Guénon Marie et Favreau Jean-Dominique

VIM / Master SSTIM

Analyse d’image

Rapport de TD5

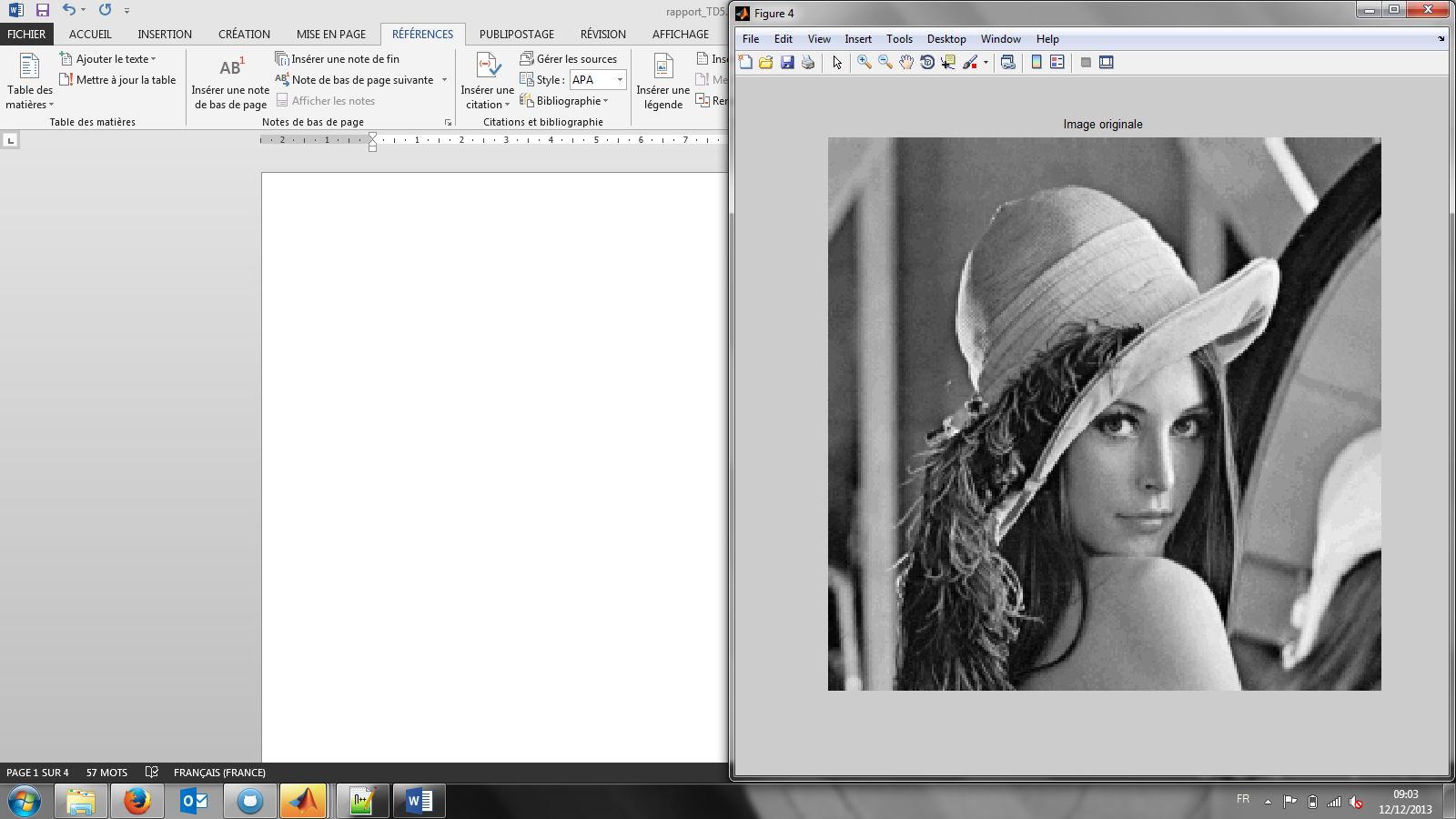
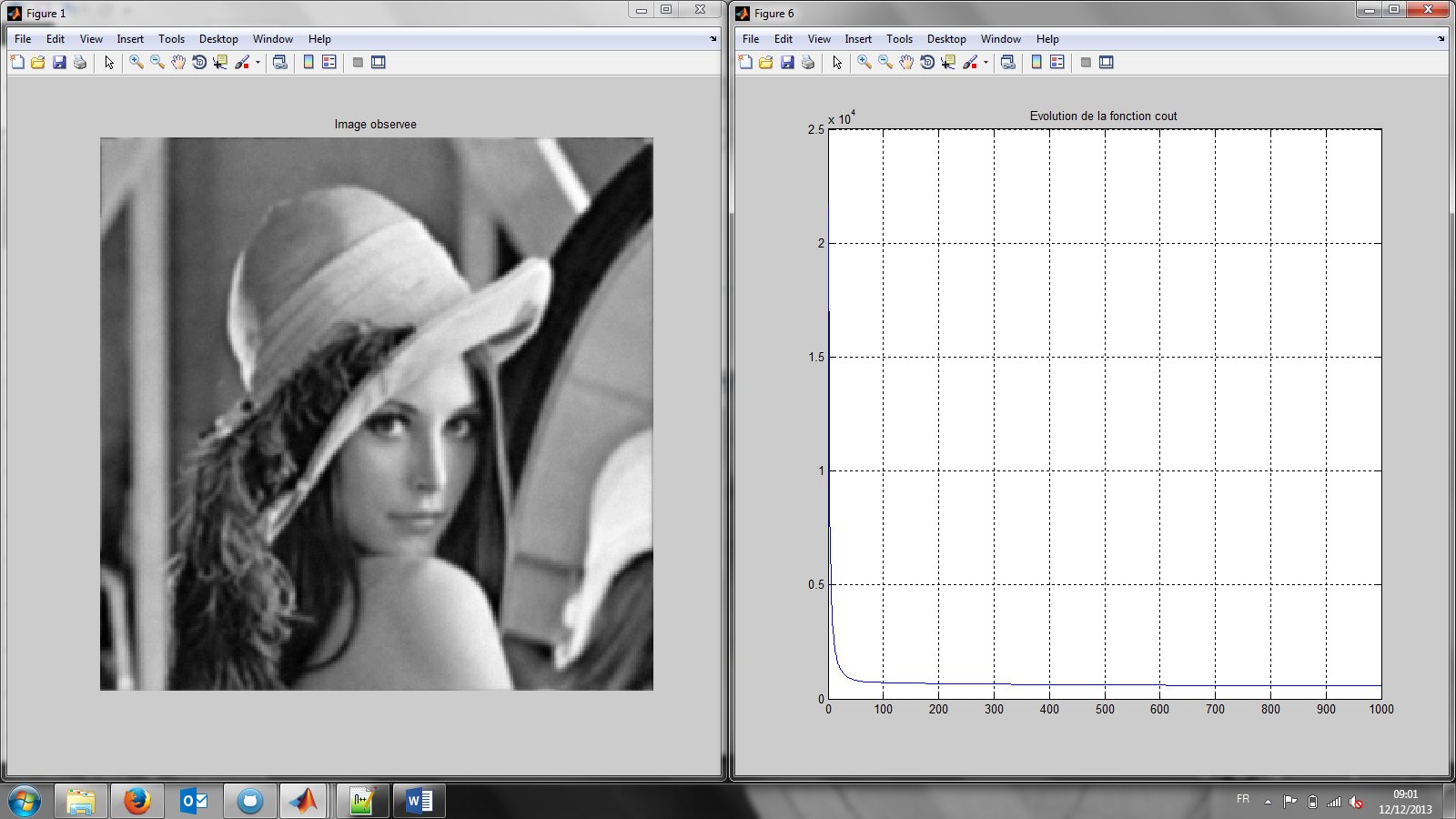
Table des matières

