Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE

Corso di Laurea Magistrale in Matematica

L1-NORM BASED REGULARIZATION FOR A NON LINEAR IMAGING MODEL IN TOMOGRAPHY

Tesi di Laurea in Analisi Numerica

Relatore:
Chiar.ma Prof.
ELENA LOLI PICCOLOMINI

Correlatore:
Chiar.ma Prof.
GERMANA LANDI

Presentata da:
RUGGERO DE SANTIS

II Sessione
Anno Accademico 2017/2018

Alla mia famiglia

Introduction

Since the first X-ray picture was available in 1895 radiography played a key role in medical research. The development of new theories and the improved performances of computers allowed the introduction of techniques that were considered unbelievable just a few decades ago, like for instance computed tomography and magnetic resonance.

During the last years digital tomosynthesis became popular, a technique that allows the reconstruction of any section of a 3D object thanks to a certain number of 2D projections. The strength of this technique is based on the limited number of angles required in order to obtain the projection; on the contrary computed tomography requires a 360 degrees rotation. This characteristic makes tomosynthesis suitable for the more delicate areas of the body, like breast. On the one hand there is a minor quantity of radiations absorbed, on the other hand the examination can be carried out in more suitable positions of the patient in order to achieve good quality images.

The mathematical model for tomosynthesis was simplified until 2010. X-ray beam was considered to be monoenergetic and the object to be made up of one material. Thanks to these simplifications the issue result in a linear inverse problem. On the contrary in this work we will consider the polyenergetic and multimaterial model and as a consequence a non-linear inverse problem of great dimensions will be taken into account.

Like any other inverse problems, the issue related to the tomosynthesis is an ill-posed problem. These type of problems requires other information on the solution in order to stabilize the issue. In other words a *regularization* of the problem is essential, that is also the purpose of this work.

Two types of regularizations will be tested: the first is based on the L1-Norm of the solution whereas the second is based on the L1-Norm of the gradient of the solution.

The last one is usually called Total Variation.

The problem related to the tomosynthesis and in general the ill-posed problems are introduced in the first chapter of this work. In addition polyenergetic and multimaterial models, that characterize the problem, are also described in the second part of the chapter.

In the second chapter the least squares problem is depicted and applied to our case. Furthermore the concept of regularization is introduced and the problem is written in its final form, the one that we will solve; in the end the two types of chosen regularizations are explained.

In the third chapter we introduce the basis of mathematical optimization and the two strategies of line search and trust region. Moreover two methods used in order to solve the minimum problems are described: the *Gradient* method and the *Non-Linear Conjugate Gradient* method.

In the fourth chapter numerical results obtained are explained and commented firstly through the comparison of the two methods of a given regularization and finally through that of the two regularizations in general. Lastly we draw conclusions and suggest ideas for future works.

Introduzione

Sin dal 1895, data della prima immagine medica ottenuta mediante raggi X, la radiografia ha svolto un ruolo fondamentale nel campo medico. Lo sviluppo di nuove teorie e l'aumento delle capacità di calcolo da parte dei computer hanno permesso l'utilizzo di tecniche, come la tomografia computerizzata o la risonanza magnetica, che risultavano proibitive fino a qualche decennio fa.

Negl'ultimi anni ha acquisito particolare interesse la tecnica della tomosintesi digitale; una tecnica in grado di ricostruire un qualsiasi numero di sezioni di un oggetto tridimensionale partendo da un insieme di proiezioni 2D. La forza di questa tecnica risiede nel fatto che le proiezioni sono prese solo da un numero ridotto di angoli, al contrario della tomografia computerizzata che richiede un'intera rotazione di 360°. Questa proprietà rende la tomosintesi digitale particolarmente adatta per le zone più delicate del corpo, ad esempio il seno, sia per il minor numero di radiazioni assorbite, sia per la possibilità del paziente di poter effuttuare l'esame in posizioni adatte all'ottenimento di buone immagini.

Fino al 2010 il modello matematico alla base della tomosintesi veniva semplificato. Il fascio di raggi X veniva considerato monoenergetico e l'oggetto composto di un solo materiale. Attraverso queste semplificazioni il problema si poteva ricondurre ad un problema inverso lineare. In questa tesi, invece, considereremo il modello polienergetico e multimateriale, affrontando quindi un problema inverso non lineare di grandi dimensioni.

Come tutti i problemi inversi anche quello legato alla tomosintesi è mal posto. Questa tipologia di problemi necessità di ulteriori informazioni sulla soluzione in modo da stabilizzare il problema, e cioè deve essere *regolarizzato*.

Lo scopo di questa tesi è proprio quello di regolarizzare questo problema. Verrano testati due tipi di regolarizzazioni: una basata sulla norma L1 della soluzione e una basata sulla norma L1 del gradiente della soluzione, quest'ultima è solitamente chiamata Variazione Totale.

Nel primo capitolo di questa tesi ci sarà un'introduzione alla tomosintesi, al problema che ne deriva e più in generale ai problemi mal posti, infine verrà esposto il modello polienergetico e multimateriale che caratterizza il problema legato alla tomosintesi.

Nel secondo capitolo introduciamo il problema ai minimi quadrati e lo applichiamo al nostro caso. In seguito introduciamo il concetto di regolarizzazione, scriviamo il problema nella sua forma finale, quella che poi andremo a risolvere ed infine descriviamo i due tipi di regolarizzazione scelti.

Nel terzo capitolo introduciamo i fondamenti dell'ottimizzazione numerica e quindi le due strategie di ricerca in linea e trust region. Subito dopo presentiamo i due metodi utilizzati per la risoluzione del problema di minimo: il metodo del *Gradiente* e il metodo del *Gradiente Coniugato non Lineare*.

Nel quarto capitolo esponiamo e commentiamo i risultati numerici ottenuti, confrontando prima i due metodi per una determinata regolarizzazione e poi, più in generale, le due regolarizzazioni. Traiamo, infine, le nostre conclusioni proponendo, inoltre, qualche idea per lavori futuri.

