ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

SCUOLA DI SCIENZE
Corso di Laurea in Matematica

MULTILEVEL METHODS IN IMAGE RESTORATION PROBLEMS

Tesi di Laurea in Ottimizzazione Numerica

Relatore:
Chiar.ma Prof.
MARGHERITA PORCELLI

Correlatore:
Chiar.ma Prof.
ELISA RICCIETTI

Presentata da:
GIOVANNI SERAGHITI

Anno Accademico 2020-2021

Contents

Abstract

Il presente lavoro è la prosecuzione dell'attività di due mesi di tirocinio svolto presso la École Normale Supérieure (ENS) di Lione, in collaborazione col team di ricerca DANTE (Dynamic Network), in particolare, con la Professoressa Elisa Riccietti, e i Professori Nelly Pustelnik e Paulo Goncavels.

Lo stage si inserisce all'interno di un progetto di ricerca più ampio del team DANTE. Lo scopo del progetto è quello di studiare nell'ambito della ricostruzione di immagini, l'uso di un insieme di tecniche di ottimizzazione numerica, note come metodi multilivello, già ampiamente utilizzate nel campo delle equazioni alle derivate parziali [4].

I metodi multilivello si basano sulla costruzione di un insieme di approssimazioni della funzione obiettivo a più livelli, gradualmente meno accurate e computazionalmente meno costose da minimizzare, e sulla definizione di specifici operatori in grado di trasferire informazioni da un livello ad un altro [3].

La prima parte del presente elaborato introduce il problema inverso per la ricostruzione di immagini e alcuni metodi numerici utilizzati per risolverlo, focalizzandosi in particolare, sugli algoritmi di ottimizzazione del gradiente.

In seguito, viene trattata la teoria dei metodi multilivello, dapprima in generale e poi, adattati al problema inverso per la ricostruzione di immagini: infatti nel corso dello stage si è cercato di sviluppare, sia dal punto di vista teorico che implementativo, un algoritmo multilivello come alternativa ai metodi già esistenti, al fine di ricostruire immagini degradate.

Una prima versione di tale algoritmo MGM (Multilevel Gradient Method) elaborata durante lo stage, viene presentata in dettaglio e testata in diverse applicazioni reali nell'ambito della ricostruzione di immagini. I risultati ottenuti durante le simulazioni sono analizzati e confrontati con i metodi di ottimizzazione non vincolata noti in letteratura.

L'ultima parte dell'elaborato illustra brevemente la teoria delle *wavelets*, strumento ampiamente utilizzato per processare immagini, in quanto consente di ottenerne una rappresentazione "sparsificata", mantenendone intatte le principali caratteristiche. Considerando le wavelets da una nuova prospettiva, è stato possibile definire degli operatori

integrabili nello schema del multilvello al fine di trasferire informazioni da un livello di approssimazione minore ad uno maggiore e viceversa. Grazie all'utilizzo delle wavelets nel nuovo contesto, si è potuta sviluppare una seconda versione dell'algoritmo MGM, che viene presenta e paragonata alla prima.

Dallo stage e dal presente lavoro, lo studio sui metodi multilivello nella ricostruzione di immagini, proseguirà nella prospettiva di trattare problemi di dimensioni maggiori e in contesti più generali, in collaborazione con l'ENS di Lione.

Introduction

In many fields of interest, for example in satellite imaging, original data are usually degraded by physical processes during their acquisition or transmission and it could also happen that they are affected by some noise which brings to uncertainty. The need of having access to original data is what justifies the formulation of inverse problems in image restoration, which consist in recovering from a degraded image, the most similar approximation of the original one, that is usually unknown. [7].

In next chapters, this class of problems will be formalized from a mathematical point of view and some techniques for their solution will be presented. In particular some unconstrained minimization techniques such as Constant Step Size Gradient method (CGM) and Backtracking Gradient Method (BGM) will be discussed and then applied to the reconstruction of degraded images [5].

Furthermore, part of the theory of the so-called multilevel approach will be reviewed with the aim of providing an algorithm (MGM), which represents an alternative to the standard gradient minimization methods. The multilevel scheme is based on the knowledge of a set of estimations for the initial function on different levels of approximation, less costly to minimize than the original one and on some operators allowing to transfer information from one level to another [4].

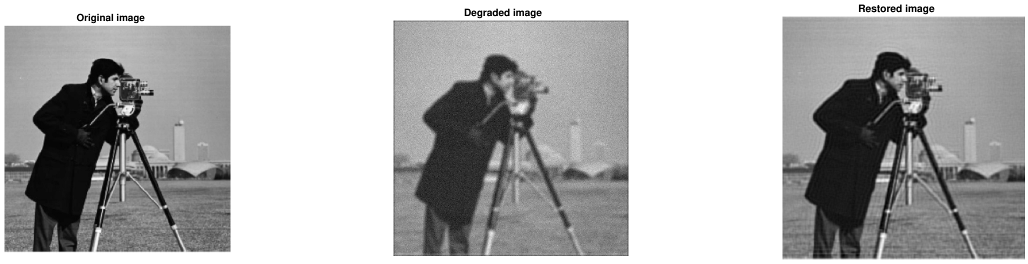
This class of methods is widely used in the field of partial differential equations (PDEs) and in this thesis, the theory will be adapted and applied to the restoration of degraded images. We will also provide some versions of multilevel algorithms that can be used to solve particular instances of the image restoration problem.

In addition a quick overview of wavelets theory will be presented, explaining how using wavelets, it is possible to decompose an image into a parsimonious representation, containing all the main information from the original image but in few sparse coefficients [2]. This feature of wavelets representation will be exploited in order to define new operators, able to transfer information between two different levels of the multilevel hierarchy scheme.

Chapter 1

Inverse problem in image restoration

The inverse problem in image restoration consists of recovering from degraded observed data the most similar image possible to the original one (see Fig. 1.1). Usually the original image is unknown and the observed data are degraded by physical processes during acquisition or storage and they could also be affected by random noise.



(a) Original image (b) Degraded observation (c) Restored image

Figure 1.1: example of image restoration problem.

1.1 Problem construction

From a mathematical point of view it is possible to describe such a problem using the following equation [7]:

$$z = D_\alpha(A\bar{x}), \tag{1.1}$$

where $z \in \mathbb{R}^m$ is the observed degraded image in vector form, $\bar{x} \in \mathbb{R}^n$ is the original image (usually unknown), $A \in \mathbb{R}^{m \times n}$ is the linear degradation operator (e.g. blur), $D_\alpha : \mathbb{R}^m \to \mathbb{R}^m$ is the noise, parametrized by the scalar $\alpha \geq 0$.

In the following, it will be considered a particular instance of this general problem, in which D_α is assumed to be an additive noise (usually gaussian noise). The new formulation, therefore is:

$$z = A\bar{x} + b, \tag{1.2}$$

where b is the realization of the noise, for example gaussian noise with variance α.
We want to find the best possible approximation $\hat{x}(z) \in \mathbb{R}^n$ of the original image $\bar{x}$ starting from z, the observed image.

Let us first assume that the image formation process is noise free. The problem $z = A\bar{x}$ is said to be well-posed if it fulfills the Hadamard conditions namely:

1. existence of a solution, i.e. $range(A) = \mathbb{R}^m$;

2. uniqueness of solution, i.e. $ker(A) = \{0\}$;

3. stability of the solution $\hat{x}$ relatively to the observation z i.e.

$$\forall (z, z') \in (\mathbb{R}^m)^2, \qquad \|z - z'\| \to 0 \Rightarrow \|\hat{x}(z) - \hat{x}(z')\| \to 0.$$

If the first and second conditions are satisfied it means that the solution exists and it is unique, while the stability condition ensures that a small perturbation on the observed image leads to a slight variation of the recovered image.

Note that, assuming A to be a square matrix, in image restoration problem, an ill conditioning of the matrix A can lead to deal with an ill-posed problem. Unfortunately for most of the common choices of A in applications, usually blur operators, the resulting image restoration problem is ill-posed.

1.2 Problem formulations

In this section we are going to introduce some possible formulations for the image restoration problem (1.2), analyzing first the non regularized models, and then focusing on different types of regularization terms that can be considered to stabilize the model.

- **Naive model**
  Assuming $n = m$ and A being a full rank matrix, a solution always exists and it is unique as A is injective. One simple way to find a possible solution to problem (1.2) is to apply the inverse matrix A^{-1} to the degraded observation:

$$\hat{x} = A^{-1}(A\hat{x} + b) = \hat{x} + A^{-1}b. \tag{1.3}$$

  Since the inverse of the matrix A has to be computed, this way of proceeding has an high computational cost. Furthermore, if the matrix A is ill-conditioned the term $A^{-1}b$ may become very large, amplifying the noise and leading to an irregular restored image.

- **Least-squares model**
  If A is not invertible, it is reasonable to think that the degraded version $A\hat{x}$ of the solution $\hat{x}$ can be close to the observed vector z. In other words, the Euclidean distance between $A\hat{x}$ and z should be minimized:

  $$\hat{x} \in \arg\min_{x \in \mathbb{R}^n} \|Ax - z\|_2^2.$$

  - If A has rank $r = m < n$, the problem is under-determined and the second Hadamard condition is not satisfied. Then, A is a "wide" matrix and the uniqueness of the solution is not guaranteed.
    The generalized inverse of A is thus equal to $A^T(AA^T)^{-1}$ so the solution can be computed by
    $$\hat{x} = A^T(AA^T)^{-1}z.$$
  - If A has rank $r = n < m$, it is a "tall" matrix and the first Hadamard condition is not fulfilled. There is no exact solution if $z \notin \mathit{Range}(A)$. The least-squares problem however has a unique solution given by

    $$\hat{x} = (A^TA)^{-1}A^Tz.$$

- **Regularized least-squares**
  In order to stabilize the solution and to guarantee its uniqueness, a regularization term can be added so that the new problem becomes:

  $$\hat{x} \in \arg\min_{x \in \mathbb{R}^n} \|Ax - z\|_2^2 + \lambda\Phi(x), \tag{1.4}$$

  where Φ: $\mathbb{R}^n \rightarrow \mathbb{R}$ is the *regularization function*, λ >0 is the *regularization parameter.*

  The first term in the objective function is called the *data fidelity term* and the λ can be interpreted as a trade off between this term and the regularization one. The more λ is large and the more the regularization term plays a relevant role on the minimization of the objective function.

  There are different possible choices for the regularization term. If we consider D a linear operator, which often corresponds to an high pass filter operator, such as a gradient or Laplacian operator, Φ can be chosen as follows:

1. l_2 **regularization term (Tikhonov regularization)**:
   Consider $\Phi(x) = \|Dx\|_2^2$, the original problem becomes:

   $$\hat{x} \in \arg\min_{x \in \mathbb{R}^n} \|Ax - z\|_2^2 + \lambda \|Dx\|_2^2. \tag{1.5}$$

   In this particular case it is possible to compute the exact solution obtained by the annihilation of the gradient of the objective function and solving the resulting linear system, that is:

   $$\hat{x} = (A^T A + \lambda D^T D)^{-1} A^T z. \tag{1.6}$$

   If the dimension of the problem is large, the computation of the exact solution in (1.6) can be very expensive, so in many cases, iterative strategies converging to the minimum of the function are used, even though an exact solution exists.
   Here again, if the matrix $(A^T A + \lambda D^T D)$ is ill conditioned, the problem is ill-posed.

2. l_1 **regularization term**
   Let us define $\Phi(x) = \|Dx\|_1$, where $\|x\|_1 = \sum_{i=1}^n |x_i|$, so problem (1.4) becomes:

   $$\hat{x} \in \arg\min_{x \in \mathbb{R}^n} \|Ax - z\|_2^2 + \lambda \|Dx\|_1.$$

   This type of regularization term is often used when the product Dx gives a sparse resulting vector. In some applications in image reconstruction, D is considered to be the operator which performs the wavelet transform of the image x, in this case Dx corresponds to the wavelet coefficients of the image x, which are in general really sparse so the l_1-regularization has to be preferred. [7].

   Another possible choice for D could be an operator which performs the difference between adjacent pixels on the image, in this case the resulting vector Dx is still sparse and we refer to this kind of regularization as TV (Total Variation) [8].