[Introduction 2](#_Toc374617948)

[Régularisation de Tychonov 5](#_Toc374617949)

# Introduction

Tout d’abord, nous commençons par appliquer une transformation ainsi qu’un terme aléatoire à notre image, et nous obtenons ainsi une image floutée, que l’on peut voir ci-après à gauche comparée avec l’image originale à droite :

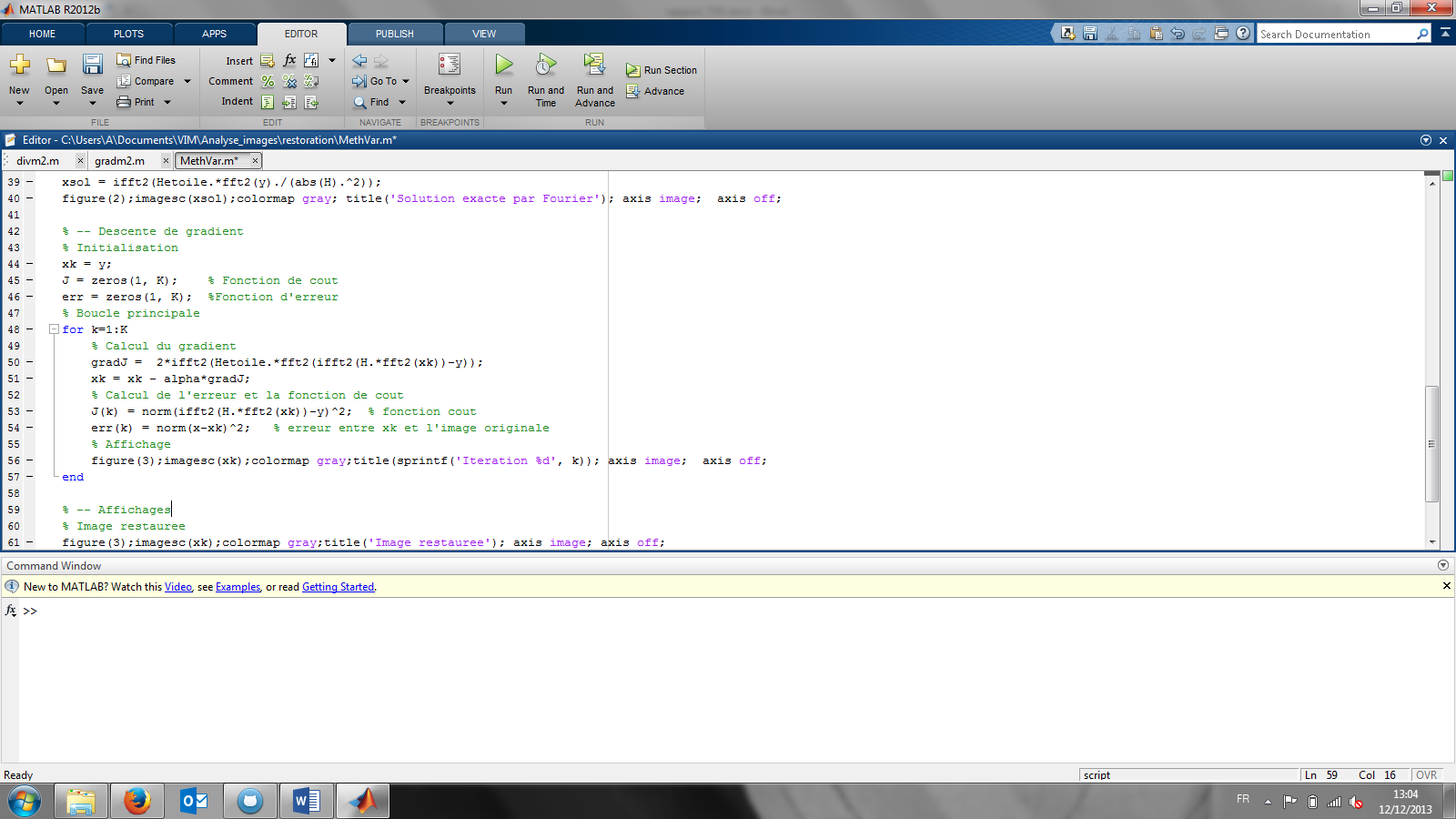
 

A partir de cette image floutée, nous cherchons maintenant à reconstruire l’image originale en utilisant un algorithme de descente de gradient. Pour cela, nous allons mettre en place l’algorithme suivant :

Démontrons déjà que  :