Indice

Capitolo 1

Digital tomosynthesis

1.1 A brief history of the tomosynthesis

Since the first medical x-ray image in 1895, made by Röntgen, projection radiography played a foundamental role in medical field. However, the conventional x-ray system have a great limitation: only one two-dimensional projection image of a three-dimensional is avaiable from each scan. Specifically in breast imaging, a false negative diagnosis may be caused by breast cancer obscured by overlapping tissue, while superimposed normal tissues may appear to be a cancerous mass, resulting in a false positive diagnosis.

Tomosynthesis is a technique for inversely constructing slices of a 3D object from a set of 2D projection images. The idea of tomosynthesis was known since the 1930s, but, only in the late 1960s and early 1970s the researchers put these ideas into practice, mainly due to issues of practical implementation, like insufficient imaging detectors and inadequate computing technology.

Techniques, such us, computed tomography (CT) and magnetic resonance imaging (MRI), had more success. CT allows the 3D reconstruction of objects by obtaining a complete 360° rotation of projection data around the object. However, CT is particularly challenging for breast imaging, the patient must be in prone position during the scan and this positioning makes it difficult to effectively image the chest wall and axilla area.

The idea behind tomosynthesis is that multiple 2D image projections of the object can provides different information about the 3D object. The projections are taken at varying incident angles and from the limited set of 2D projections, reconstruction algorithms should be able to reconstruct any number of slice of the 3D object.

Until 2010, to semplify the problem, the x-ray source was assumed monoenergetic, that is, that all incident photons have the same energy level. This assumption led to a linear optimization problem, easier to resolve, but also, to the phenomenon called *beam hardening*: x-ray photons emitted from an x-ray tube have a continuous distribution of energies, and as the x-ray beam passes through any attenuating medium there is a preferential absorption of low-energy photons, resulting in an increase in the mean energy of the x-ray beam. Ignoring this energy dependence can lead to the so called beam hardening artifacts in the reconstructed image, such as, "halo" effect around high density object or "cupping" artifacts.

In this paper we consider the polyenergetic model proposed in 2010 by Chung, Nagy and Sechopoulos.

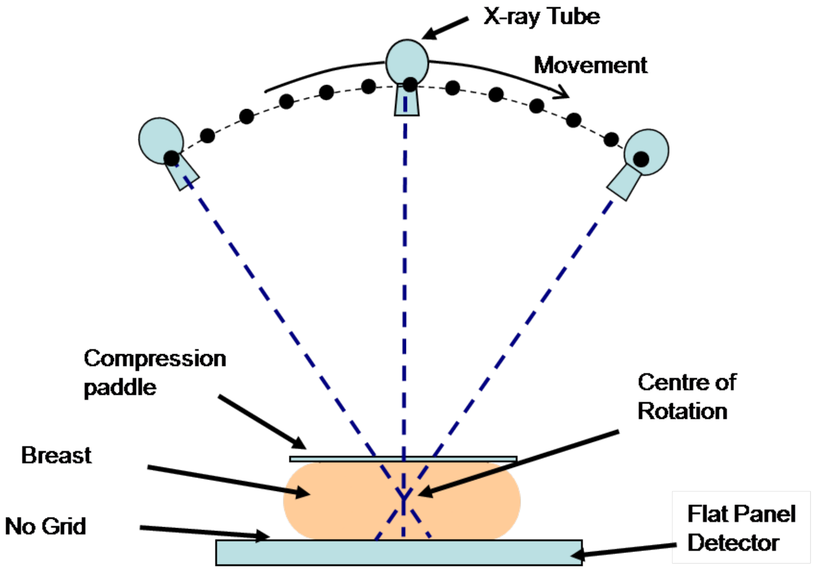


Figura 1.1: Sistem of breasts tomosyntesis

1.2 Ill-posed and inverse problems

The concept of *ill-posed* goes back to Hadamard at the beginning of the 20th century. Hadamard defined a problem well-posed if it satisfies:

- **Existence:** The problem must have a solution.
- **Uniqueness:** There must be one and only one solution to the problem.
- **Stability:** The solution must depend continuosly on the data.

If the problem violates one or more of these conditions, it is said to be ill-posed. For example:

$$x_1 + x_2 = 1 \qquad \text{(the world's simpliest ill-posed problem)}$$

has infinitely many solution. If we require the 2-norm of x, given by $||x||_2 = (x_1^2+x_2^2)^{\frac{1}{2}}$, is minimun, then the solution in unique $x_1 = x_2 = 0.5$.

Hadamard believed that ill-posed probelms wouldn't describe physical system. He was wrong, today ill-posed problems arise in the form of *inverse problems*. Inverse problems born naturally if one is interested, for example, in determining the unknown input that generate a measured output signal.

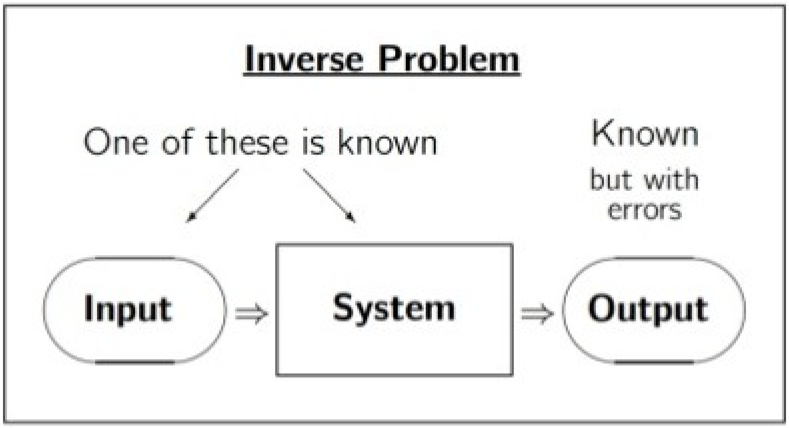


Figura 1.2: The inverse problem is to compute either the input or the system, given the other two quantities.