3. **Huber function regularization term**:
   We can also consider $\Phi(x) = h_\eta(Dx)$, where $\eta > 0$ is called the Huber parameter and the function h_η is defined as:

   $$h_\eta(y) = h_\eta(y_1, \dots, y_n) = \sum_{i=1}^n \tilde{h}_\eta(y_i), \qquad \forall y \in \mathbb{R}^n,$$

where $\tilde{h}$ is the 1-D Huber function [1] (see Fig 1.2), defined as:

$$\tilde{h}_\eta(x) = \begin{cases} \frac{1}{2}x^2 & \text{se } |x| < \eta, \\ \eta|x| - \frac{1}{2}\eta^2 & \text{se } |x| > \eta, \end{cases} \qquad \forall x \in \mathbb{R}.$$

The Huber function $\tilde{h}_\eta$ is a smooth approximation of the l_1 norm obtained reshaping the function in a neighborhood of the origin, which is a non-differentiability point, and considering instead the l_2 norm in that interval. The more the Huber parameter η is small and the more the function is close to the l_1 norm.

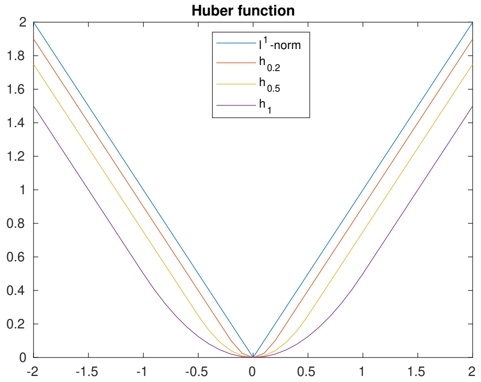


Figure 1.2: 1-D Huber function with different parameters.

In this thesis, we focus on the problem formulation which uses the Huber function regularization term, in order to consider a smooth imaging model that allows to apply the class of multilevel optimization algorithms, typically employed for partial differential equation (PDEs). For this reason, from now on the image restoration problem to which we refers to will be:

$$\min_{x \in \mathbb{R}^n} f(x) := \|Ax - z\|_2^2 + \lambda h_\eta(Dx). \tag{1.7}$$

In the next chapter, some general minimization algorithms will be reviewed, and they will be used to solve the problem (1.7).

Chapter 2

Iterative methods for unconstrained minimization problem

We address the problem of minimizing a smooth function of real variables, that is solving the following unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x), \tag{2.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously-differentiable function called the *objective function*. The algorithms for the numerical solution of (2.1) are iterative, that is, starting from an initial guess $x_0 \in \mathbb{R}^n$, a sequence $\{x_k\}_{k \in \mathbb{N}}$ of approximations of a solution of (2.1) is generated. Typically these methods are constructed to be convergent to a stationary point $x^* \in \mathbb{R}^n$ of problem (2.1), that is such that:

$$\lim_{k \to \infty} \|\nabla f(x_k)\| = 0. \tag{2.2}$$

In addition the convergence to a (local) minimum is ensured by imposing the simple decrease of the objective function, $f(x_{k+1}) \leq f(x_k)$, $\forall k \in \mathbb{N}$.

In particular, an algorithm is said to be *globally convergent* if the condition (2.2) is guaranteed for any initial guess x_0.

Moreover a vector $d \in \mathbb{R}^n$ is said to be a *descent direction* for f in x if $\nabla f(x)^T d < 0$.

One of the standard way to design a convergent algorithm is to pick a descent direction $d_k \in \mathbb{R}^n$ for f in x_k and consider the iteration:

$$x_{k+1} = x_k + \alpha_k d_k, \tag{2.3}$$

where α_k is the step length at each iteration k [5].

A standard choice for the descent direction is $d_k = -\nabla f(x_k)$, which is called the *steepest* *descent direction* and it is the one in which the value of f decreases the fastest. In addition, the steepest descent direction is also the one that minimizes the directional derivative of f in x_k, in fact:

$$\frac{\partial f}{\partial d_k}(x_k) = \nabla f(x_k)^T d_k = \|\nabla f(x_k)\| \|d_k\| \cos(\theta),$$

where θ denotes the angle between $\nabla f(x_k)$ and d_k. The minimum value is reached by $cos(\theta) = \pi$, which gives the direction of maximum decrease:

$$d_k = -\frac{\nabla f(x_k)}{\|\nabla f(x_k)\|}.$$

For what it concerns the step length computation, assuming d_k to be a descent direction, ideally we would like to choose the step length that solves the problem:

$$\min_{\alpha} \phi(\alpha) = f(x_k + \alpha d_k). \tag{2.4}$$

In fact, solving the problem (2.4) exactly will provide the step that best minimizes the objective function. Unfortunately, the solution of (2.4) would require an high computational cost, therefore in practice, the function ϕ is not exactly minimized but a step length that achieves an adequate reductions of ϕ at low cost is identified.
Different techniques can be used for this purpose and some are reviewed in the next sections.

2.1 Armijo and Wolfe conditions

In the following section, we will investigate under which assumptions on the step α_k, it is possible to provide a proof of the global convergence of the sequence $\{x_k\}$ to a stationary point of the objective function f.
Armijo rule:
Given x_k and d_k descent direction for f in x_k i.e. $\nabla f(x_k)^T d_k < 0$ and $c_1 \in (0, 1)$, choose α_k such that:

$$f(x_k + \alpha_k d_k) < f(x_k) + \alpha_k c_1 \nabla f(x_k)^T d_k. \tag{2.5}$$

Since $\nabla f(x_k)^T d_k < 0$ this condition is stronger than just imposing the simple decrease of the objective function.

Define

$$\phi(\alpha) = f(x_k + \alpha d_k),$$
$$l(\alpha) = f(x_k) + \alpha c_1 \nabla f(x_k)^T d.$$

The Armijo condition states that the step α is acceptable if $\phi(\alpha) \leq l(\alpha)$.
Note that for small positive values of α, the linear function l lies above the graph of ϕ.

This is true because the slope of $l(\alpha)$ is $c_1\nabla f(x_k)^T d_k$ which is equal to $c_1\phi'(0)$ and, since $c_1 < 1$ and both terms are negative, it holds $c_1\nabla f(x_k)^T d_k > \nabla f(x_k)^T d_k$.

Choosing α_k according to (2.5) avoids selecting too large steps, but this condition is not still sufficient to ensure the convergence of the method, since it may happen that too small steps are taken not allowing the algorithm to make reasonable progress on decreasing the function.

Wolfe rule:
Given x_k and d_k descent direction for f in x_k and $c_2 \in (c_1, 1)$, choose α_k such that:

$$\nabla f(x_k + \alpha_k d_k)^T d_k \geq c_2 \nabla f(x_k)^T d_k. \tag{2.6}$$

This condition can be interpreted as a comparison between the slope of
$\phi'(\alpha_k) = \nabla f(x_k + \alpha_k d_k)^T d_k$ and a desired slope $c_2\nabla f(x_k)^T d_k$. We want the slope $\phi(\alpha_k)$ to be significantly negative so that we can expect to reduce the function moving further on that direction.
Choosing α_k according to Wolfe condition avoids to select to small steps.

Theorem 2.1.1. *[5, lemma 3.1] Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable and bounded below in $\{x_k + \alpha d_k | \alpha > 0\}$, with d_k descent direction for f in x_k, and let*
c_1, c_2: $0 < c_1 < c_2 < 1$.
It exists $I \subseteq (0, +\infty)$ non empty such that every $\alpha \in I$ satisfies Armijo and Wolfe conditions.

Proof. Let
$\phi(\alpha) = f(x_k + \alpha d_k)$, $\qquad l(\alpha) = f(x_k) + \alpha c_1 \nabla f(x_k)^T d.$

Define $g(\alpha) = \phi(\alpha) - l(\alpha)$, Armijo condition requires that $g(\alpha) < 0$.

We have:

$$g(0) = \phi(0) - l(0) = f(x_k) - f(x_k) = 0,$$

and

$$g'(0) = \phi'(0) - l'(0) = \nabla f(x_k)^T d_k - c_1\nabla f(x_k)^T d_k = (1 - c_1)\nabla f(x_k)^T d_k < 0.$$

Since $g(0) = 0$, g decreases and $g \in C^0$, it exists a right neighbourhood of zero where $g(\alpha) < 0$. Let $\bar{\alpha}$ be the smallest positive zero of $g(\alpha)$.
It holds $g(\alpha) < 0$, $\forall \alpha \in [0, \bar{\alpha}]$, which is equivalent to state that all $\alpha \in [0, \bar{\alpha}]$ satisfy Armijo rule.
In particular in $\bar{\alpha}$ it holds $g(\bar{\alpha}) = 0$, so it follows

$$f(x_k + \bar{\alpha} d_k) - f(x_k) = c_1 \bar{\alpha} \nabla f(x_k)^T d_k.$$

For the mean value theorem applied to $\phi(\alpha)$ in $[0, \bar{\alpha}]$, it exists $\tilde{\alpha} \in (0, \bar{\alpha})$ such that

$$\phi(\bar{\alpha}) - \phi(0) = \bar{\alpha}\phi'(\tilde{\alpha}),$$

that is:

$$\bar{\alpha}\nabla f(x_k + \tilde{\alpha}d_k)^T d_k = f(x_k + \bar{\alpha}d_k) - f(x_k) = c_1\bar{\alpha}\nabla f(x_k)^T d_k > c_2\bar{\alpha}\nabla f(x_k)^T d_k.$$

Deleting $\bar{\alpha}$ we obtain:

$$\nabla f(x_k + \tilde{\alpha}d_k)^T d_k > c_2\nabla f(x_k)^T d_k,$$

The strictly Wolfe condition is satisfied in $\tilde{\alpha}$, so it exists a neighbourhood I_W of $\tilde{\alpha}$ where Wolfe condition is satisfied. This means that in $I_W \cap [0, \bar{\alpha}]$ both Armijo and Wolfe criterion are fulfilled.

□

We now provide the main convergence result.

Theorem 2.1.2. *The Zoutendijk's Theorem [5, Theorem 3.2]*
Let $\Omega = \{x \in \mathbb{R}^n | f(x) \leq f(x_0)\}$, $f \in C^1(\Omega)$ and lower bounded on Ω, d_k descent direction for f, and assume that α_k satisfies Armijo and Wolfe condition and that $\nabla f(x)$ is Lipschitz continuous in Ω.
Let θ_k be the angle between $-\nabla f(x_k)$ and d_k, i.e. the angle such that

$$\cos(\theta_k) = -\frac{\nabla f(x_k)^T d_k}{\|\nabla f(x_k)\| \|d_k\|}.$$

The numerical series

$$\sum_{j=0}^{+\infty} \cos^2(\theta_j) \|\nabla f(x_j)\|^2$$

is convergent.

Proof. Start considering the Wolfe condition:

$$\nabla f(x_k + \alpha_k d_k)^T d_k \geq c_2\nabla f(x_k)^T d_k.$$

Add $-\nabla f(x_k)^T d_k$ to both members:

$$\nabla f(x_k + \alpha_k d_k)^T d_k - \nabla f(x_k)^T d_k \geq c_2\nabla f(x_k)^T d_k - \nabla f(x_k)^T d_k,$$

that is,

$$\begin{aligned}(c_2-1)\nabla f(x_k)^T d_k &\leq (\nabla f(x_k+\alpha_k d_k)-\nabla f(x_k))^T d_k\\ &\leq \|\nabla f(x_k+\alpha_k p_k)-\nabla f(x_k)\|\|d_k\|\\ &\leq L\|(x_k+\alpha_k d_k)-x_k\|\|d_k\| = L\alpha_k\|d_k\|^2,\end{aligned}$$

That gives

$$\alpha_k \geq \frac{(c_2-1)\nabla f(x_k)^T d_k}{L\|d_k\|^2},$$

which is a positive quantity because $c_2-1<0$ and $\nabla f(x_k)^T d_k < 0$.

Note that

$$\begin{aligned}f(x_{k+1}) &\leq f(x_k)+\alpha_k c_1 \nabla f(x_k)^T d_k \leq f(x_k)+\frac{(c_2-1)c_1}{L}\frac{(\nabla f(x_k)^T d_k)^2}{\|d_k\|^2}\\ &= f(x_k)-q\frac{(\nabla f(x_k)^T d_k)^2}{\|\nabla f(x_k)\|^2\|d_k\|^2}\|\nabla f(x_k)\|^2 = f(x_k)-q\cos^2(\theta_k)\|\nabla f(x_k)\|^2,\end{aligned}$$

where $q=-\frac{(c_2-1)c_1}{L}>0$.