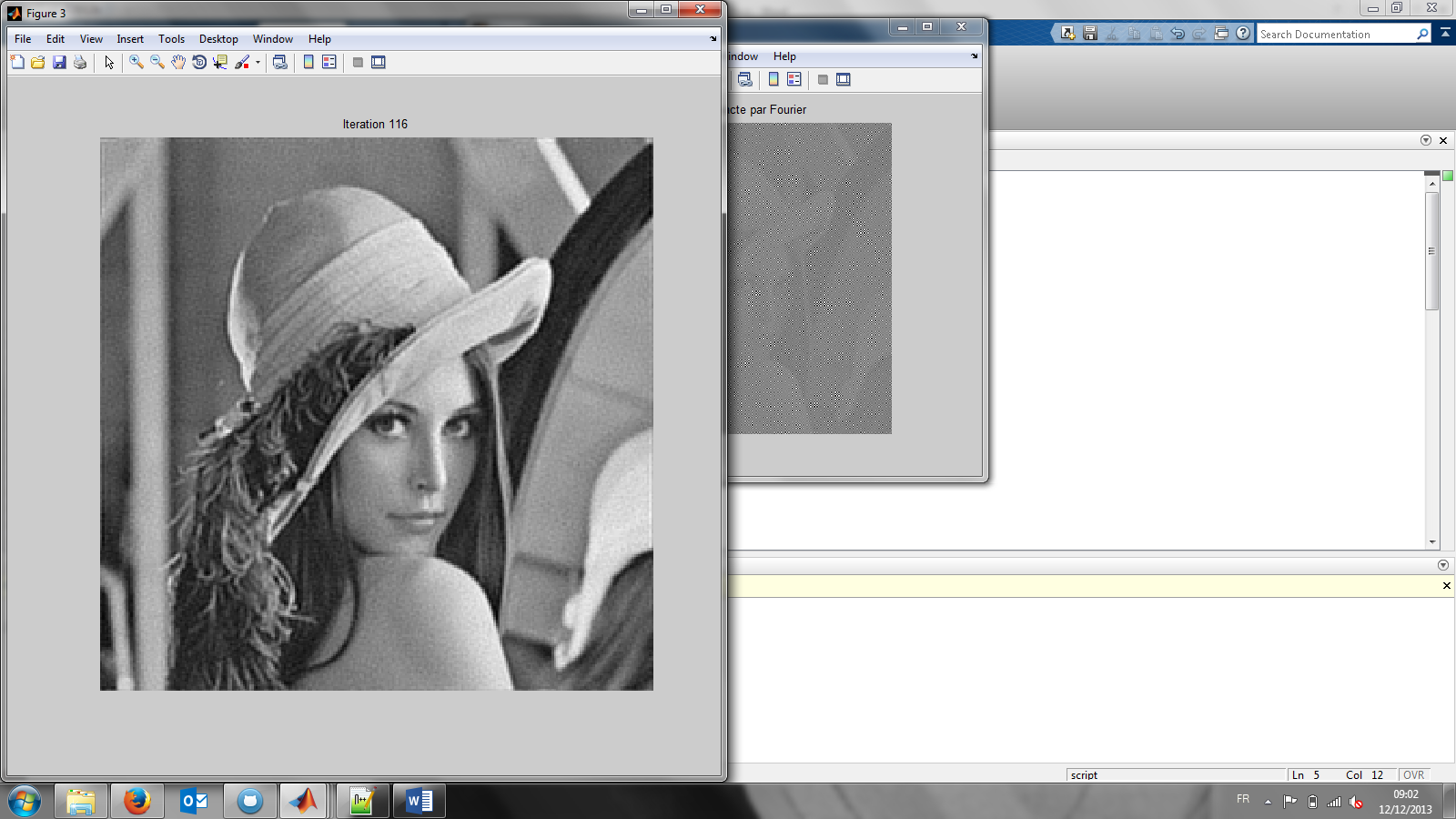
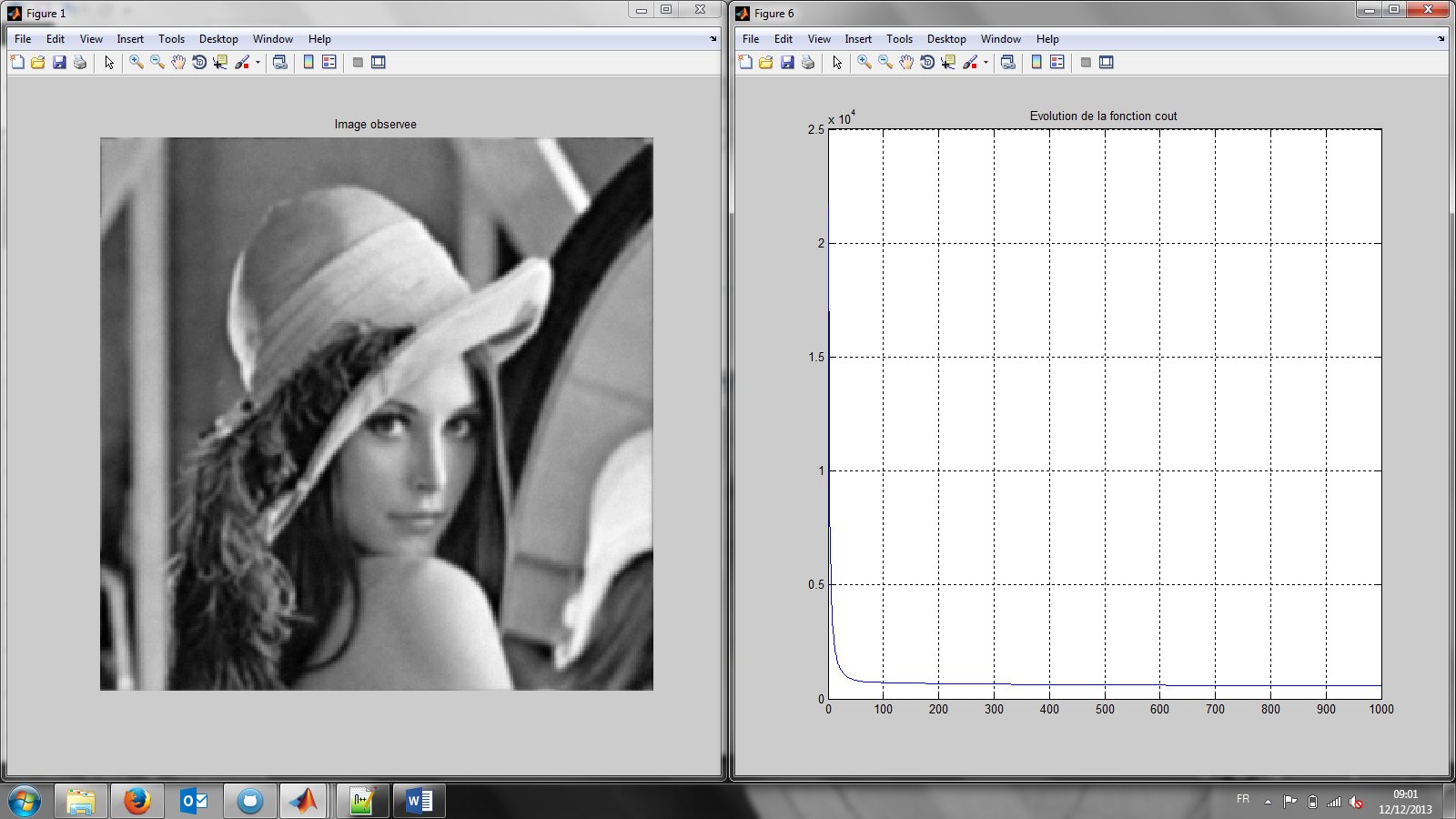
Par identification, nous avons qui est bien la formule que nous cherchions à démonter.