In this paper, we focus on digital tomosynthesis, where the "input" is an X-ray source, the "system" is the object being scanned, and the "output" is the measured damping of the X-rays. The problem can be formulated in the following general form:

$$b = K(X)s + \eta \tag{1.1}$$

where:

- b is the measure data's vector.
- K is a matrix that depend on the unkown X,and $K(X)$ depends on the specific application.
- η is the noise's vector.

1.3 The Mathematical Model

The model is based on Beer's law. Let's suppose that an X-ray, that cross an object, has intensity I_s. I_s decreases because it is partially absorbed by the object. The Beer-Lambert's law say that, if I_s is the initial intensity and I_f the outgoing one, these two measure are bound by the following relation:

$$I_f = I_s \cdot \mathrm{e}^{(-\int u(s)ds)}.$$

Now let's consider the polyenergetic case: let $b_i^{(\theta)}$ be the intesity measured at the i-th pixel of a digital x-ray detector the previously relation became:

$$b_i^{(\theta)} = \int_{\varepsilon} s(\varepsilon)\mathrm{e}^{-\int_{L_\theta} \mu(\vec{x},\varepsilon)d\ell} d\varepsilon + \eta_i^{(\theta)} \qquad i = 1, 2, \cdots, N_p \qquad \mathit{and} \qquad \theta = 1, 2, \cdots, N_\theta. \tag{1.2}$$

Where:

- N_p is the number of pixels (typically a few million) in the digital x-ray detector.
- N_θ is the number of projection images obtained when the x-ray source is moved to a new position, which is defined by an angle θ. In a typical tomosynthesis system $15 \leq N_\theta \leq 30$.

- ε represents the spectrum of energies that are emitted by the source x-ray beam, which can, for example, range from $10keV$ to $28keV$.
- $s(\varepsilon)$ is the energy fluence, which is a product of the x-ray energy with the number of incident photons at that energy.
- L_θ is the line on which the x-ray beam travels through the object.
- $\mu(\vec{x}, \varepsilon)$ is the linear attenuation coefficient, which depends on the energy of the x-ray beam, and on the material in the object at the position $\vec{x}$; lower energy will be attenuated more than higher energy, and denser materials will attenuate more than soft material.
- $\eta_i^{(\theta)}$ rapresents additional contributions (noise) measured at the detector, which can include x-ray scatter and electronic noise.

We have to discretize the previous equation, and that lead to the discrete model:

$$b_i^{(\theta)} = \sum_{\varepsilon=1}^{N_\varepsilon} s_\varepsilon \text{exp}\left(-\sum_{\ell=1}^{N_v} a_{i,\ell}^{(\theta)} \mu_{\ell,\varepsilon} \right) + \eta_i^{(\theta)}, \qquad i = 1, 2, \cdots, N_p \quad and \quad \theta = 1, 2, \cdots, N_\theta. \tag{1.3}$$

Where:

- N_v is the number of voxel (typically a few billion) in the discretized 3D object.
- N_ε is the number of discrete energy level. Becuase N_v is extremely large, in general $N_\varepsilon \ll N_v$.
- $a_{i,\ell}^{(\theta)}$ is the lenght of the x-ray that passes through voxel ℓ, contributing to pixel i.

Now we want to compact the equation 1.3. We define a matrix $\mathcal{A}^{(\theta)}$ with entries $a_{i,\ell}^{(\theta)}$ and a matrix $\mathcal{M}$ with entries $\mu_{\ell,\varepsilon}$. Now we can write the equation in matrix-vector form:

$$\mathbf{b}^{(\theta)} = \exp(-\mathcal{A}^{(\theta)} \mathcal{M})\mathbf{s} + \eta^{(\theta)}, \quad \theta = 1, 2, \cdots, N_\theta. \tag{1.4}$$

Where the exponentiation operation is done element-wise on the matrix $-\mathcal{A}^{(\theta)}\mathcal{M}$. A discrete model that comprises all projections can be written as:

$$\mathbf{b} = \exp(-\mathcal{A}\mathcal{M})\mathbf{s} + \eta, \quad \theta = 1, 2, \cdots, N_\theta. \tag{1.5}$$

where

$$\mathbf{b} = \begin{bmatrix} b^{(1)} \\ b^{(2)} \\ \vdots \\ b^{(N_\theta)} \end{bmatrix} \quad \text{and} \quad \mathcal{A} = \begin{bmatrix} A^{(1)} \\ A^{(2)} \\ \vdots \\ A^{(N_\theta)} \end{bmatrix}$$

An accurate estimate of the x-ray energy distribution can be obtained using well-known x-ray spectra models, and calibration measurements can be obtained by taking x-ray transmission measurements of objects (e.g. high-purity aluminum) that have known dimension, density and material composition. Information about the additive noise term, η, can be also estimated through preprocessing or calibration steps. Using this x-ray spectra modeling, the image reconstruction problem assumes $\mathbf{b}$, $\mathcal{A}$ and $\mathbf{s}$ are known, and we have to solve 1.5 for $\mathcal{M}$, or we can solve the general inverse problem 1.1 setting:

$$\mathbf{K}(\mathbf{X}) \equiv \exp(-\mathcal{A}\mathbf{X}), \quad \text{with} \quad \mathbf{X} \equiv \mathcal{M}$$