This inequality is verified for each α_j that satisfies the assumptions, so it holds:

$$f(x_{j+1}) \leq f(x_j)-q\cos^2(\theta_j)\|\nabla f(x_j)\|^2, \qquad \forall j\leq k.$$

Using it recursively we have:

$$f(x_{k+1}) \leq f(x_k)-q\cos^2(\theta_k)\|\nabla f(x_k)\|^2 \leq f(x_0)-q\sum_{j=0}^{k}\cos^2(\theta_j)\|\nabla f(x_j)\|^2,$$

which gives the inequality

$$\sum_{j=0}^{k}\cos^2(\theta_j)\|\nabla f(x_j)\|^2 \leq \frac{f(x_0)-f(x_{k+1})}{q}.$$

It holds:

$$\begin{aligned}\sum_{j=0}^{+\infty}\cos^2(\theta_j)\|\nabla f(x_j)\|^2 &= \lim_{k\to+\infty}\sum_{j=0}^{k}\cos^2(\theta_j)\|\nabla f(x_j)\|^2 \leq \lim_{k\to+\infty}\frac{f(x_0)-f(x_{k+1})}{q}\\ &= \frac{f(x_0)}{q}-\frac{1}{q}\lim_{k\to+\infty}f(x_{k+1}) \neq +\infty.\end{aligned}$$

This limit can not be infinite because Armijo rule implies the simple decrease of the objective function f, so it means that $x_k\in\Omega$, $\forall k\in\mathbb{N}$, so f is both lower and upper bounded.

This implies that the positive term series $\sum_{j=0}^{+\infty}\cos^2(\theta_j)\|\nabla f(x_j)\|^2$ is not divergent, so in this case it is also convergent. □

Implication of the theorem: Global convergence

Since the series $\sum_{j=0}^{+\infty} \cos^2(\theta_j)\|\nabla f(x_j)\|^2$ is convergent, it holds:

$$\lim_{k\to+\infty} \cos^2(\theta_j)\|\nabla f(x_j)\|^2 = 0.$$

There are two possibilities: the first one is that $\lim_{k\to+\infty} \nabla f(x_j) = 0$, which means that every accumulation point of $\{x_k\}$ is a stationary point for the function f.

The second possibility is that $\lim_{k\to+\infty} \cos(\theta_k) = 0$, which means that $\lim_{k\to+\infty} \nabla f(x_k)^T d_k = 0$. This situation occurs when $\nabla f(x_k)$ and d_k tend to be orthogonal and it can be avoided choosing a descent direction d_k such that $\cos(\theta_k) > M$ for some $M > 0$.
For example in the gradient descent method, selecting $d_k = -\nabla f(x_k)$ we have:

$$\cos(\theta_k) = -\frac{\nabla f(x_k)^T d_k}{\|\nabla f(x_k)\|\|d_k\|} = \frac{\nabla f(x_k)^T \nabla f(x_k)}{\|\nabla f(x_k)\|\|\nabla f(x_k)\|} = 1.$$

With this choice of $d_k = -\nabla f(x_k)$ the deriving gradient method is globally convergent to a stationary point for any choice of the starting point x_0 under the assumption of Zoutendijk's Theorem (2.1.2).

Note that the method does not necessarily converge to a minimum of the function but it is only guaranteed that the limit point is a stationary point. In order to be sure that the iteration gives a minimum, additional information from the Hessian of the function f has to be added in the computation of the descent direction d_k, as it happens on Newton method or Quasi-Newton methods [5].

2.2 Backtracking gradient method (BGM)

Applying the two rules already presented and choosing as descent direction the steepest descent direction $d_k = -\nabla f(x_k)$, yield the so-called the Backtracking Gradient Method (BGM).

An implementation of the BGM is reported in Algorithm 1, where the step length α_k is chosen using an adaptive strategy (the backtracking) and in such a way that it satisfies Armijo and Wolfe conditions.

Algorithm 1 Backtracking Gradient Method (BGM)

```
Given $x_0$, $\alpha_0$, $b_{max}$, $c_1 \in (0,1)$, $\gamma \in (0,1)$, $\epsilon$.
Set $k = 0$.
while $\|\nabla f(x_k)\| \geq \epsilon$ do
    for $b = 0, 1, ..., b_{max}$ do
        if $f(x_k - \alpha_k \nabla f(x_k)) < f(x_k) - \alpha_k c_1 \|\nabla f(x_k)\|^2$, i.e. Armijo condition then
            stop
        else
            $\alpha_{k+1} = \gamma \alpha_k$
        end if
    end for
    $x_{k+1} = x_k - \alpha_k \nabla f(x_k)$
    $k = k + 1$
end while
```

The backtracking algorithm works as follows: if $\alpha_k = \alpha_0$ does not satisfy Armijo rule, we reduce α_k by the multiplication with γ and this is repeated until the new step α_k satisfies Armijo.

This procedure is justified by Wolfe's lemma (Theorem 2.1.1) which ensures that it exists $\bar{\alpha}$ such that $\forall \alpha \in [0, \bar{\alpha}]$, α satisfies Armijo condition, so if $\alpha_k = \alpha_0 \geq \bar{\alpha}$ we decrease it multiplying by γ until the new $\alpha_k < \bar{\alpha}$. After a finite number of reductions we will find an appropriate α_k so the backtracking strategy never fails if Wolfe's lemma's assumptions are satisfied.

In the algorithm a maximum number of iterations b_{max} is fixed, because if $\bar{\alpha}$ is too small, we will be forced to take too small steps, leading to a slow convergence, so we stop before this may occur.

Since Wolfe's condition avoids to consider too small steps, and since b_{max} plays the same role, explicitly verifying Wolfe's condition would be redundant, for this reason only Armijo condition is checked in backtracking algorithm.

2.3 Constant step length gradient method (CGM)

We now present an implementation of the gradient method in which the step length is no longer chosen using and adaptive strategy but it is kept constant.
Assume $f : \mathbb{R}^n \to \mathbb{R}$ twice differentiable with Lipschitz continuous gradient, i.e.

$$\exists L_{\nabla f} > 0 \qquad \text{such that} \qquad \|\nabla f(x) - \nabla f(y)\| \leq L_{\nabla f} \|x - y\|, \qquad \forall x, y \in \mathbb{R}^n. \quad (2.7)$$

Then define an approximation of the function using the Taylor series of order one to define the model $m_k(d)$ as a function of the direction of descent d:

$$\begin{aligned} f(x_k+\alpha_k d) \leq & f(x_k)+\alpha_k(\nabla f(x_k))^T d+\frac{(\alpha_k)^2}{2}d^T H(z_k)d \\ & \leq \underbrace{f(x_k)+\alpha_k(\nabla f(x_k))^T d+\frac{(\alpha_k)^2 L_{\nabla f}}{2}\|d\|^2}_{m_k(d)}, \end{aligned} \tag{2.8}$$

where $z_k \in [x_k, x_k+\alpha_k d]$, $H(x)$ is the Hessian matrix of the function f.

The step length α_k is selected by minimizing the model $m_k(d)$, that is by annihilating its gradient:

$$\alpha_k d_k = -\frac{1}{L_{\nabla f}}\nabla f(x_k).$$

The equation shows that at each iteration k the step length can be chosen to be always equal to $\alpha_k = -\frac{1}{L_{\nabla f}}$. Therefore the corresponding iteration scheme is given in Algorithm 2.

Algorithm 2 Constant Step Size Gradient Method (CGM)

```
Given $x_0$, $L_{\nabla f}$, $\epsilon$.
Set $k = 0$.
while $\|\nabla f(x_k)\| \geq \epsilon$ do
    $x_{k+1} = x_k - \frac{1}{L_{\nabla f}}\nabla f(x_k)$
    $k = k + 1$
end while
```

2.4 Numerical results in image restoration problem

In this section we will show the compared results applying both the backtracking gradient method (BGM) and the constant step size gradient method (CGM) to the image restoration problem that has been presented in Chapter 1.

In particular, considering the notation previously introduced, we want to solve the problem:

$$\min_{x\in\mathbb{R}^n} f(x) = \|Ax-z\|_2^2 + \lambda h_\eta(Dx), \tag{2.9}$$

where, $f : \mathbb{R}^n \to \mathbb{R}$, is smooth, convex, $\bar{x}$ is the original image, z is the degraded observation, h_η is the Huber function with Huber parameter η, $\lambda \geq 0$ is the regularization parameter and D an high pass filter operator.

In our experiments we fixed the initial parameters in Algorithms 1 and 2: $b_{max} = 20$,

$c_1 = 10^{-4}$, $\gamma = 0.5$, $\epsilon = 10^{-7}$ and we started from x_0 random vector. Moreover, we considered the following setting:

- $\bar{x} \in \mathbb{R}^n$, $n = 512^2$ is the vectorized version of an image,
- $z = A\bar{x} + b$, where $A \in \mathbb{R}^{512^2 \times 512^2}$, square matrix that performs the blur, by the convolution of the image with a gaussian filter of variance 3 (see Fig. 2.1) and b is the realization of gaussian noise with variance 0.03.

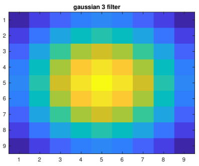


Figure 2.1: Gaussian filter for the blur with variance 3.

- D is a tall matrix $D = \begin{bmatrix} D_1 \\ D_2 \end{bmatrix} \in \mathbb{R}^{2n \times n}$ where D_1 performs the horizontal differences between adjacent pixels on the image and D_2 performs the vertical differences.
- The hyper-parameter λ is fixed: $\lambda = 0.05$.

We now discuss the results obtained using CGM and the BGM for different values of the Huber parameter η. In particular, focusing on the behaviour of both algorithms when the η is decreased, which means that the Huber function is close to be equal to the l_1 norm and the problem is close to a non-smooth problem.

In these experiments, the noise level is evaluated with a signal-to-noise-ratio (SNR) [7] measure defined as

$$10 \log_{10} \left(\frac{\text{variance of } \bar{x}}{\text{variance of noise}} \right).$$

The higher the SNR is and the more the restored image is considered a good approximation of the original one.

In Fig. 2.2 and Fig. 2.4 the first picture represents the degraded observation and the second is the decrease of the objective function analysis, where the red line is the CGM and the blue line is the BGM.
In Fig. 2.3 and Fig. 2.5 the comparison between the restored images after 100 iterations of the two methods is proposed.

Huber parameter $\eta = 1$

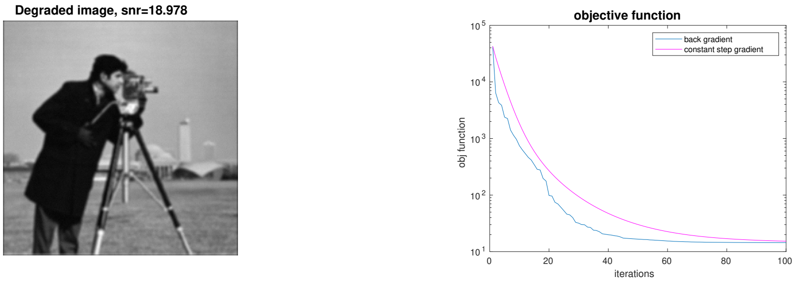


Figure 2.2: Results for Huber parameter $\eta = 1$, after 100 iterations.

(a) CGM restored image

(b) BGM restored image

Figure 2.3: Restored images comparison for $\eta = 1$, after 100 iterations.

We now decrease the value of the Huber parameter η from 1 to 0.01 and we repeat the same analysis of the results in Fig 2.4 and in Fig 2.5.

Huber parameter $\eta = 0.01$

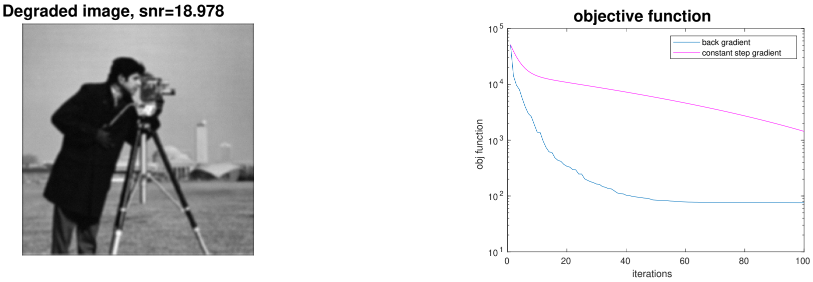
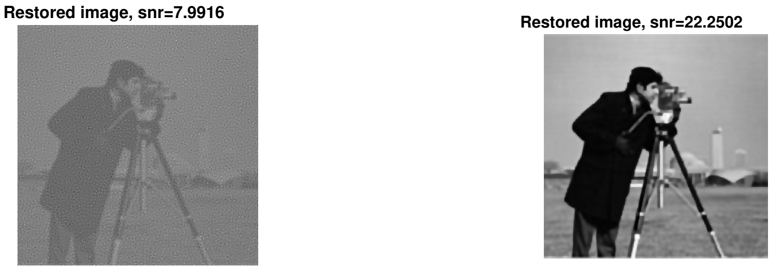


Figure 2.4: Results for Huber parameter $\eta = 0.01$, after 100 iterations.