Une fois implémenté, cet algorithme nous donne :

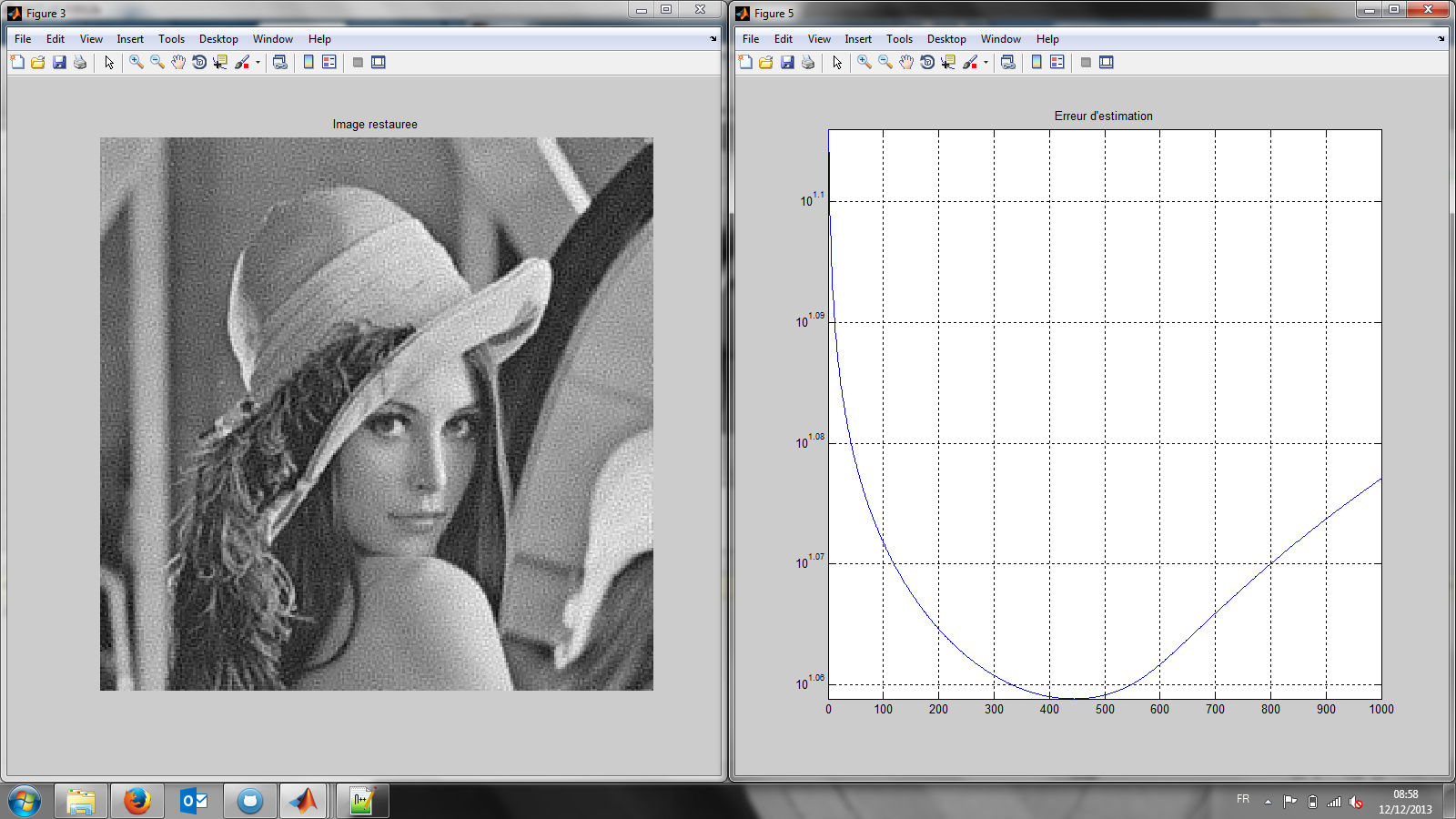
