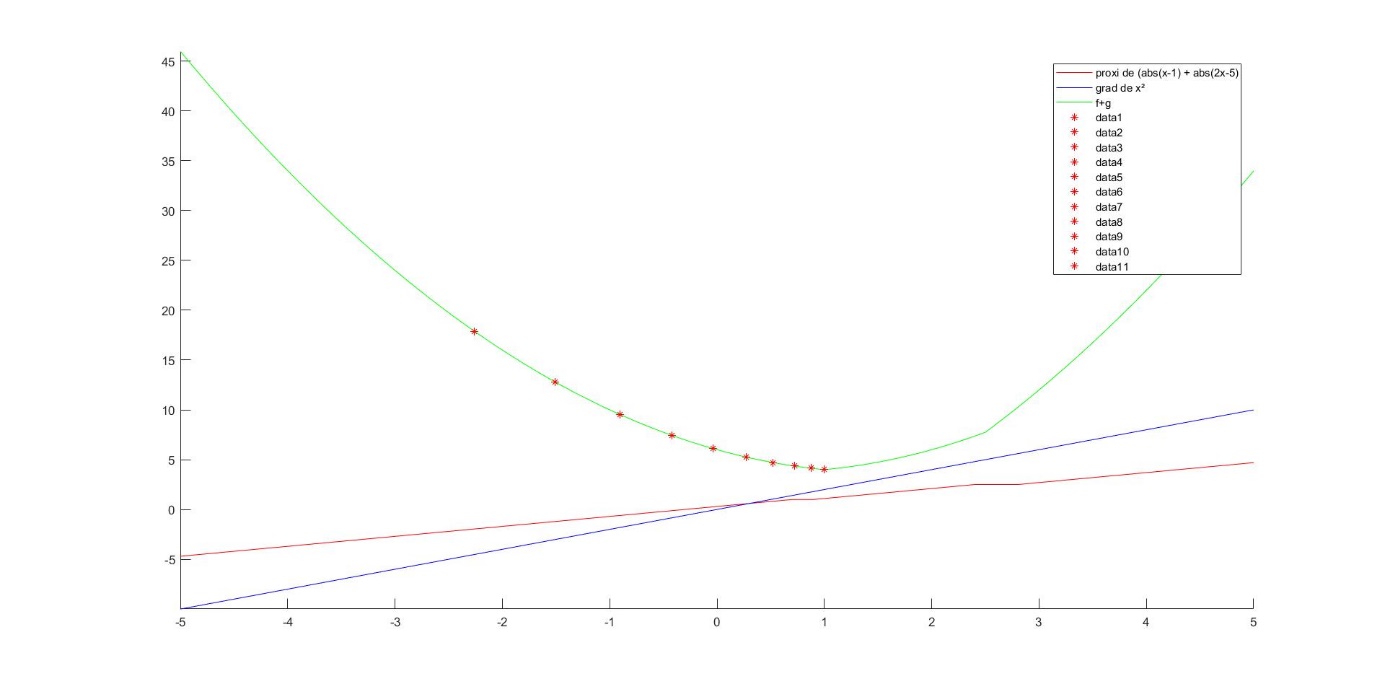
Avec cet algorithme on cherche à se ramener à la résolution du problème d’origine c’est-à-dire :

Donc de la forme avec f différentiable liée à l’attache aux données et g qui peut être non différentiable et qui est le terme de régulation.

L’algorithme de gradient proximal a pour principe d’utiliser l’opérateur proximal tel que :

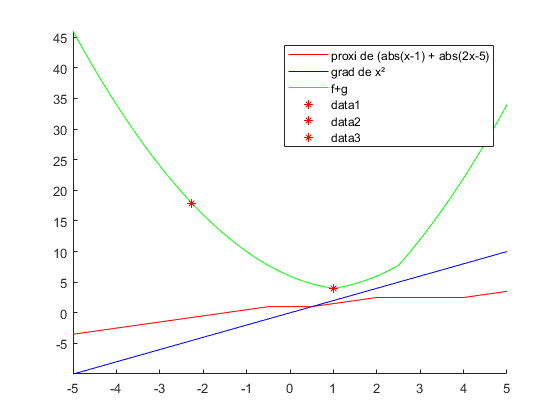
Cet algorithme est aussi connu sous le nom d’algorithme de Forward-Backward

Pour les exemples de fonctions précédente on trouve le résultat suivant.