The problem is non linear due to the exponential in the equation. It is not easy to computationally solve this problem. Tipically are used simplifying assumption to get an approximate linear model. For example, if we consider a monoenergetic model, i.e, $N_\varepsilon = 1$, $\mathcal{M}$ is a vector and $\mathbf{s}$ is a scalar. With this the equation 1.5 became a linear inverse problem:

$$\widehat{b} = \mathcal{A}\mathbf{X} + \eta$$

and the entries of $\widehat{b}$ are:

$$\widehat{b}_i = -\log\left(\frac{b_i}{s}\right)$$

1.3.1 Multimaterial Model

In this section we introduce the general framework for material decomposition design by Nagy, Feng and Sechopoulos. Under the assumption that the densities of different components are similar, the linear attenuation coefficients $\mu_{\ell,\varepsilon}$, of the composite material making the object (e.g., the breast) can be approximated as a linear combination of individual materials. Suppose that there are N_m distinct materials making up the object, we have:

$$\mu_{\ell,\varepsilon} \approx \sum_{m=1}^{N_m} w_{\ell,m} c_{m,\varepsilon}, \tag{1.6}$$

where:

- $c_{m,\varepsilon}$ are known linear attenuation coefficients for the m-th material in voxel ℓ at x-ray energy ε.
- $w_{\ell,m}$ are unknown weight fractions (or percentages) of the m-th material in the ℓ-th voxel of the object.

We also assume:

$$\sum_{m=1}^{N_m} w_{\ell,m} = 1, \quad \ell = 1, 2, \cdots, N_v$$

or:

$$w_{\ell,1} = 1 - \sum_{m=2}^{N_m} w_{\ell,m} \tag{1.7}$$

Now replacing 1.5 in 1.3 we obtain:

$$b_i^{(\theta)} = \sum_{\varepsilon=1}^{N_\varepsilon} s_\varepsilon \exp\left(- \sum_{\ell=1}^{N_v} a_{i,\ell}^{(\theta)} \sum_{m=1}^{N_m} w_{\ell,m} c_{m,\varepsilon} \right) + \eta_i^{(\theta)} \quad i = 1, \cdots, N_p; \quad \theta = 1, \cdots, N_\theta. \tag{1.8}$$

$\eta_i^{(\theta)}$ rapresent noise measured at the detector, which can include x-ray scatter and electronic noise. Normally it follow a Poisson distribution, but, with opportune hypothesis, we can replace the Poisson distribution with a Gaussian distribution.

Now setting $W = [w_{\ell,m}]$ e $C = [c_{\varepsilon,m}]$ the previous equation become:

$$\mathbf{b} = \exp(-\mathcal{A}\mathbf{W}\mathbf{C}^{\mathbf{T}})\mathbf{s} + \eta, \tag{1.9}$$

where:

- W is a $N_v \times N_m$ matrix and his elements $w_{\ell,m}$ are the unknown weights of the m-th material in the ℓ-th voxel.
- C is a $N_\varepsilon \times N_m$ matrix and his elements are the known linear attenuation coefficients of the m-th matreial.

Capitolo 2

The Non Linear Reconstruction Model

2.1 Non Linear Least-Square Problems

In least-squares problems, the objective function ϕ has the following form:

$$\phi(x) = \frac{1}{2}\sum_{j=1}^{m} r_j^2(x), \tag{2.1}$$

where each r_j is a smooth function from $\mathbb{R}^n$ to $\mathbb{R}$. We refer to each r_j as a *residual.* If the residuals r_j are affine the problems are linear least-squares problems otherwise are non linear least-square problems.

Now we assemble the individual components r_j into a *residual vector* $r : \mathbb{R}^n \to \mathbb{R}^m$, as follow:

$$r(x) = (r_1(x), r_2(x), \cdots, r_m(x))^T.$$

Using this notation, we can rewrite ϕ as $\phi(x) = \frac{1}{2}||r(x)||_2^2 = \frac{1}{2}r(x)^T r(x)$. In this way the derivates of $\phi(x)$ can be expressed in terms of the *Jacobian* $J(x)$:

$$J(x) = \left[\frac{\partial r_j}{\partial x_i}\right]_{j=1,2,\ldots,m;\, i=1,2,\ldots,n},$$

and can be express using the gradients:

$$\nabla r_i = \left[\frac{\partial r_i}{\partial x_1}, \frac{\partial r_i}{\partial x_2}, \cdots, \frac{\partial r_i}{\partial x_n}\right]^T$$

as:

$$J(x) = [\nabla r_1(x)^T, \nabla r_2(x)^T, \cdots, \nabla r_m(x)^T]^T.$$

Now, getting back to our problem, under a Gaussian noise assumption, the solution to 1.8 is obtained by solving the non linear least squares problem:

$$\min_{\mathbf{W}} \frac{1}{2}||\mathbf{b} - \exp(-\mathcal{A}\mathbf{W}\mathbf{C}^{\mathbf{T}})\mathbf{s}||^2 \tag{2.2}$$

where η was absorbed inside the vector $\mathbf{b}$. By imposing 1.7 on 2.2 we obtain the non linear least square problem:

$$\min_{\mathbf{X}} f^*(\mathbf{X}) = \min_{\mathbf{X}} \left(\frac{1}{2}||\mathbf{r}(\mathbf{X})||^2\right) \tag{2.3}$$

where the unknown $\mathbf{X}$ is defined as:

$$\mathbf{X} = [\mathbf{w_2}|\mathbf{w_3}|\cdots|\mathbf{w_{N_m}}]$$

and $\mathbf{w_i}$ is the ith column of $\mathbf{W}$. If $\mathbf{x_m}$ denotes the mth column of $\mathbf{X}$, then the residual $\mathbf{r}(\mathbf{X})$ has the form:

$$\mathbf{r}(\mathbf{X}) = \mathbf{b} - \exp\left(-\mathcal{A}\left[1 - \sum_{m=1}^{N_m-1} \mathbf{x_m}\middle|\mathbf{X}\right]\mathbf{C}^{\mathbf{T}}\right)\mathbf{s}. \tag{2.4}$$