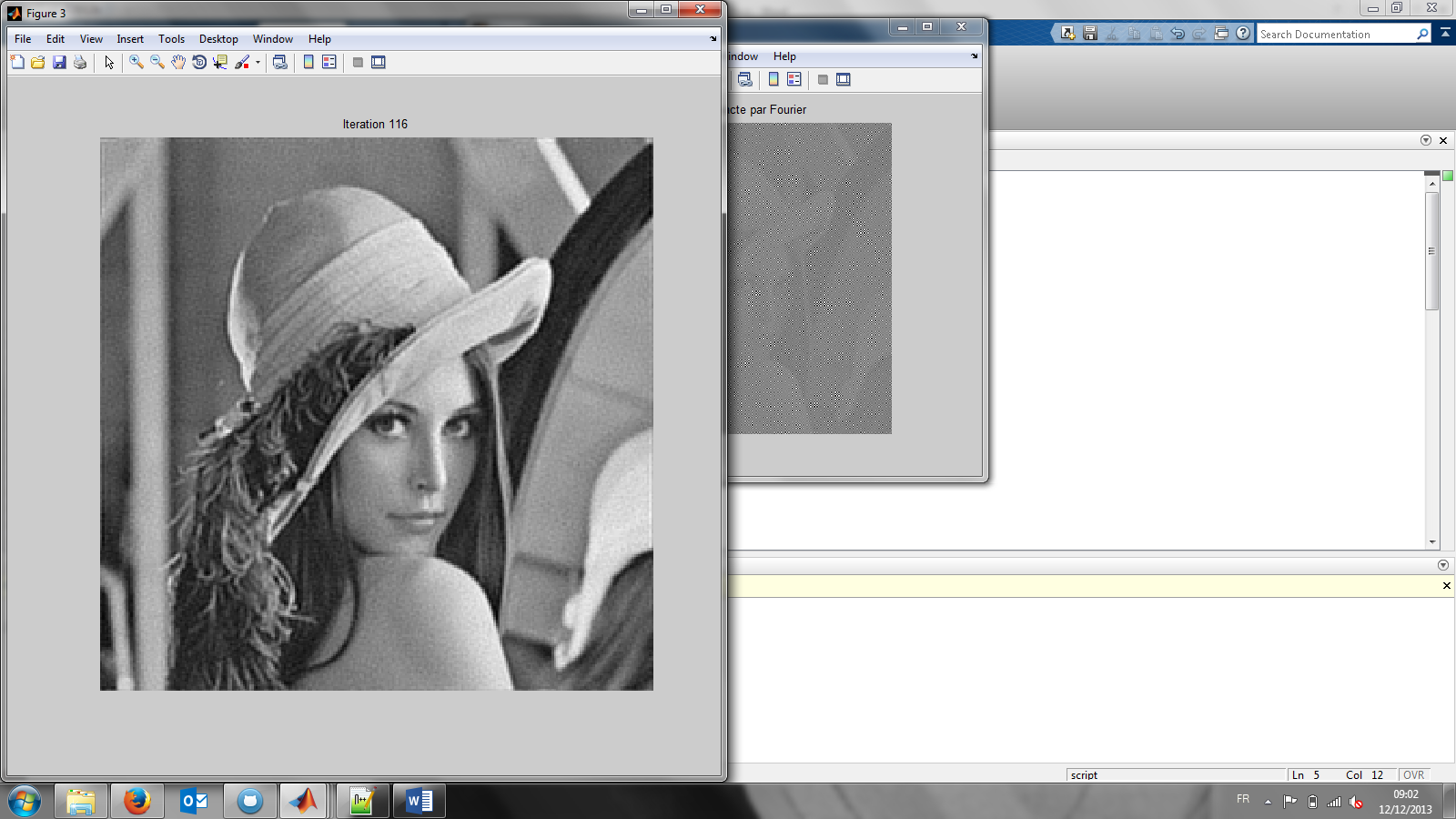


Dans notre cas, nous avons pris et .