(a) CGM restored image

(b) BGM restored image

Figure 2.5: Restored images comparison for $\eta = 0.01$, after 100 iterations.

Observations:

It seems clear from the results that for large values of Huber parameter, for example $\eta = 1$, both CGM and BGM have better performances in terms of decrease of the objective function and quality of restoration after the same number of iterations.

The more the problem is close to a non smooth problem, which is equivalent to choosing a small η parameter, and the more both algorithms slow down their convergence and also the quality of the restoration decreases.

Comparing the CGM to the BGM, it seems clear that in inverse problems, an adaptive step length computation is more efficient, since it brings to a faster convergence and an higher quality restored image after the same number of iterations.

In the next chapters, considering what has been observed so far, the Huber parameter η will be fixed to $\eta = 1$, that is the Huber function is expected to behave simile to the l_2 norm.

Since this study is a preliminary analysis, the problem that has been considered is a small size problem and for this reason it is not provided an analysis in terms of computational time.

When larger problems are considered, gradient methods can have really slow convergence, especially dealing with ill-posed problem, for example when the matrix $A^TA + \lambda D^TD$ is ill-conditioned.

For this reason, other methods could be used to solve the problem. One possibility, assuming the objective function to be twice differentiable, could be to use a second order Taylor model to create the approximation of the objective function used to define the step α_k. Newton methods, or Quasi Newton methods are some examples of these two order methods [5].

Chapter 3

Multilevel gradient method

In order to accelerate the performance of the gradient method it is possible to introduce a multilevel approach. The main idea that stands behind the class of multilevel methods is to reduce the cost of the step computation at each iteration by reducing the dimension of the problem and by exploiting the knowledge of alternative simplified expressions of the objective function.

This class of methods, called multigrid methods [3], is often used in partial differential equations (PDEs) with good results both in terms of computational time and decrease of the objective function, as shown in the paper [4].

In the next section, the general theory of multilevel methods will be presented and applied to the inverse problem in image restoration.

3.1 General multilevel scheme

Let us consider a minimization problem of the form:

$$\min_{x \in \mathbb{R}^n} f(x),$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a bounded below, continuously differentiable function.

Assumption

1. We assume to know a collection of continuously differentiable functions $\{f^l\}_{l=1}^{l_{max}}$, where $n_l \geq n_{l-1}$, $f^l : \mathbb{R}^{n_l} \to \mathbb{R}$ and $f^{l_{\max}}(x) = f(x)$.
   This collection of functions has to be such that for each $l = 2, \dots, l_{\max}$, f_l is more costly to minimize than f_{l-1}.

2. We have at disposal some operators to transfer variables from one level to another: restriction operators $R^l : \mathbb{R}^{n_l} \to \mathbb{R}^{n_{l-1}}$ and prolongation operators $P^l : \mathbb{R}^{n_{l-1}} \to \mathbb{R}^{n_l}$ [4].

It is possible to summarize the structure of the multilevel as in Table 3.1:

level	variables	approximation
$l_{\max}$	$x^{l_{\max}} \in \mathbb{R}^{n_{l_{\max}}}$	$f^{l_{\max}} = f$
$\vdots$		$\vdots$
$l+1$	$x^{l+1} \in \mathbb{R}^{n_{l+1}}$	f^{l+1}
	$R^{l+1} \Downarrow \quad \Uparrow P^{l+1}$	
l	$x^{l} \in \mathbb{R}^{n_{l}}$	f^{l}
$\vdots$		$\vdots$
1	$x^{1} \in \mathbb{R}^{n_{1}}$	f^{1}

Table 3.1: Hierarchy structure in the multilevel framework.

Example of restriction and prolongation operators

The standard R^l and P^l operators are simple restriction and interpolation operators which are commonly used with PDEs [3]. A simple example of 3×3 matrix is proposed below in order to give an idea on how variables are transferred from one level to another. As the example shows, the dimension of the matrix is increased by simply adding the mean value between two adjacent entrances of the matrix in the position between each couple of elements.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \quad \rightarrow \quad \mathrm{P}^l \quad \rightarrow \begin{bmatrix} 1 & 1.5 & 2 & 2.5 & 3 \\ 2.5 & 3 & 3.5 & 4 & 4.5 \\ 4 & 4.5 & 5 & 5.5 & 6 \\ 5.5 & 6 & 6.5 & 7 & 7.5 \\ 7 & 7.5 & 8 & 8.5 & 9 \end{bmatrix}$$

It is a common choice in multigrid theory, to construct the prolongation operator P^l and to consider the relationship $R^l = \alpha(P^l)^T$ to determine R^l. The scalar α is often set $\alpha = \frac{1}{2^d}$, where d is the dimension of the problem considered. For example the image restoration problem is a two dimensional problem, so we set $\alpha = \frac{1}{2^2} = \frac{1}{4}$ [3].

Note that P^l and R^l are not orthogonal operators, so starting from a matrix, restricting it and after projecting it back, does not give exactly the original matrix.

3.2 Two level method

For simplicity we reduce the analysis to the two level case, introducing the index h to denote the fine level or upper level and H to denote the coarse level or lower level.

In this particular case the general scheme becomes:

$$f = f^h : \mathbb{R}^{n_h} \to \mathbb{R}, \qquad f^H : \mathbb{R}^{n_H} \to \mathbb{R},$$

$$R : \mathbb{R}^{n_h} \to \mathbb{R}^{n_H}, \qquad P : \mathbb{R}^{n_H} \to \mathbb{R}^{n_h}.$$

The problem that has to be solved in this new notation has the form:

$$\hat{x} \in \arg\min_{x \in \mathbb{R}^n} f(x) \iff \hat{x}^h \in \arg\min_{x^h \in \mathbb{R}^{n_h}} f^h(x^h). \tag{3.1}$$

Note that the aim is to minimize the function f^h at fine scale, while the coarse approximation f^H is just a tool that is used to reduce the computational cost of the algorithm, it does not necessarily need to be decreased because there is nothing ensuring that minimizing f^H, also the fine function f^h is decreased.

For this reason, it needs to be defined a model at coarse scale starting from the coarse function f^H which is less costly to minimize than the function f^h at fine scale and such that it ensures that the fine objective function is decreased while the coarse model is minimized (see Section 3.2.1).

Moreover, a condition stating when it is advantageous to use the coarse model has to be defined (see Section 3.2.2) together with a starting point to start the coarse minimization (see Section 3.2.1).

3.2.1 Coarse model construction

Consider the k-th iterate at fine level x_k^h, in order to define a starting point for the coarse minimization the current fine iterate is simply restricted to coarse scale:

$$x_{0,k}^H = R x_k^h,$$

where in this notation, the k refers to the iteration at fine scale, while the 0 tells that this is the starting point at coarse scale. The index of the iteration at coarse scale will be denoted by ℓ.

If at fine iteration k it is decided to perform a coarse minimization, the coarse model $m_k^H(s)$ is defined as function of the step direction s and a regularization term has to be added to the coarse function f^H, this is due to the fact that, minimizing f^H is in general not sufficient to guarantee that f^h is minimized as well[4]. So let us consider the vector

$$v_H = R \nabla f^h(x_k^h) - \nabla f^H(x_{0,k}^H), \tag{3.2}$$

and define the coarse model as:

$$m_k^H(s) = f^H(x_{0,k}^H + s) + \underbrace{(R \nabla f^h(x_k^h) - \nabla f^H(x_{0,k}^H))}_{v_H}{}^T s. \tag{3.3}$$

In this way if the minimization of the model after m iteration yields a coarse step direction $s^H_{m,k}$, we will show that the definition of such a model ensures that the angle between $\nabla f^h(x^h_k)$ and $Ps^H_{m,k}$ is equal to the angle between $\nabla m^H_k(0)$ and $s^H_{m,k}$.
In order to prove this fact, we first show:

$$(R\nabla f^h(x^h_k))^T s^H_{m,k} = \nabla m^H_k(0)^T s^H_{m,k} \sim m^H_k(s^H_{m,k}) - m^H_k(0) < 0. \tag{3.4}$$

From this relation, if we project the coarse step to the fine level, defining $s^h_k = Ps^H_{m,k}$ we have that the fine step is a descent direction for f^h in a neighbourhood of x^h_k in fact:

$$\nabla f^h(x^h_k)^T s^h_k = \nabla f^h(x^h_k)^T Ps^H_{m,k} = (R\nabla f^h(x^h_k))^T s^H_{m,k} = \nabla_s m^H_k(0)^T s^H_{m,k} < 0. \tag{3.5}$$

In order to obtain the coarse step $s^H_{m,k}$, the model m^H_k can be minimized using any optimization method (gradient method, Newton, etc.). For example applying the gradient method the coarse model scheme gives:

$$s^H_{\ell+1,k} = s^H_{\ell,k} - \tau_\ell \nabla m^H_k(s^H_{\ell,k}), \tag{3.6}$$

$$\nabla m^H_k(s^H_{\ell,k}) = (\nabla f^H(x^H_{0,k} + s^H_{\ell,k}) + R\nabla f^h(x^h_k) - \nabla f^H(x^H_{0,k})), \tag{3.7}$$

where the step τ_ℓ can be chosen using a backtracking strategy. After m iterations the final coarse step will be:

$$s^H_{m,k} = \sum_{\ell=1}^{m} -\tau_\ell \nabla m^H_k(s^H_{\ell,k}). \tag{3.8}$$

The strength of the multilevel approach is the fact that for a single fine scale iteration, at coarse scale much more steps are performed and they are less costly than computing the same number of iterations at higher level (see Algorithm 3).

Even though there are no theoretical results, it has also been observed in PDEs that when the function is minimized at coarse scale, the resulting step direction is usually a better descent direction than the one that could have been computed at fine scale.

After the minimization of the coarse model the step is projected at fine scale and the fine iterate is updated using a backtracking gradient step:

$$x^h_{k+1} = x^h_k - \gamma_k Ps^H_{m,k} = x^h_k - \gamma_k s^h_k. \tag{3.9}$$

We can briefly resume the procedure with the scheme in Fig 3.1.

$$\begin{tikzcd}
x_k^h \arrow[r, dotted, "s_k^h = -\gamma_k \nabla f^h(x_k^h)"] \arrow[d, "R"'] & x_{k+1}^h = x_k^h + s_k^h \\
x_{0,k}^H := R x_k^h \arrow[r, "\min\limits_{s^H \in \mathbb{R}^{n_H}} m_k^H(s^H)", "\text{m iterations}"'] & s_{m,k}^H \arrow[u, "s_k^h = P s_{m,k}^H"']
\end{tikzcd}$$

Figure 3.1: Scheme of iteration k of the multilevel procedure. Option 1 (dotted line): take a gradient step at high level. Option 2 (straight line): exploit lower level model: take m steps of an optimization method to decrease the lower model.

3.2.2 Coarse step condition

The last step that needs to be done is defining a condition stating when the coarse model can be used instead of performing a backtracking gradient step at fine scale.

Obviously, it is not always possible to use the lower level model. For example, it may happen that $R\nabla f(x_k^h)$ lies in the nullspace of R which means that $R\nabla f(x_k^h)$ is zero while $\nabla f(x_k^h)$ is not. In this case, the current iterate appears to be first-order critical at lower level while it is not at higher level. Using the model m_k^H is hence potentially useful only if $R\nabla f(x_k^h)$ is large enough compared to $\nabla f(x_k^h)$. [4]
We therefore restrict the use of the model m_k^H to iterations where

1. $\|R\nabla f^h(x_k^h)\| > \kappa\|\nabla f^h(x_k^h)\|$,
2. $\|R\nabla f_h(x_k^h)\| > \theta$,

for some constant $\kappa \in (0; min\{1; \|R\|\})$ and where $\theta \in (0, 1)$.