On remarque que la convergence de cette fonction est encore très rapide par rapport à la descente de gradient.

De plus gamma =0.5 on a :



Ce qui montre bien que tout comme précédemment le paramètre gamma a pour conséquence d’augmenter la vitesse de convergence.

L’avantage de cet algorithme de pouvoir résoudre des problèmes de la forme

Cependant il est possible de résoudre ce problème seulement si

## Algorithme de Forward-Backward

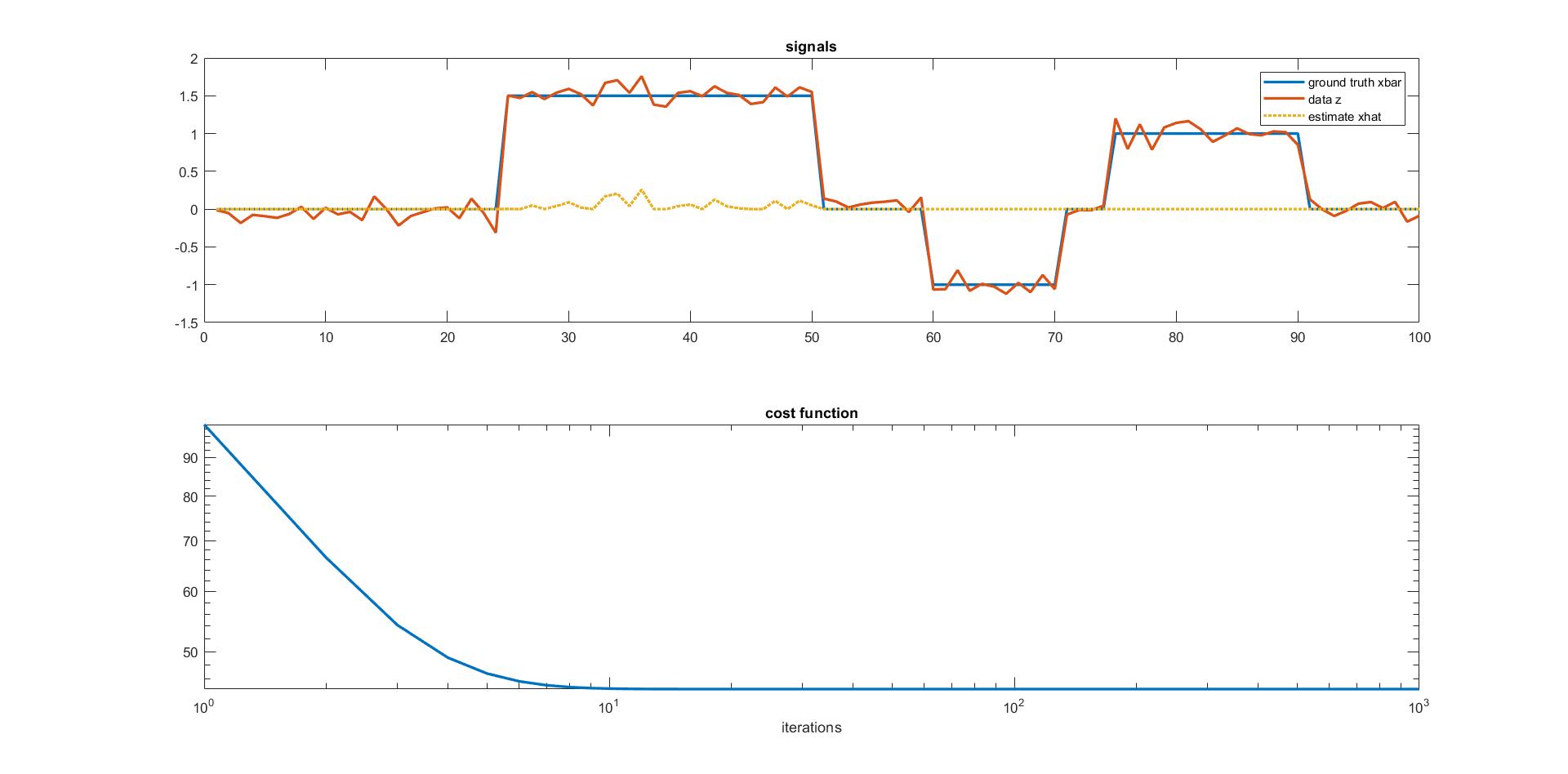
On cherche à résoudre le problème suivant dans le cas général