If we define the $N_\varepsilon \times (N_m - 1)$ matrix as:

$$\hat{\mathbf{C}} = [\mathbf{c_2} - \mathbf{c_1}|\mathbf{c_3} - \mathbf{c_1}|\cdots|\mathbf{c_{N_m}} - \mathbf{c_1}]$$

where $\mathbf{c}_\ell$ denotes the ℓth column of $\mathbf{C}$, then equation 2.4 can be written component wise as:

$$r_i^\theta = b_i^\theta - \sum_{\varepsilon=1}^{N_\varepsilon} s_\varepsilon \exp\left(-\sum_{\ell=1}^{N_v} a_{i,\ell}^\theta \left(c_{1,\varepsilon} + \sum_{m=1}^{N_m-1} x_{\ell,m}\hat{c}_{m,\varepsilon}\right)\right).$$

2.1.1 Computing the Gradient

Due to the high dimensionality of tomosynthesis imaging problems, computing the gradient of $f^*(\mathbf{X})$ is a crucial issue for the implementation of any numerical method for the solution of 2.3. The gradient of $f^*(\mathbf{X})$ is expressed in terms of the Jacobian $\mathbf{J}(\mathbf{X})$ of $\mathbf{r}(\mathbf{X})$:

$$\nabla f^*(\mathbf{X}) = \mathbf{J}(\mathbf{X})^{\mathbf{T}}\mathbf{r}(\mathbf{X})$$

where $\mathbf{J}(\mathbf{X})$ is the $N_pN_\theta \times N_vN_m$ matrix defined by:

$$\begin{aligned}
\{\mathbf{J}(\mathbf{X})\}_{i,\theta,j,m} &= \frac{\partial}{\partial x_{j,m}} r_i^\theta \\
&= \frac{\partial}{\partial x_{j,m}}\left(b_i^\theta - \sum_{\varepsilon=1}^{N_\varepsilon} s_\varepsilon \text{exp}\left(-\sum_{\ell=1}^{N_v} a_{i,\ell}^\theta\left(c_{1,\varepsilon} + \sum_{m=1}^{N_m-1} x_{\ell,m}\hat{c}_{m,\varepsilon}\right)\right)\right) \\
&= -\sum_{\varepsilon=1}^{N_\varepsilon} s_\varepsilon \text{exp}\left(-\sum_{\ell=1}^{N_v} a_{i,\ell}^\theta\left(c_{1,\varepsilon} + \sum_{m=1}^{N_m-1} x_{\ell,m}\hat{c}_{m,\varepsilon}\right)\right)(-a_{i,j}^\theta \hat{c}_{m,\varepsilon}) \qquad (2.5)\\
&= a_{i,j}^\theta \sum_{\varepsilon=1}^{N_\varepsilon} \text{exp}\left(-\sum_{\ell=1}^{N_v} a_{i,\ell}^\theta\left(c_{1,\varepsilon} + \sum_{m=1}^{N_m-1} x_{\ell,m}\hat{c}_{m,\varepsilon}\right)\right) s_\varepsilon \hat{c}_{m,\varepsilon} \\
&= a_{i,j}^\theta \sum_{\varepsilon=1}^{N_\varepsilon} \left\{\text{exp}\left(-\mathcal{A}\left[1 - \sum_{m=1}^{N_m-1} \mathbf{x_m}\middle|\mathbf{X}\right]\mathbf{C^T}\right)\right\}_{i,\theta,\varepsilon} (\mathbf{s} \odot \hat{\mathbf{c}}_\mathbf{m})_\varepsilon
\end{aligned}$$

for $i = \cdots, N_p \quad \theta = 1, \cdots, N_\theta, \quad j = 1, \cdots, N_v, \quad m = 1, \cdots, N_m$ and $\odot$ denotes component wise multiplication.

Now let's see how we regularize the least-square problem 2.3.

2.2 Regularization

We already see, in chapter one, that problem 2.3 is an ill-posed problem. The primary difficulty with ill-posed problems is that they are practically underdetermined due to the cluster of small singular values of K. Hence, it is necessary to incorporate further information about the desired solution in order to stabilize the problem and to single out a useful and stable solution. This is the purpose of *regularization.*

Recall the problem 2.3:

$$\min_{\mathbf{X}} f^*(\mathbf{X}) = \min_{\mathbf{X}} \frac{1}{2}||\mathbf{r}(\mathbf{X})||^2,$$

the dominating approach to regularization is using one of the following four schemes.

1. Minimize $f^*(\mathbf{X})$ subject to the constraint that $\mathbf{X}$ belongs to a specified subset.

2. Minimize $f^*(\mathbf{X})$ subject to the constraint that a measure of $\omega(\mathbf{X})$ of the "size" of $\mathbf{X}$ is less than some specified upper bound δ.

3. Minimize $\omega(\mathbf{X})$ subject to the constraint $f^*(\mathbf{X}) \leq \alpha$.

4. Minimize a linear combination of $f^*(\mathbf{X})$ and $\omega(\mathbf{X})$:

   $$\min\{f^*(\mathbf{X}) + \lambda\omega(\mathbf{X})\}, \tag{2.6}$$

   where λ is a specified weighting factor.

Here, α, δ and λ are known as reguarization parameters, and the function ω is sometimes referred to as the *smoothing norm.* The underlying idea in all four schemes is that a regularized solution having a suitably small residual norm and satisfying the additional constraint will be not too far from the desired unknown solution.

In this paper we'll use the fourth scheme with:

$$\omega(\mathbf{X}) = ||\Phi(\mathbf{X})||_{1,\beta},$$

where

$$||\Phi(\mathbf{X})||_{1,\beta} = \sum_{i=1}^{N_v}(|\Phi(x_i)|^2 + \beta^2)^{\frac{1}{2}} \quad \beta > 0, \tag{2.7}$$

and

$$\Phi(\mathbf{X}) = \mathbf{X} \quad \text{or} \quad \Phi(\mathbf{X}) = \nabla\mathbf{X}.$$

The second choice leads to the *Total Variation regularization.* We have to use the approximation $||\cdot||_{1,\beta}$ of the $1-$norm due to the nondifferentiability of this norm at the origin.