Algorithm 3 Multilevel Gradient Method (MGM)

```
Set $k = 0$ and initialize $x_k^h$.
while $\nabla f^h(x_k^h) \geq \epsilon$ do
    if $\|R\nabla f^h(x_k^h)\| > \kappa\|\nabla f^h(x_k^h)\|$ and $\|R\nabla f^h(x_k^h)\| > \theta$ then
        Project to smaller dimension
        $x_{0,k}^H = Rx_k^h,\ s_{0,k}^H = 0$
        Minimize the coarse model to find the coarse step
        for $\ell = 0, \dots, m-1$ do
            $|\nabla m_k^H(s_{\ell,k}^H) = \big(\nabla f^H(x_{0,k}^H + s_{\ell,k}^H) + R\nabla f^h(x_k^h) - \nabla f^H(x_{0,k}^H)\big)$
                $\lfloor\ s_{\ell+1,k}^H = s_{\ell,k}^H + -\tau_\ell \nabla m^H(s_{\ell,k}^H)$
                    Reproject into the original domain
                    $x_{k+1}^h = x_k^h - \gamma_k P(s_{m,k}^H)$
        end for
    else
        $x_{k+1}^h = x_k^h - \gamma_k \nabla f^h(x_k^h)$
    end if
    $k = k + 1$
end while
```

3.3 Multilevel in image restoration

In this section it will be presented a version of the MGM algorithm applied to the inverse problem in image restoration. The aim of the study is to identify some situations in which a multilevel approach can be more effective with respect to the backtracking gradient method. Since it is a preliminary study on a small dimension problem, we will not focus on the computational time comparison but only on decreasing the objective function and providing a good quality restored image.

We consider an inverse problem in image restoration choosing the Huber function on the regularization term in order to have an objective function which is smooth. This choice is justified by the fact that as a preliminary study, the problem is constructed in such a way that it has a similar formulation to PDEs minimization problems [4].
It has already been proved that in PDEs the multilevel scheme can be really effective, so the first goal is to recreate similar conditions for the image problem to see if the MGM brings the same results as in PDEs and if it so, in which cases it happens.

3.3.1 Problem formulation

The problem to solve is to find a minimizer of:

$$f(x) = \|Ax - z\|_2^2 + \lambda h_\eta(Dx), \tag{3.10}$$

which using the notation of the two level scheme becomes:

$$f^h(x^h) = \|A^h x^h - z^h\|^2 + \lambda_h h_\eta(D^h x^h), \tag{3.11}$$

where

- $\bar{x}^h \in \mathbb{R}^{n_h}$ is the vectorized original image (dimensions usually 512 × 512);
- $z^h = A^h \bar{x}^h + b^h$, where $A^h \in \mathbb{R}^{n_h \times n_h}$ performs the blur convolving the image with a gaussian filter and b is the realization of gaussian noise;
- $D^h \in \mathbb{R}^{2n_h \times n_h}$ is a tall matrix $D^h = \begin{bmatrix} D_1^h \\ D_2^h \end{bmatrix}$, where D_1^h performs the horizontal differences between adjacent pixels on the image and D_2^h performs the vertical differences;
- The hyper-parameter $\lambda_h \geq 0$, is fixed before performing the method.

In order to construct the coarse model m_k^H we need to define a coarse function f^H and then modify it adding the regularization term v^H.
As first step of the procedure, the variables and the operators need to be restricted in order to be used at coarse scale. The procedure applied is a standard choice in multigrid theory as explained in [3]:

- $A^H = RA^hP$, restricted blur matrix,
- $z^H = Rz^h$, restricted observed image,
- $D^H = \begin{bmatrix} D_1^H \\ D_2^H \end{bmatrix} = \begin{bmatrix} RD_1^hP \\ RD_2^hP \end{bmatrix}$ restricted high pass filter operator.

At this point the coarse function f^H and the coarse model m_k^H can be defined:

$$f^H(x^H) = \|A^H x^H - z^H\|_2^2 + \lambda_H h_\eta(D^H x^H), \tag{3.12}$$

$$\nabla f^H(x^H) = 2(A^H)^T(A^H x^H - z^H) + \lambda_H (D^H)^T \nabla h_\eta(D^H x^H), \tag{3.13}$$

$$m_k^H(s) = f^H(x_{0,k}^H + s) + \underbrace{(R\nabla f^h(x_k^h) - \nabla f^H(x_{0,k}^H))}_{v^H}{}^T s. \tag{3.14}$$

Once the model is defined the MGM algorithm previously described can be applied in this context and the results can be compared to the ones obtained using the backtracking gradient method.

3.4 Numerical comparison of the MGM and the backtracking gradient method

We performed experiments where the original image is 512 × 512 pixels, both blur and noise are considered to be Gaussian, the Huber parameter η is fixed ($\eta = 1$), the maximum number of coarse iterations allowed at coarse scale fixed to 50 and the coarse condition parameter $\kappa = 0.3$. In all the experiments x_0^h is selected as a random vector. The analysis with different value of noise, blur and parameters is provided in the following sections in order to individuate particular configurations in which the use of the MGM may be advantageous .

3.4.1 Ideal situation

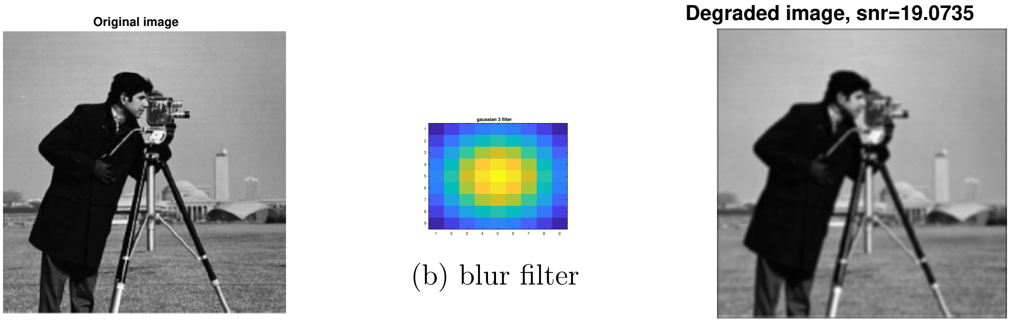
After trying different combinations of noise, blur and Hyper-parameters, it has been observed that, in order for the multilevel to be effective with respect to the gradient method, the gradient method's convergence has to be slow.

Since the behaviour of the MGM is really affected by the choice of parameters, it is convenient to move the focus on those situations in which the backtracking gradient's convergence is particularly slow. If the gradient is not too fast, the multilevel approach has more probability to decrease the function better than the gradient even for a choice of parameters which is not optimal.

Finding those particular instances is really useful because it is not always possible to set the best parameters' configuration since in many cases some hyper-parameters are kept constant .
Configurations in which the convergence of the gradient has been observed to be slow are:

- Huber parameter $\eta = 1$,
- high value of gaussian filter variance on blur operator,
- low value of gaussian noise's variance,
- small λ_h hyper-parameter.

This is a particularly promising situation for the MGM and it is described in Fig. 3.2, fixing the gaussian noise at 0.5% so the resulting SNR of the degrade image will be 19.07, fixing the gaussian filter variance for the blur at 3 and the $\lambda_h = 0.005$ we have that:



(b) blur filter

(a) Original image $\bar{x}$

(c) Degraded image z

Figure 3.2: Initial configuration on ideal case.

Starting from this configuration and applying both methods, since the problem is convex they converge to the same optimal value which gives the restored image in Fig. 3.3:

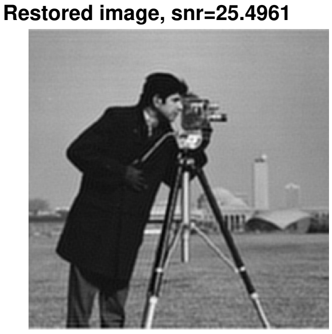
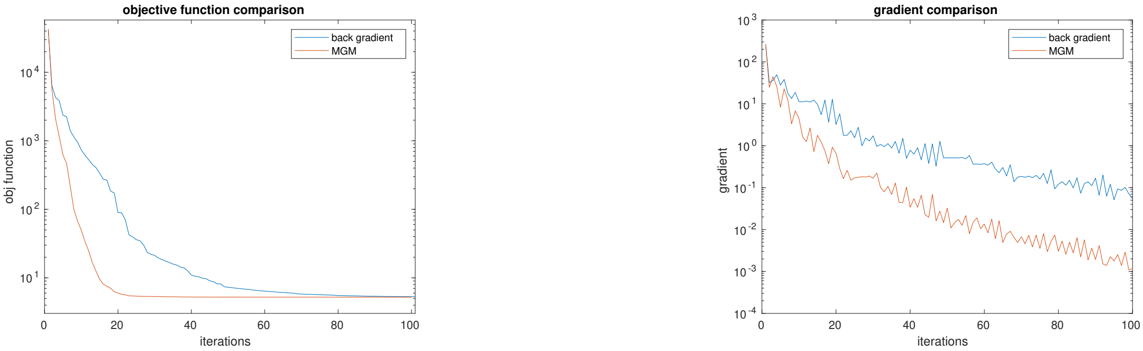


Figure 3.3: Restored image $\hat{x}$.

In Fig. 3.4 the comparison between MGM and backtracking gradient, is plotted in terms of the the decrease of the objective function in the first picture and the second, in terms of the convergence of the norm of the gradient to zero. In this plots the red line refers to the MGM and the blue line to the BGM.

We can observe in Fig. 3.4 that in this ideal situation, once the optimal parameters are fixed, the MGM is able to have a better decrease on the objective function and also the convergence of the norm of the gradient to zero seems faster.
As next step an analysis varying this ideal parameters will be provided in order to understand how the performance of both methods are affected by different types of variations on the model.



(a) Objective function

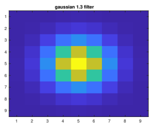
(b) Gradient

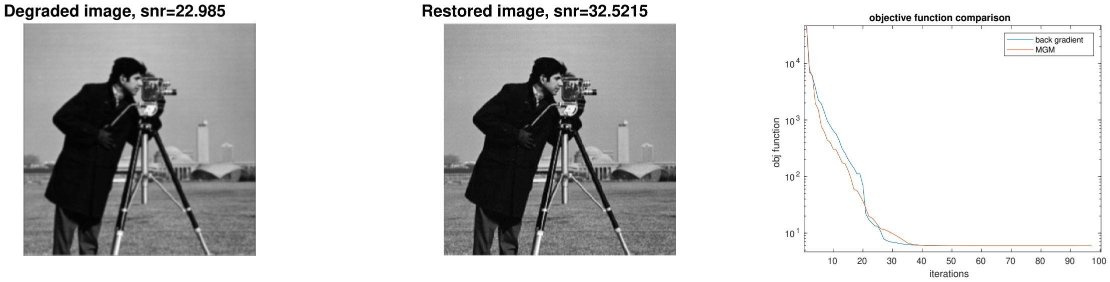
Figure 3.4: Objective function and norm of the gradient comparison.

3.4.2 Blur variation

Different gaussian filters have been applied to the original image in order to perform the initial degradation of the observation, while all the other parameters are kept constant. The results are presented decreasing the value of the variance of the gaussian filter whose convolution with the original image generates the blurred image. In other words the first image is less blurred while the last one has more blur.

Gaussian filter variance 1.3




Degraded image, snr=22.985
Restored image, snr=32.5215
objective function comparison
back gradient
MGM
obj function
iterations

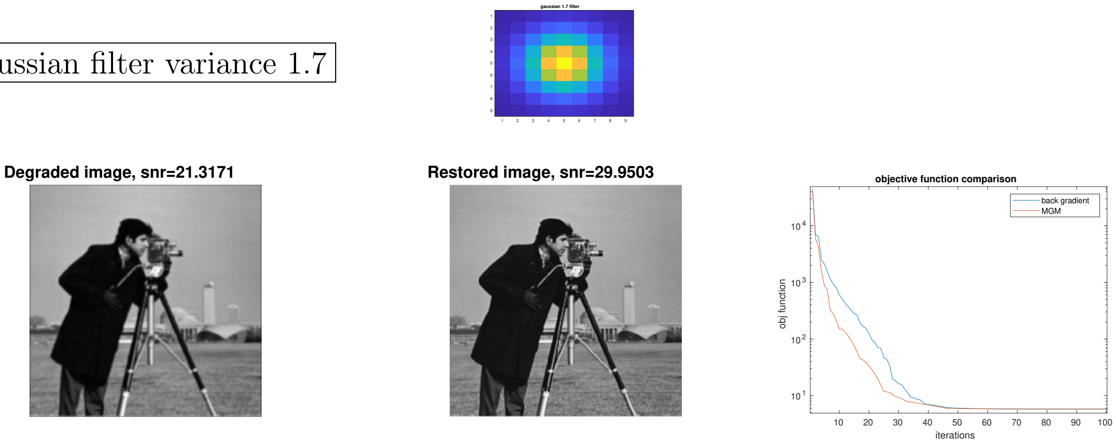
(a) Degraded image

(b) Restored image

(c) Objective function

Figure 3.5: Results comparison for gaussian filter variance 1.3.