Pour cela on va considérer ce problème comme primal et résoudre le problème dual associé :

Pour cela on va utiliser l’algorithme de gradient proximal pour résoudre ce problème avec F différentiable et G non différentiable.

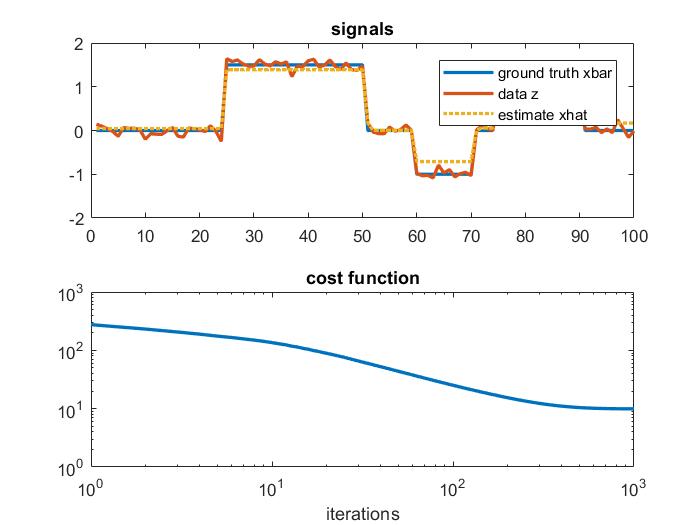
Cette méthode nous permet donc d’utiliser des opérateurs autre que l’identité comme le gradient et le Laplacien

Pour l’opérateur identité on trouve :