At last the optimization problem we'll resolve, will be:

$$\min_{\mathbf{X}} f(\mathbf{X}) = \min_{\mathbf{X}}\left(\frac{1}{2}||\mathbf{r}(\mathbf{X})||^2 + \lambda||\Phi(\mathbf{X})||_{1,\beta}\right). \tag{2.8}$$

2.2.1 Total Variation Regularization

As we said the choice of $\Phi(\mathbf{X}) = \nabla\mathbf{X}$ lead to Total Variation regularization. Now we'll study it in more detail.

Let ϕ be a smooth function on the interval $[0, 1]$, we can define the total variation of ϕ as:

$$TV(\phi) = \int_0^1 \left|\frac{d\phi}{dx}\right| dx.$$

A generalization to two and three space dimension is:

$$TV(\phi) = \int_0^1 \int_0^1 ||\nabla\phi||_2 dxdy,$$

$$TV(\phi) = \int_0^1 \int_0^1 \int_0^1 ||\nabla\phi||_2 dxdydz.$$

An extension of this representation, valid even when ϕ is not smooth, is:

$$TV(\phi) = \sup_{v\in\mathcal{V}} \int_0^1 \int_0^1 \int_0^1 \phi(x,y,z)\,\mathrm{div}\,v\,dxdydz, \tag{2.9}$$

where $\mathcal{V}$ consist of vector-valued function $v = (v_1(x,y,z), v_2(x,y,z), v_3(x,y,z))$ whose Euclidean norm is bounded by 1 and whose components v_i are continuously differentiable and vanish on the boundary of the unit square. $\mathrm{div}\,v = \frac{\partial v_1}{\partial x} + \frac{\partial v_2}{\partial y} + \frac{\partial v_3}{\partial z}$ is the divergence of v. We will take 2.9 as definition of Total Variation. Using the fourth scheme of regularization (2.6) with the Total Variation functional as $\omega(\cdot)$, the optimization problem can be write as:

$$T_\lambda(\mathbf{X}) = \frac{1}{2}||\mathbf{r}(\mathbf{X})||^2 + \lambda TV(\mathbf{X}).$$

To overcome the non-differentiability of the norm at the origin, we take the approximation $\sqrt{|x|^2 + \beta^2}$. This yields the following approximation to $TV(\phi)$, valid for a smooth function ϕ defined on the unit interval in one dimension:

$$J_\beta(\phi) = \int_0^1 \sqrt{\left(\frac{d\phi}{dx}\right)^2 + \beta^2 dx}.$$

In two and three space dimension, becomes:

$$J_\beta(\phi) = \int_0^1 \int_0^1 \sqrt{\left(\frac{\partial\phi}{\partial x}\right)^2 + \left(\frac{\partial\phi}{\partial y}\right)^2 + \beta^2 dxdy}.$$

$$J_\beta(\phi) = \int_0^1 \int_0^1 \int_0^1 \sqrt{\left(\frac{\partial\phi}{\partial x}\right)^2 + \left(\frac{\partial\phi}{\partial y}\right)^2 + \left(\frac{\partial\phi}{\partial z}\right)^2 + \beta^2 dxdydz}. \tag{2.10}$$

Then $T_\lambda(\mathbf{X})$ becomes:

$$T_\lambda(\mathbf{X}) = \frac{1}{2}||\mathbf{r}(\mathbf{X})||^2 + \lambda J(\mathbf{X}), \tag{2.11}$$

where J is a discretization of $J_\beta(\mathbf{X})$ and it's often call the *penalty functional*

Discretization in One Space Dimension

Suppose $\phi(x)$ is a smooth function defined on the unit interval in $\mathbb{R}$ and $\vec{\phi} = (\phi_0, \dots, \phi_n)$ with $\phi_i = \phi(x_i)$, $x_i = i\Delta x$, $\Delta x = 1/n$. Next let:

$$D_i\vec{\phi} = \frac{\phi_i - \phi_{i-1}}{\Delta x}, \quad i = 1, \dots, n$$

be the derivate approximation.

Then the penalty functional J becomes:

$$J(\vec{\phi}) = \frac{1}{2}\sum_{i=1}^{n} \psi((D_i\phi)^2)\Delta x, \tag{2.12}$$

where ψ is a smooth approximation to twice the square root function with the property:

$$\psi'(t) > 0 \quad \text{whenever} \quad t > 0.$$

To simplify notation, we'll omit the factor Δx. This factor can be absorbed in the regularization parameter λ.

In this paper and in the numerical implementation we'll use:

$$\psi(t) = 2\sqrt{t + \beta^2}.$$

Note that with this choice we obtain:

$$J(\mathbf{X}) = \sum_{i=1}^{n} \sqrt{(D_i\mathbf{X})^2 + \beta^2} = ||\Phi(\mathbf{X})||_{1,\beta},$$

when $\Phi(\mathbf{X}) = \nabla\mathbf{X}$

We need also the gradient of J. For any $\mathbf{v} \in \mathbb{R}^{n+1}$,

$$\begin{aligned} \frac{d}{d\tau}J(\vec{\phi} + \tau\mathbf{v}) &= \sum_{i=1}^{n} \psi'((D_i\vec{\phi})^2)(D_i\vec{\phi})(D_i\mathbf{v}) \\ &= (D\mathbf{v})^T\mathrm{diag}(\psi'(\vec{\phi}))(D\vec{\phi}) \\ &= \langle D^T\mathrm{diag}(\psi'(\vec{\phi}))D\vec{\phi}, \mathbf{v}\rangle, \end{aligned} \tag{2.13}$$

where $\text{diag}(\psi'(\vec{\phi}))$ denote the $n\times n$ diagonal matrix whose ith diagonal entry is $\psi'((D_i\vec{\phi})^2)$, D is the $n \times (n+1)$ matrix whose ith row is D_i, and $\langle\cdot,\cdot\rangle$ denotes the Euclidean inner product on $\mathbb{R}^{n+1}$. Then the gradient will be:

$$\nabla J(\vec{\phi}) = D^T \text{diag}(\psi'(\vec{\phi}))D\vec{\phi} = L(\vec{\phi})\vec{\phi},$$

where $L(\vec{\phi})$ is a positive semidefinite and symmetric $(n+1)\times(n+1)$ matrix.