Gaussian filter variance 1.7



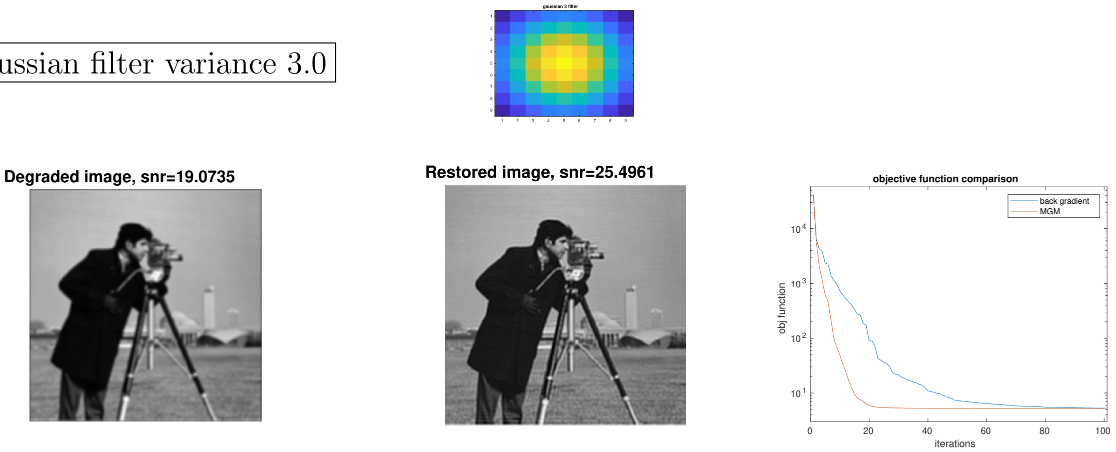
(a) Degraded image

(b) Restored image

(c) Objective function

Figure 3.6: Results comparison for gaussian filter variance 1.7.

Gaussian filter variance 3.0



(a) Degraded image

(b) Restored image

(c) Objective function

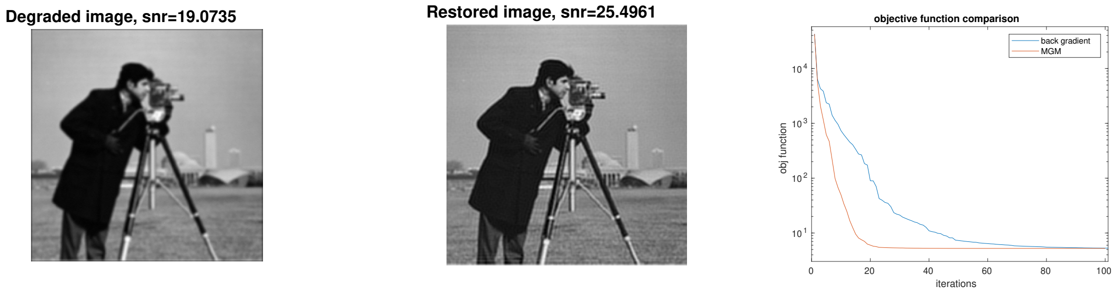
Figure 3.7: Results comparison for gaussian filter variance 3.

In Fig. 3.5, Fig. 3.6 and Fig. 3.7 we can observe that in all the situations the restored image improves the observation as the higher value of SNR index confirms. Clearly the restoration of one observation in which an high value of blur has been applied has a lower quality if compared to the restored image of a less blurred observation.
Analyzing the plots of the objective function, we can observe that for higher value of blur, the BGM as a slower convergence, allowing the MGM to gain significantly in decreasing the objective function.
This behaviour is particularly clear looking at the case with gaussian filter variance equal to 3 in Fig. 3.7, in which the red line of the MGM gets close to the minimum value of the function much faster than the blue line of the BGM.

3.4.3 λ_h hyper-parameter variation

In this section, fixing the gaussian filter variance of the blur to 3 and maintaining the other parameters unchanged, we try increasing value of the λ_h parameter on the definition of the objective function.
The λ_h parameter is strictly related to the value of noise. In order to have an high quality restored image, for small value of noise a small λ_h parameter has to be chosen and increasing the noise λ_h has to be increased as well.

$\boxed{\lambda_h = 0.05}$



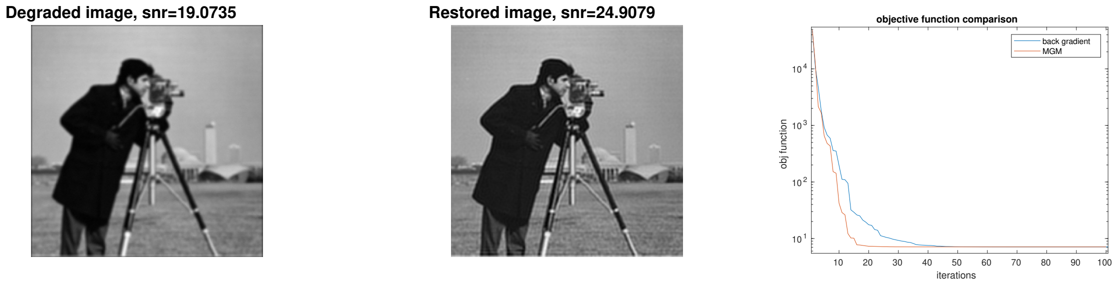
(a) Degraded image

(b) Restored image

(c) Objective function

Figure 3.8: Results comparison for $\lambda = 0.05$.

$\boxed{\lambda_h = 0.1}$



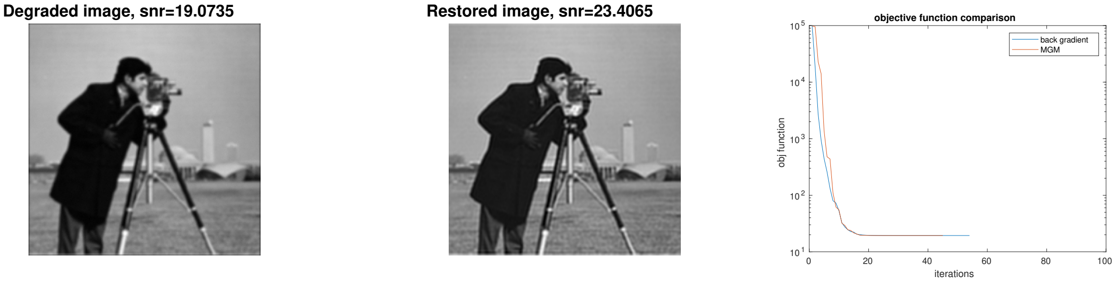
(a) Degraded image

(b) Restored image

(c) Objective function

Figure 3.9: Results comparison for $\lambda = 0.1$.

$\boxed{\lambda_h = 0.5}$



(a) Degraded image (b) Restored image (c) Objective function

Figure 3.10: Results comparison for $\lambda = 0.5$.

Changing the λ_h on the definition of the objective function, really affects the convergence of both methods. In particular, it the first case (Fig. 3.8) form a small $\lambda_h = 0.05$, the backtracking gradient method has a slow convergence if compared to the MGM, while increasing $\lambda_h = 0.5$ (Fig. 3.10), the backtracking gradient becomes faster and it gets close to the minimum value of the function in almost 20 iterations. In this last case, we can not observe a gain in decreasing the function using the MGM method because the gradient is already too fast.

Note that a faster convergence does not imply a better quality in the restored image, indeed in Fig. 3.10, even if the function decreases fast to the minimum, this minimum represents a restored image which is a lower quality approximation of the original, if compared to the one obtained in Fig. 3.8 for $\lambda_h = 0.05$, where the convergence is slower.

We could say that for small value of λ_h the gradient method is in general slow, so potentially the MGM represents a good alternative since, for a good choice of hyper-parameter, it decreases the objective function faster.

3.4.4 Instability of the model

One of the most important problems in the analysis and interpretation of the results is the initial choice of hyper-parameters used to set the problem.

In many situations a variation of just one hyper-parameter brings to very different behaviours of the MGM algorithm; let us provide some examples.
As it has been already discussed in previous sections, the main hyper-parameters involved in the MGM method are:

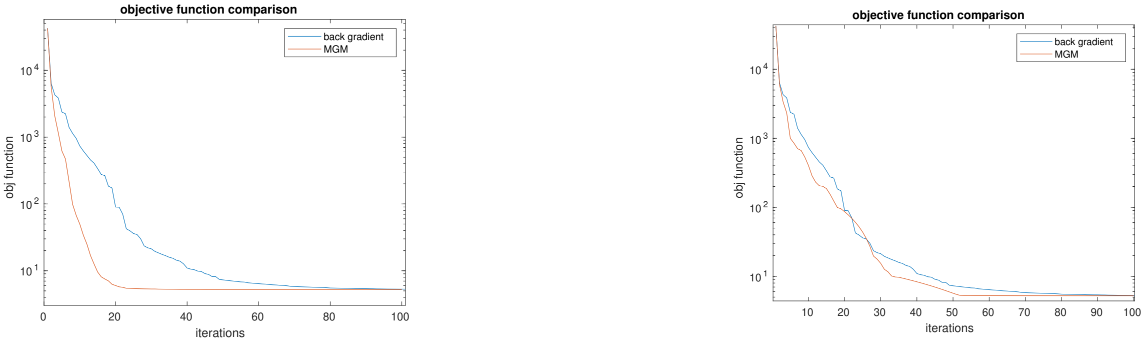
- λ_h: in the definition of the objective function,
- λ_H : in the definition of the coarse model,

- κ, θ : in the coarse step condition,
- m: maximum number of iterations at coarse scale.

Analysis of parameter m

The impact of λ_h variation has already been discussed, so we will provide an example of how the convergence of the MGM is affected by the choice of the maximum number of iteration at coarse scale.

A new situation will be analyzed, where all the parameters are unchanged with respect to the previous ideal case except for the maximum number of coarse iteration m, which is reduced from $m = 50$ to $m = 10$.



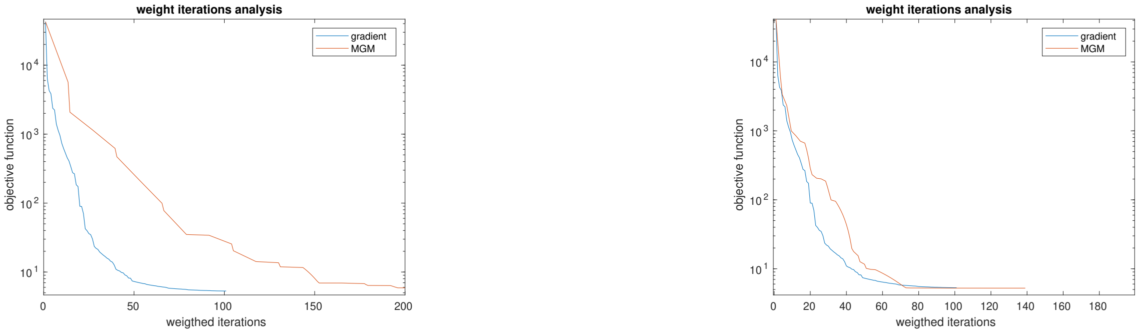
(a) Objective function, m=50

(b) objective function, m=10

Figure 3.11: effect of m parameter's variation on the objective function.

Setting $m = 50$ leads to a more efficient decrease of the objective function as the plots clearly show, but considering too many iterations at coarse scale is not in general the optimal alternative. The reason for which setting m too large may not be the best choice is that performing too many iteration to minimize the coarse model could be really computationally expensive.

To explain better this concept the weighted analysis of the objective function is provided in the next plots. This means that we penalized the multilevel considering each coarse step on the minimization of m_k^H as a $\frac{1}{4}$ of a standard step at fine scale. This kind of analysis allows to take into account the steps done at coarse scale, which unless would not be observable on a simple plot of the objective function.



(a) Weighted analysis, m=50

(b) Weighted analysis, m=10

Figure 3.12: Weighted analysis of objective function for different m parameters

It is clear from the plots in Fig. 3.12 that we can not increase m as much as we want, because it will lead the algorithm not to be computationally efficient, for this reason a compromise should be found between a reduced number of iterations at coarse scale and a sufficient decrease of the objective function.