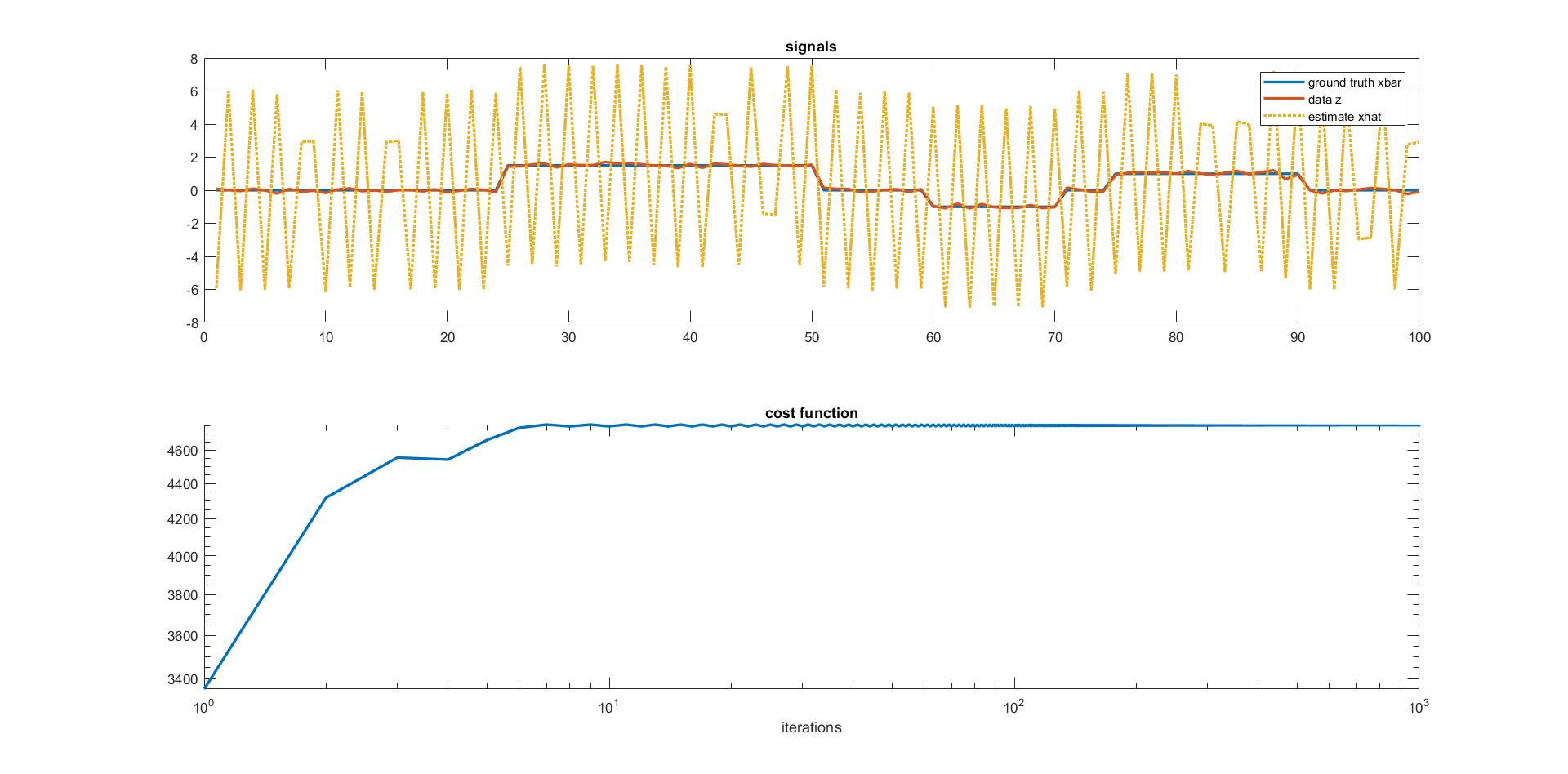


On remarque l’opérateur identité fait tendre vers 0 les valeurs de xhat.

Pour l’opérateur gradient on a :

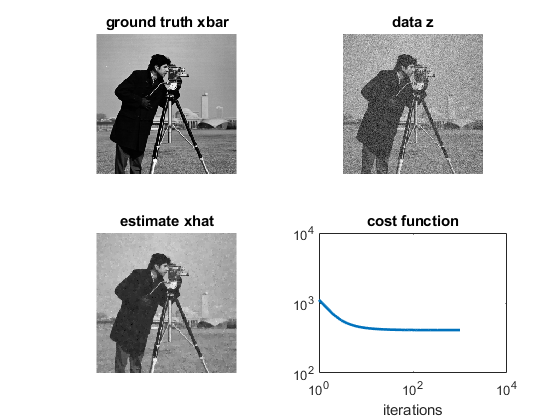


Au contraire de l’opérateur identité le gradient prend en compte le signal l’origine pour calculer xhat



Pour cet opérateur xhat utilise les données voisine pour se construire ce qui a pour conséquence de ne pas bien restituer un signal proche d’une porte.

On peut de plus utiliser cette méthode en 2D pour des images



On remarque tout comme précédemment que l’opérateur gradient permet un bon lisage de l’image tout en préservant les détails et les forts gradients de l’image.

# CONCLUSION

On peut conclure que la résolution d’un problème inverse n’est pas toujours facile. En effet les méthodes de résolutions varient grandement selon les fonctions du problème. Par exemple les fonctions peuvent être différentiable ou non.

De plus la méthode la plus aboutie qui est l’algorithme de Forward-Backward peut par exemple utiliser pour enlever le bruit d’une image de façon plutôt efficace.

On peut aussi imaginer d’autres applications à cet algorithme comme par exemple un flou où on utilisera d’autre opérateur que le gradient.

Il faut tout de même noter que les méthodes utilisant l’opérateur proximal sont très efficaces, elles nécessitent des calculs fait avant le code. Pour généraliser ces méthodes il faudrait mettre en place une méthode numérique qui peut avoir une complexité importante.