Discretization in Two Space Dimension

Suppose $\phi = \phi_{ij}$ is defined on an equispaced grid in two space dimension, $\{(x_i, y_j)\,|\, x_i = i\Delta x,\, y_j = j\Delta y, \quad i = 0,\dots,n_x,\, j = 0,\dots,n_y\}$. The discrete penalty functional $J : \mathbb{R}^{(n_x+1)\times(n_y+1)} \to \mathbb{R}$ become:

$$J(\phi) = \frac{1}{2}\sum_{i=1}^{n_x}\sum_{j=1}^{n_y}\psi\left[(D^x_{ij}\phi)^2 + (D^y_{ij}\phi)^2\right],$$

where

$$D^x_{ij}\phi = \frac{\phi_{i,j} - \phi_{i-1,j}}{\Delta x}, \quad D^y_{ij}f = \frac{\phi_{i,j} - \phi_{i,j-1}}{\Delta y}.$$

Gradient computations are similar to those in one dimension:

$$\frac{d}{d\tau}J(\phi+\tau v)|_{\tau=0} = \sum_{i=1}^{n_x}\sum_{j=1}^{n_y}\psi'_{ij}\left[(D^x_{ij}f)(D^x_{ij}v) + (D^y_{ij}\phi)(D^y_{ij}v)\right]$$

where $\psi'_{ij} = \psi'\left[(D^x_{ij}\phi)^2 + (D^y_{ij}\phi)^2\right]$.

Now let $\vec{\phi} = \mathbf{vec}(\phi)$ and $\mathbf{v} = \mathbf{vec}(v)$ corresponding to lexicographical column ordering of the two-dimensional array components, i.e given an array $v \in \mathbb{C}^{n_x\times n_y}$ we can obtain $\mathbf{v} \in \mathbb{C}^{n_x n_y}$ in this way:

$$\mathbf{v} = \mathbf{vec}(v) = [v_{1,1},\dots,v_{n_x,1},v_{1,2},\dots,v_{n_x,2},\dots,v_{1,n_y},\dots,v_{n_x,n_y}];$$

let D_x and D_y denote the resulting $n_x n_y \times (n_x+1)(n_y+1)$ matrices corresponding to the grid operator $D^x_{ij}\phi = \frac{\phi_{i,j}-\phi_{i-1,j}}{\Delta x}$, and $D^y_{ij}\phi = \frac{\phi_{i,j}-\phi_{i,j-1}}{\Delta y}$; let $\text{diag}(\psi'(\vec{\phi}))$ denote the $n_x n_y \times n_x n_y$ diagonal matrix whose diagonal entries are the ψ'_{ij}s; and let $\langle\cdot,\cdot\rangle$ denote the Euclidean inner product on $\mathbb{R}^{(n_x+1)(n_y+1)}$. Then:

$$\frac{d}{d\tau}J(\phi+\tau v)|_{\tau=0} = \langle \text{diag}(\psi'(\vec{\phi}))D_x\vec{\phi}, D_x\mathbf{v}\rangle + \langle \text{diag}(\psi'(\vec{\phi}))D_y\vec{\phi}, D_y\mathbf{v}\rangle.$$

From this we obtain a gradient rapresentation:

$$\nabla J(\vec{\phi}) = L(\vec{\phi})\vec{\phi},$$

where:

$$L(\vec{\phi}) = D_x^T \mathrm{diag}(\psi'(\vec{\phi}))D_x + D_y^T \mathrm{diag}(\psi'(\vec{\phi}))D_y$$

2.2.2 $1-$Norm Regularization

The second choice we made is $\Phi(\mathbf{X}) = \mathbf{X}$. Now let $\tilde{\mathbf{X}} = \mathbf{vec}(\mathbf{X})$ be the lexicographical column ordering of the three-dimensional array components, the optimization problem become:

$$f(\tilde{\mathbf{X}}) = \frac{1}{2}||\mathbf{r}(\tilde{\mathbf{X}})||^2 + \lambda J_\beta(\tilde{\mathbf{X}}),$$

where

$$J_\beta(\tilde{\mathbf{X}}) = ||\tilde{\mathbf{X}}||_{1,\beta} = \sum_{i=1}^{N_v} \sqrt{(|x_i|^2 + \beta^2)}.$$

We'll, also, need the gradient $\nabla J_\beta(\tilde{\mathbf{X}})$, and we can obtain it with a direct calculation:

$$\nabla J_\beta(\tilde{\mathbf{X}})_i = \frac{\partial}{\partial x_i} J_\beta(\tilde{\mathbf{X}}) = \frac{x_i}{\sqrt{|x_i|^2 + \beta^2}}; \quad i = 1, \ldots, N_v.$$

Capitolo 3

Optimization Alghoritms

Mathematically speaking, optimization is the minimization or maximization of a function subject to constraints on its variables. The optimization problem can be written as follow:

$$\min_{x\in\mathbb{R}^n} \phi(x) \quad \text{subject to} \quad c_i(x) = 0, \quad i \in \mathcal{I}$$

in this paper we try to resolve the problem defined in 2.3, or should i say, the regularization of that problem.

For this purpose, in this chapter, we'll summarise the basics of numerical optimization and, furthermore, we'll study the optimization alghoritms of the Gradient and of the Conjugate Gradient.