Chapter 4

Multilevel and wavelet theory

4.1 Wavelet in brief

Wavelets theory has many applications in many fields, one of the most important is the signal and image processing. The purpose of a wavelets analysis of a signal or an image is to find a parsimonious representation which preserves the initial features of the signal/image but expresses the signal/image using a relatively small set of coefficients.

In wavelets analysis, two functions play a particularly important role: the scaling function or father wavelet ϕ and the mother wavelet or simply wavelet ψ. This two functions generate a family of functions that can be used to break up or reconstruct a signal or an image[2].
In the next section a brief overview of this topic will be presented.

Definition 4.1.1. *Multiresolution analysis Let $\{V_j\}$, $j = \ldots, -1, 0, 1, \ldots,$ be a sequence of subspaces in $L^2(\mathbb{R})$. The collection of spaces $\{V_j, j \in \mathbb{Z}\}$ is called a multiresolution analysis with scaling function ϕ if the following conditions hold:*

1. $V_j \subset V_{j+1}, \qquad \forall j \in \mathbb{Z}$
2. $\overline{\bigcup_{j\in\mathbb{Z}} V_j} = L^2(\mathbb{R})$
3. $\bigcap_{j\in\mathbb{Z}} V_j = \emptyset$
4. $f(x) \in V_j \iff f(2^{-j}x) \in V_0$
5. $\phi \in V_0$, *and the set $\{\phi(x-k), k \in \mathbb{Z}\}$ is an orthonormal basis for V_0 with respect to the inner product of $L^2(\mathbb{R})$*

The $V_j's$ are called *approximation spaces.*

The scaling functions ϕ can be of very different types, but the most useful class of scaling functions are those that have compact support. Different choice for ϕ may yield different multiresolution analyses.

Theorem 4.1.1. *[2, Theorem 5.5] Suppose $\{V_j, j \in \mathbb{Z}\}$ is a multiresolution analysis with scaling function ϕ, then for any $j \in \mathbb{Z}$, the set of functions*

$$\{\phi_{jk}(x) = 2^{\frac{j}{2}} \phi(2^j x - k); k \in \mathbb{Z}\}$$

is an orthonormal basis for V_j.

Theorem 4.1.2. *[2, Theorem 5.6] Suppose $\{V_j, j \in \mathbb{Z}\}$ is a multiresolution analysis with scaling function ϕ, then the following scaling relation holds:*

$$\phi(x) = \sum_{k\in\mathbb{Z}} p_k \phi(2x - k), \qquad \textit{where} \qquad p_k = 2\int_{-\infty}^{+\infty} \phi(x)\overline{\phi(2x - k)}dx.$$

Moreover

$$\phi(2^{j-1}x - l) = \sum_{k\in\mathbb{Z}} p_{k-2}\phi(2^j x - k),$$

equivalently

$$\phi_{j-1,k} = 2^{-\frac{1}{2}} \sum_{k\in\mathbb{Z}} p_{k-2}\phi_{j,k},$$

where $\phi_{jk}(x) = 2^{\frac{j}{2}}\phi(2^j x - k)$.

In order to visualize better these concepts, we propose the Haar wavelet example, which is the simplest wavelet possible. First we define the father wavelet or scaling function ϕ

$$\phi(x) = \begin{cases} 1 & \text{if } x \in [0, 1), \\ 0 & \text{otherwise.} \end{cases}$$

Now it is possible to define the approximation sets. V_0 will be the set of all functions of the form $\sum_{k\in\mathbb{Z}} a_k\phi(x - k)$, V_1 the set of all functions $\sum_{k\in\mathbb{Z}} a_k\phi(2x - k)$ and in general V_j the set of all functions $\sum_{k\in\mathbb{Z}} a_k\phi(2^j x - k)$.
Using Theorem (4.1.1) we have that the set of functions $\{\phi_{jk}(x) = 2^{\frac{j}{2}}\phi(2^j x - k); k \in \mathbb{Z}\}$ is an orthonormal basis of V_j.

In fact fixing $j = 0$ it holds:

$$\|\phi(x - k)\|^2_{L^2} = \int_{-\infty}^{+\infty} \phi(x - k)^2 dx = \int_k^{k+1} 1dx = 1,$$

$$\langle \phi(x-j), \phi(x-k) \rangle = \int_{-\infty}^{+\infty} \phi(x-j)\phi(x-k)dx = 0.$$

The second equality holds because the intersection between the supports of $\phi(x-j)$ and $\phi(x-k)$ is empty.

The next step on the study is to decompose V_{j+1} into the sum of V_j and its orthogonal complement which will be called W_j and as we did for V_j, identify a function ψ, whose translation generate all the set W_j.
First will be examined the Haar case with $j = 0$, and we have that $V_1 = V_0 \oplus V_0^{\perp}$ and we want to find a function ψ whose translates generate $V_0^{\perp}$. Such a function should have the properties:

- $\psi \in V_1$ so $\psi(x) = \sum_l a_l \phi(2x-l)$,
- $\psi \perp V_0$, so $\int_{-\infty}^{+\infty} \psi(x)\phi(x-k)dx = 0$, $\forall k \in \mathbb{Z}$.

A function of this type is $\psi(x) = \phi(2x) - \phi(2x-1)$ and it is called Haar wavelet.
The set $W_0 = V_0^{\perp}$ and it is the space of all functions of the form $\sum_{k\in\mathbb{Z}} a_k \psi(x-k)$.
This type of construction can be done starting from a general scaling function ϕ and approximation set V_j and it holds the following theorem.

Theorem 4.1.3. *[2, Theorem 5.10] Suppose $\{V_j, j \in \mathbb{Z}\}$ is a multiresolution analysis with scaling function*

$$\phi(x) = \sum_{k\in\mathbb{Z}} p_k \phi(2x-k).$$

Let W_j be the span of $\{\psi(2^j x - k); k \in \mathbb{Z}\}$, where

$$\psi(x) = \sum_{k\in\mathbb{Z}} (-1)^k \overline{p_{1-k}} \phi(2x-k).$$

Then $W_j \subset V_{j+1}$ is the orthogonal complement of V_j in V_{j+1}.
Furthermore, $\{\psi_{j,k} := 2^{\frac{j}{2}} \psi(2^j x - k); k \in \mathbb{Z}\}$ is an orthonormal basis for the W_j.

Iterating this decomposition we obtain:

$$V_j = W_{j-1} \oplus V_{j-1} = W_{j-1} \oplus W_{j-2} \oplus \cdots \oplus W_0 \oplus V_0.$$

This means that for all $f \in V_j$ we can write $f = w_{j-1} + w_{j-2} + \cdots + \Phi_0 + f_0$, where $w_j \in W_j$ and $f_0 \in V_0$.

As $j \to +\infty$ a similar decomposition can be extended to all function $f \in L^2(\mathbb{R})$ and the following theorem holds:

Theorem 4.1.4. *[2, Theorem 5.11] Suppose $\{V_j, j \in \mathbb{Z}\}$ is a multiresolution analysis with scaling function ϕ. Let W_j be the orthogonal complement of V_j in V_{j+1}, then*

$$L^2(\mathbb{R}) = \dots \oplus W_{-1} \oplus W_0 \oplus W_1 \oplus \dots$$

In particular each function of $L^2(\mathbb{R})$ can be uniquely expressed as a sum $\sum_{k=-\infty}^{+\infty} w_k$, with $w_k \in W_k$. Equivalently the set of all wavelets, $\{\psi_{j,k}\}_{j,k\in\mathbb{Z}}$ is an orthonormal basis for $L^2(\mathbb{R})$.

Thanks to those results it is possible to fix a scale $j = j_0$ and this will give an approximation of each function $f \in L^2(\mathbb{R})$ in terms of the coefficients of the decomposition in the base we presented up to scale j_0 [9], which means:

$$f(t) = \sum_{k=-\infty}^{\infty} \underbrace{\langle f, \phi_{j_0,k}\rangle}_{\text{smooth coefficients}} \phi_{j_0,k}(t) + \sum_{j=-\infty}^{j_0} \sum_{k=-\infty}^{\infty} \underbrace{\langle f, \psi_{j,k}\rangle}_{\text{details coefficient}} \psi_{j,k}(t).$$

Very efficient algorithms exist to easily evaluate both the wavelet decomposition of a discretized function (image/signal) and on the other hand, starting from the coefficients, recover the original function.
This algorithms are actually used when wavelet theory is applied in image or signal processing and denoising.

4.2 Wavelets and images

The next step will be to understand how wavelets can be used in image restoration problem and how they can be combined with multilevel theory.

For simplicity, the wavelet theory that has been presented refers to a continuous 1-D signal, when we deal with images we consider each image as the discretized version of a 2-D function $f(x, y)$ on a finite grid, where $f(x_0, y_0)$ represent the grey scale value at point (x_0, y_0) of the grid.

Keeping this in mind, in Fig. 4.1 we provide at first, an example of a decomposed image using Haar wavelet in order to visualize how the approximation and detail coefficients look like.

If we start from an original image $N \times N$ and we fix $j_0 = 1$, which means that the wavelet decomposition is done up to scale one, what we get is an approximation of the image $\frac{N}{2} \times \frac{N}{2}$ (approximation coefficient) and three matrices $\frac{N}{2} \times \frac{N}{2}$ containing the horizontal, vertical and diagonal details of the image (detail coefficients).

Notice that performing the wavelet decomposition up to scale 2 is equivalent to perform a wavelet decomposition up to scale 1 on the approximation coefficient that has been already obtained.

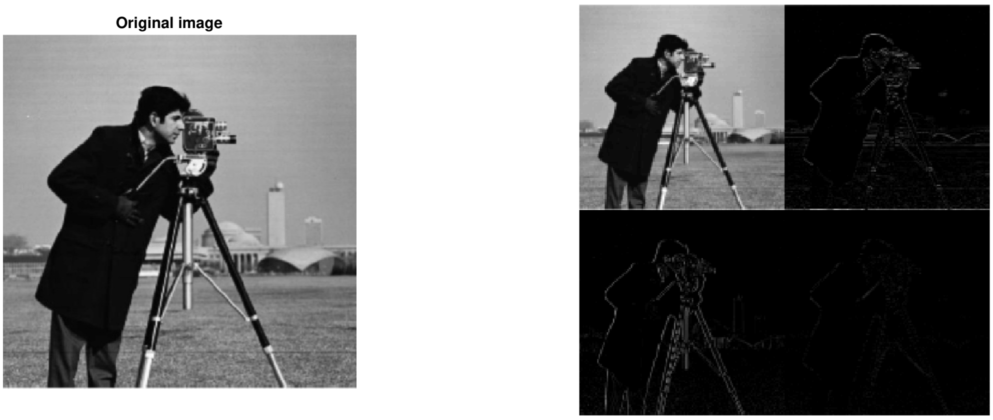


Figure 4.1: 1 level Wavelet decomposition.

4.3 Wavelet in multilevel scheme

Performing such a decomposition of an image allows in particular to have a smaller approximation at a different scale of the original image. Intuitively, this fact can be used in the multilevel scheme in order to define the starting point of the coarse minimization. In addition, wavelets can be used to provide a different version of the MGM in which the projection operator P and the restriction operator R (see Chapter 3) are substituted by others operators which perform 1 level wavelet transform and inverse wavelet transform.

The R and P operators play a fundamental role on the multilevel scheme. In Fig. 4.2 we provide an example of their behaviour when applied to images, in order to better understand how they can be replaced by others operators coming from wavelet theory.

Consider to start from an original image $\hat{x}$, then using the restriction operator we can obtain $R\hat{x}$, and this restricted image can be reprojected back using P operator: $PR\hat{x}$.

Note that R and P are not orthogonal operators, so what we get at the end is not the original image $\hat{x}$.

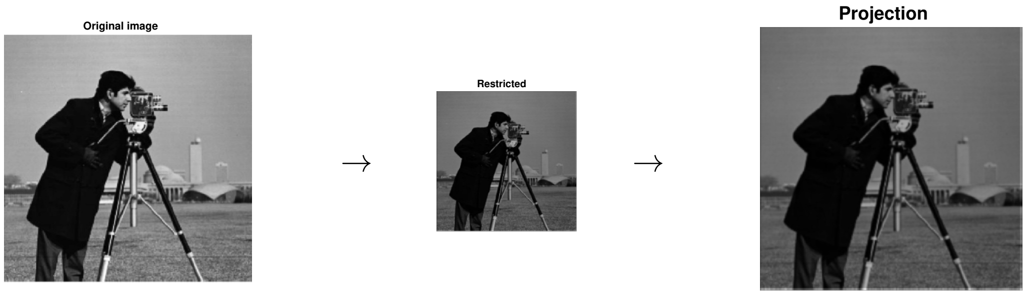


Figure 4.2: Restriction and prolongation example.