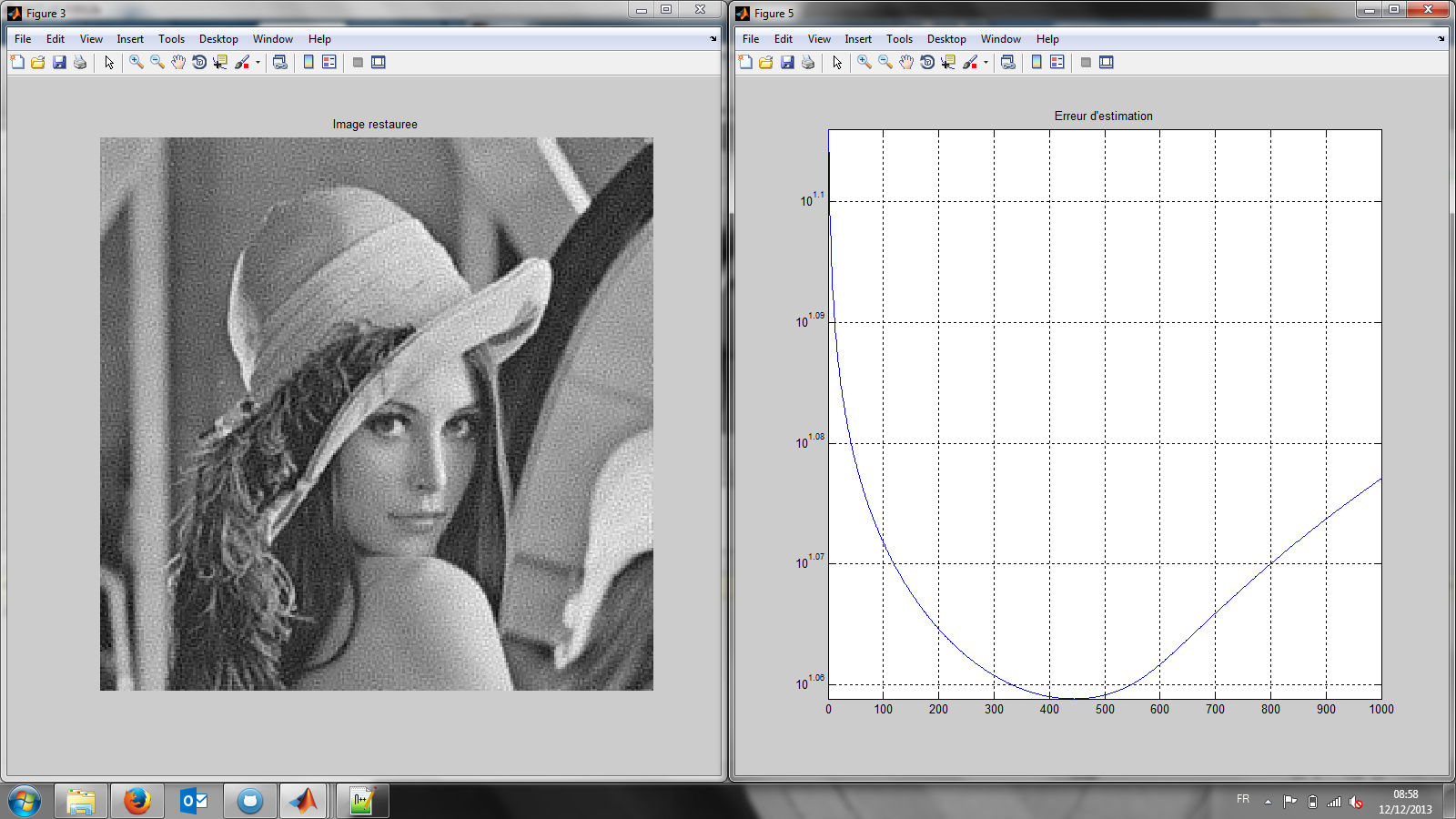
En faisant tourner notre programme de restauration d’image, nous avons constaté une amélioration de l’image transformée dans un premier temps. Comme nous pouvons ici le constater où nous comparons l’image transformée (à gauche) et l’image améliorée après 116 itération (à droite).



Cependant, nous avons aussi constaté qu’après un certain nombre d’itérations, l’image cesse de s’améliorer et nous pouvons voir une nette augmentation du bruit au sein de l’image restaurée. Comme nous pouvons le voir ci-après lors de la comparaison de l’image restaurée après 116 itérations (à gauche) et de l’image restaurée après 1000 itérations (à droite).



Ceci se voit aussi très bien sur la fonction d’estimation d’erreur qui compare la distance entre l’image restaurée au fil des itérations et l’image originale :



Comme nous pouvons le voir, l’erreur commence par diminuer puis après environ 450 itérations augmente nettement. Ceci reflète nettement l’explosion du bruit au cours de la restauration de l’image.

Ce problème de bruit est due au fait que nous n’avons pas de terme de régularisation dans notre algorithme de descente de gradient, et donc accordons de l’importance aux bruits qui finissent par l’emporter sur l’information essentielle de l’image. Pour résoudre ce problème, il faudrait donc imposer un terme de régularisation dans notre équation . Pour cela, comme nous le verrons après, nous avons décidé d’utiliser une régularisation de Tychonov d’ordre 1.