This procedure can be reproduced substituting R with the operator Φ_0 which performs the Haar wavelet transform on the original image, from which the approximation coefficients is taken as restricted image (see Fig. 4.3) .

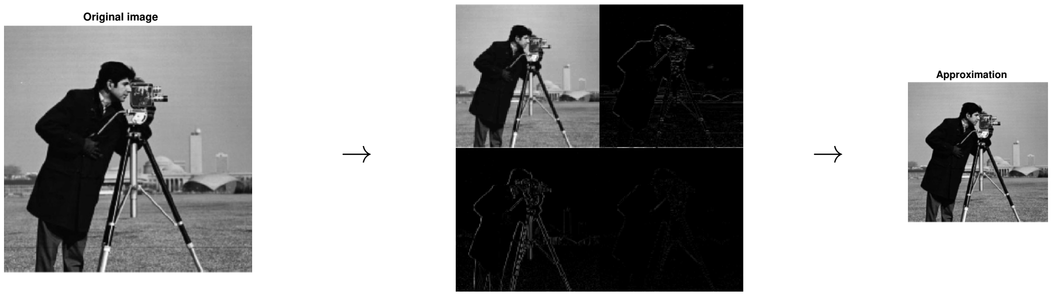


Figure 4.3: Φ_0 operator.

The P operator instead is substituted by a new operator called Φ_0^T that performs the inverse wavelet transform considering the details coefficient constantly equal to zero (see Fig. 4.4).
The choice of considering the details coefficients equal to zero when projecting from coarse scale to fine scale, is justified by the fact that in the MGM algorithm, the details coefficients are known only for the starting point $x_{0,k}^H$ of the coarse minimization but they can not be evaluated for the successive coarse iterates.

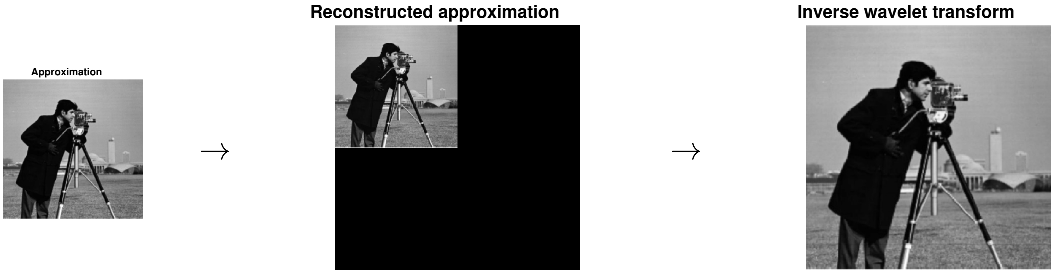


Figure 4.4: Φ_0^T operator.

4.3.1 MGM with wavelet operators Φ_0 and Φ_0^T

Consider the same problem at fine scale:

$$f^h(x^h) = \|A^h x^h - z^h\|^2 + \lambda_h h_\eta(D^h x^h). \tag{4.1}$$

In this new formulation of the MGM algorithm it only changes the coarse model construction, in which R and P operators are replaced by Φ_0 and Φ_0^T, so it becomes:

- $A^H = \Phi_0 A^h \Phi_0^T$, restricted blur matrix,
- $z^H = \Phi_0 z^h$, restricted observed image,
- $D^H = \begin{bmatrix} \Phi_0 D_1^h \Phi_0^T \\ \Phi_0 D_2^h \Phi_0^T \end{bmatrix}$ restricted high pass filter operator.

At this point the new coarse function $\tilde{f}^H$ and the new coarse model $\tilde{m}^H$ can be defined:

$$\tilde{f}^H(x^H) = \|A^H x^H - z^H\|_2^2 + \lambda_H h_\eta(D^H x^H), \tag{4.2}$$

$$\nabla \tilde{f}^H(x^H) = 2(A^H)^T(A^H x^H - z^H) + \lambda_H (D^H)^T \nabla h_\eta(D^H x^H), \tag{4.3}$$

$$\tilde{m}^H(s) = f^H(x_{0,k}^H + s) + \underbrace{(\Phi_0 \nabla f^h(x_k^h) - \nabla f^H(x_{0,k}^H))}_{\tilde{v}^H}{}^T s. \tag{4.4}$$

4.4 Numerical results

In this section we present the results obtained comparing the new version of the MGM with wavelets operators to the BGM and the standard MGM.

Unfortunately it has been not possible to identify well defined situations in which the new version of the MGM is more effective than the previous one, but one important feature of the new algorithm emerges in several of the studied configurations.

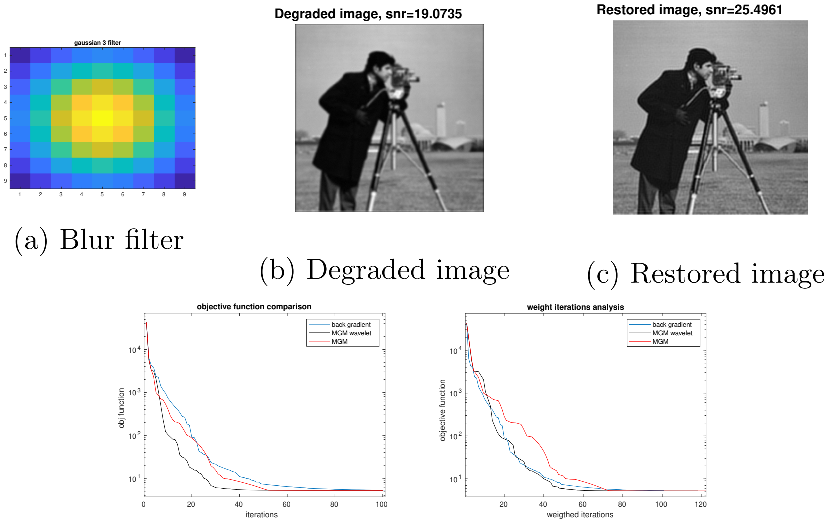
It has been observed that decreasing m, the maximum number of iterations allowed to minimize the coarse model m_k^H can have negative effects on the convergence of the MGM; for this reason in previous experiments we considered $m = 50$, in order to have good results in terms of decrease of the objective function.

What emerges from the experiments on the MGM using wavelets operators Φ_0 and Φ_0^T is that, in general the maximum number of iterations needed at coarse scale m can be decreased and still obtain good results.
It seems that the coarse model constructed using wavelets needs less iterations to be minimized in order to provide a good descent direction $s_{m,k}^H$.
One reason that can justify this behaviour could be the fact that the approximation $\Phi_0 x_k^h$, from which the coarse model starts to be minimized, is potentially a better approximation than Rx_k^h used in standard MGM.

As it has been considered in previous analyses, the original image is 512×512 pixels, both blur and noise are Gaussian, the Huber parameter η is fixed ($\eta = 1$), the maximum number of coarse iterations allowed at coarse scale is decreased from 50 to 10, the coarse condition parameter remains $\kappa = 0.3$ for the standard MGM while it is $\tilde{\kappa} = 0.9$ for the wavelet MGM.

In Fig. 4.5 and Fig. 4.6 we present the gaussian filter used to perform the blur, the degraded and the restored image, the plot of the objective function and the weighted objective function in which the blue line is the backtracking gradient method, the red line is the standard MGM and the black line is the wavelet MGM.

First situation: In this first case it has been considered a 0.5% noise, a blur given by the convolution with a gaussian filter with variance 3 and $\lambda_h = 0.05$.



(a) Blur filter

(b) Degraded image

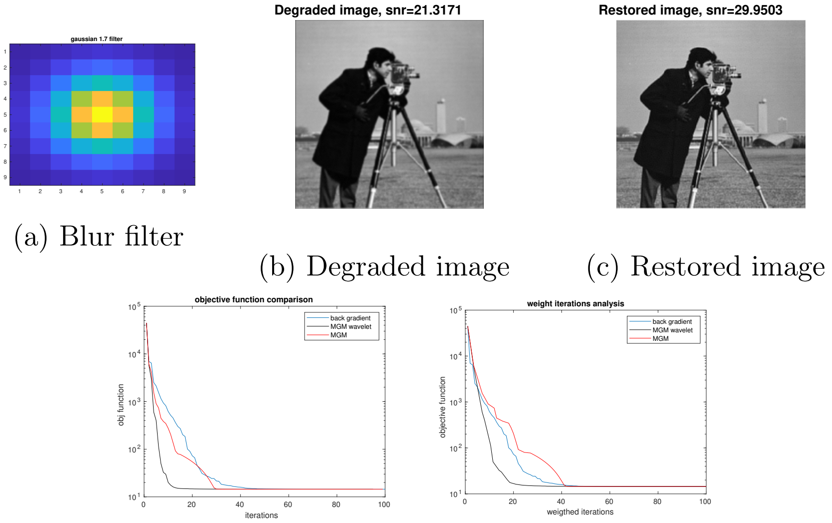
(c) Restored image

(d) Objective function

(e) Weighted function

Figure 4.5: Results comparison in first situation.

Second situation: In this second case it has been considered a 0.1% noise, a blur given by the convolution with a gaussian filter with variance 1.7 and $\lambda_h = 0.05$.



(a) Blur filter

(b) Degraded image

(c) Restored image

(d) Objective function

(e) Weighted function

Figure 4.6: Results comparison in second situation.

Although it is not observable for all the choice of parameters, in several cases, as in the two showed above, it seems that few number of iterations at coarse scale are needed for the wavelet MGM to obtain a good restored image and a decrease on the objective function faster than the other two methods.

It is not always true that for any choice of parameters the wavelet MGM has better results than the standard MGM, but it could be an alternative algorithm to solve the problem.
In these experiments, we selected the Haar wavelet, since it is similar to the standard R and P operators, but different type of wavelets can be used in the algorithm.

Conclusions

It emerges from this preliminary study of the multilevel scheme applied in image restoration problem, that it has been possible to identify some situations in which it could be worth it to apply the MGM in order to solve the problem. For example for high value of blur and small regularization parameter. The gain that has been observed is only in terms of decrease of the objective function, but one further goal of the study is to consider a larger problem and optimize the algorithm in order to have similar results also in terms of computational time.

Furthermore, since only the two level case has been tested, the number of levels on the multilevel scheme has to be increased in order to spend less computational time on the minimization at coarse scale.

It has also been observed that wavelets operators can be used to transfer information from one level to another, replacing R and P operators. The resulting approximated image at coarse scale obtained restricting the original image with the operator Φ_0, seems to be a better approximation than the one obtained using R, since the minimization at coarse scale needs less iterations to give consistent results.

On the current version of the algorithm the information contained in the detail coefficients of the wavelet transform of the current image are not used; for this reason another goal will be to include such information in order speed up the minimization process.

One last goal for the future is to design a multilevel scheme for the non smooth case, for example considering the l_1 norm on the regularization term in order to promote sparsity. This change on the design of the problem does not allow to use minimization techniques requiring smoothness of the objective function such as the gradient method.

For this reason the multilevel method should be applied to another class of method called proximal methods. Even though some algorithm has been proposed [6], the field of multilevel proximal methods is still growing and the future goal of this study is to contribute to enlarge the theory and to provide practical solutions to solve the general image restoration problem.

Bibliography

[1] A. Beck and M. Teboulle. Smoothing and first order methods: A unified framework. *SIAM Journal on Optimization*, 22(2):557–580, 2012.

[2] A. Boggess and F. J. Narcowich. *A first course in wavelets with Fourier analysis.* John Wiley & Sons, 2015.

[3] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A multigrid tutorial.* SIAM, 2000.

[4] H. Calandra, S. Gratton, E. Riccietti, and X. Vasseur. On high-order multilevel optimization strategies. *SIAM Journal on Optimization*, 31(1):307–330, 2021.

[5] J. Nocedal and S. Wright. *Numerical optimization.* Springer Science & Business Media, 2006.

[6] P. Parpas. A multilevel proximal gradient algorithm for a class of composite optimization problems. *SIAM Journal on Scientific Computing*, 39(5):S681–S701, 2017.

[7] N. Pustelnik, A. Benazza-Benhayia, Y. Zheng, and J.-C. Pesquet. Wavelet-based image deconvolution and reconstruction. *Wiley Encyclopedia of Electrical and Electronics Engineering*, 2016.

[8] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.

[9] G. Yuan. An introduction to wavelet theory and its applications in statistics (nonstationary, time series). 1998.