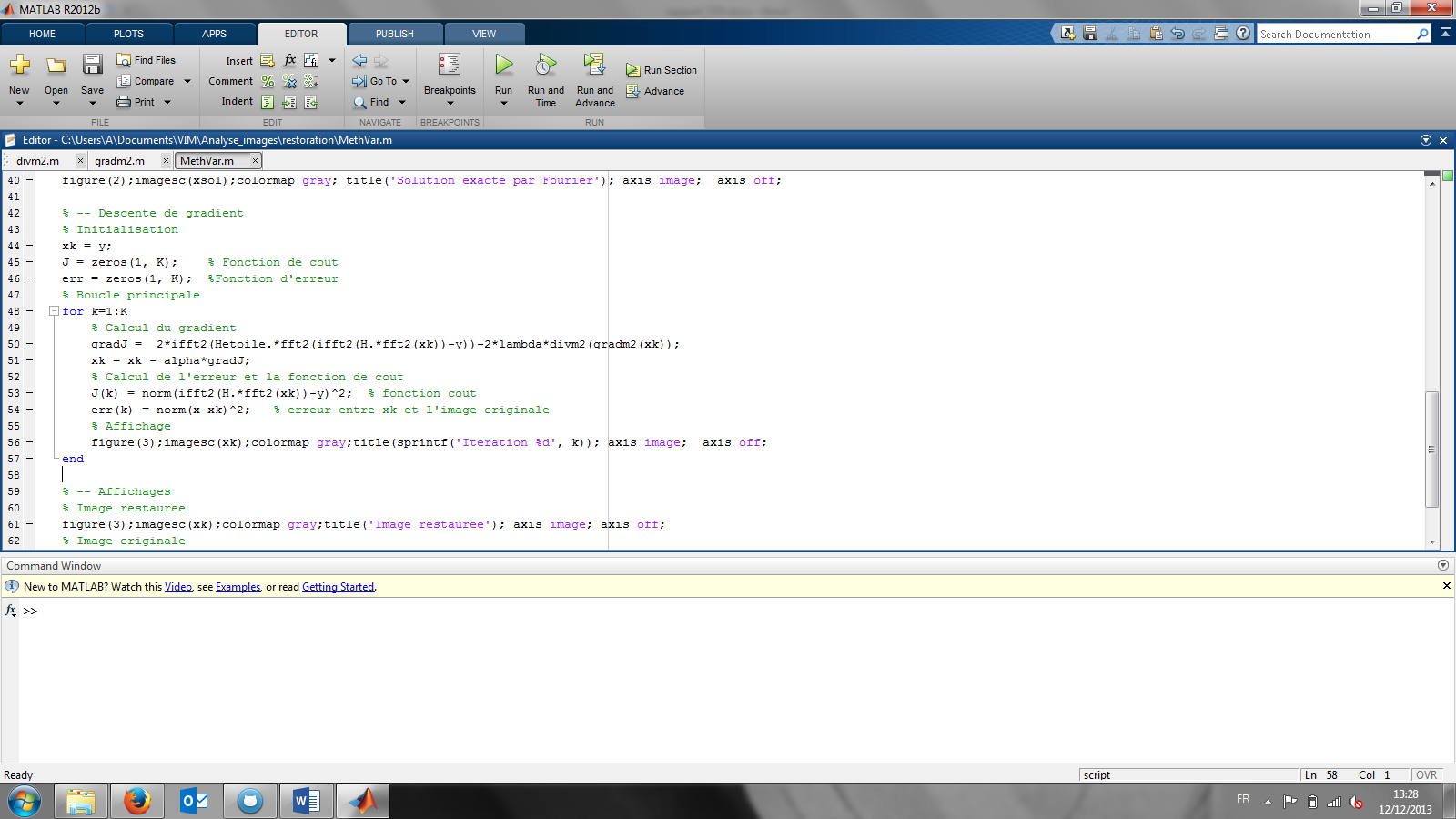
# Régularisation de Tychonov

Le terme de régularisation de Tychonov se retrouve dans , qui ne s’écrit plus mais , nous avons donc à recalculer :

D’où par identification :

Et en recomposant tous les termes, on obtient :

Ce qui nous donne le code suivant :



Nous appliquons notre algorithme avec les paramètres ; et .

En appliquant l’algorithme de descente de gradient ainsi obtenue sur une image dégradée (à gauche), nous pouvons voir apparaitre une image restaurée (à droite) à la qualité tout à fait acceptable et où l’on reconnait une girafe.



Si l’on compare les résultats obtenus sur cette image avec (à gauche) et sans (à droite) régularisation, nous pouvons constater que dans le premier cas, nous obtenons une image floutée. Ceci est due au fait que nous effectuons une sorte de lissage sur l’image pour enlever les effets de bruit.

Cependant, sans régularisation nous avons une image plus nette si l’on fait abstraction du bruit. Mais d’une part la stabilité de l’algorithme n’est pas assurée et d’autre part, si l’on augmente trop le nombre d’itération, on se rapproche de la solution exacte donnée par Fourier qui est totalement illisible.



Pour éviter le floue due à la régularisation, nous pourrions utiliser une méthode anisotropique, comme vue lors du TP précédent, qui effectuerait le lissage évoqué ci-dessus seulement aux endroits où il n’y a pas de contours et éviterait ainsi de trop flouter l’image.