3.1 Basics of Numerical Optimization

All algorithms for unconstrained minimization require the user to supply a starting point, which we usually denote by x_0. Beginning at x_0, optimization algorithms generate a sequence of iterates $\{x_k\}_{k=0}^{\infty}$ that terminate when either no more progress can be made or when it seems that a solution point has been approximated with sufficient accuracy. In deciding how to move from one iterate x_k to the next, the algorithms use information about the function ϕ at x_k , and possibly also information from earlier iterates $x_0, x_1, \cdots, x_{k-1}$. They use this information to find a new iterate x_{k+1} with a lower function value than x_k. There are two fundamental strategies for moving from the current point x_k to a new iterate x_{k+1}: *line search* and *trust region* methods.

In the *line search* strategy, the algorithm chooses a direction p_k and searches along this direction from the current iterate x_k for a new iterate with a lower function value. The distance to move along p_k can be found by approximately solving the following one-dimensional minimization problem to find a step length α_k:

$$\min_{\alpha>0} \phi(x_k + \alpha p_k). \tag{3.1}$$

The search direction has often the form:

$$p_k = -B_k^{-1} \nabla \phi_k,$$

where B_k is a symmetric and non singular matrix. In the steepest descent method, $B_k = I$ where I is the identity matrix, in Newton's method, B_k is the exact Hessian and in quasi-Newton method is an approximation to the Hessian. When p_k is defined as above and B_k is positive definite, we have:

$$p_k^T \nabla \phi_k = -\nabla \phi_k^T B_k^{-1} \nabla \phi_k < 0,$$

and therefore p_k is a descent direction.

the ideal choice of the step lenght α_k would be the global minimizer of 3.1, but in general, it is too expensive to identify this value. More practical strategies perform *inexact* line search to identify a step length that achieves a sufficient reduction in ϕ. Typically line search algorithms try out a sequence of possible values of α and accept the one that satisfied certain conditions.

A simple condition we could impose on α_k is to require a reduction in ϕ , i.e., $\phi(x_k + \alpha_k p_k) < \phi(x_k)$. This requirement in not enough to produce convergence. A popular inexact line search condition is the so called *Armijo condition*:

$$\phi(x_k + \alpha p_k) \leq \phi(x_k) + c_1 \alpha \nabla \phi_k^T p_k, \tag{3.2}$$

for some constant $c_1 \in (0,1)$. This condition assure a sufficient decrease in the objective function ϕ.

The Armijo condition is not enough by itself because it is satisfied for all sufficiently small values of α. To avoid unacceptably short steps we required the *curvature condition*, which requires α_k to satisfy:

$$\nabla \phi(x_k + \alpha_k p_k)^T p_k \leq c_2 \nabla \phi_k^T p_k, \tag{3.3}$$

for some constant $c_2 \in (c_1, 1)$.

The Armijo and curvature condition collectively are know as the *Wolfe conditions*:

$$\begin{aligned} &\phi(x_k + \alpha p_k) \leq \phi(x_k) + c_1 \alpha \nabla\phi_k^T p_k, \\ &\nabla\phi(x_k + \alpha_k p_k)^T p_k \leq c_2 \nabla\phi_k^T p_k, \end{aligned} \tag{3.4}$$

with $0 < c_1 < c_2 < 1$.

Thank to Zoutendijk theorem the choice of α that verify the Wolfe condition guarantee the convergence of line search method. The Wolfe condition are expansive to verify and, usually, the line search algorithms chooses its candidate step lengths using the *backtracking* approach:

Armijo with Backtracking Algorithm:

```
Choose $\bar{\alpha} > 0$, $\rho \in (0,1)$, $c \in (0,1)$; Set $\alpha \leftarrow \bar{\alpha}$;
repeat until $\phi(x_k + \alpha p_k) \leq \phi(x_k) + c\alpha\nabla\phi_k^T p_k$
    $\alpha \leftarrow \rho\alpha$;
end(repeat)
Terminate with $\alpha_k = \alpha$.
```

In the *trust region* strategy, we use the information about the objective function ϕ to built a *model function* m_k whose beavior, near the current iterate x_k, is similar to that of ϕ. The model m_k may not be a good approximation of ϕ far from x_k, so we restrict the search of a minimizer to some region around x_k. In other word, we search the step p by approximately solving:

$$\min_p m_k(x_k + p), \quad \text{where} \quad x_k + p \quad \text{lies inside the trust region.} \tag{3.5}$$

If the candidate solution does not produce a significant decrease in ϕ the trust region is too large, so we shrink it and re-solve 3.1. Usually, the trust region is a ball defined by $||p||_2 \leq \Delta$ where $\Delta > 0$ is called trust region radius and the model m_k is defined to be a quadratic function of the form:

$$m_k(x_k + p) = \phi_k + p^T \nabla\phi_k + \frac{1}{2} p^T B_k p,$$

where ϕ_k and $\nabla\phi_k$ are the function and the gradient of the function calculated in x_k, while, B_k is the Hessian of ϕ or some approximation of it.

The general strategy of trust region method is:

Trust region strategy:
at the iterate k:

- we define the model m_k;
- we define the trust region, choosing the trust region radius Δ;
- we find the solution p_k of the problem:

  $$\min_p m_k(p),$$

  with the constraint $||p|| \leq \Delta$;
- if p_k produce a significant decrease of $\phi(x_k)$, we set $x_{k+1} = x_k + p_k$ and Δ could be increased or kept constant. Otherwise, if p_k is unaccettable, we reduce Δ and resolve again the previous problem.

3.2 Gradient Method

Due to the possible huge size of the problem, first-order algorithms exploiting only the gradient of the ϕ are very appealing approaches. Then, the Steepest Descend or Gradient method could be a good choice. The search direction for the Gradient method is:

$$p_k = -\nabla \phi_k$$

and using an inexact line search strategies a general scheme for the Gradient method will be:

Algorithm 3.2.1 (Gradient method)**.**

```
Set $x_0 \in \mathbb{R}^n$, $\beta$, $\sigma \in (0,1)$ and $\alpha$.
for $k = 0, 1, \ldots$
    $d_k = -\alpha * \nabla\phi$;
    $\eta_k = 1$;
    WHILE $\big(\phi(x_k + \eta_k d_k) > \phi(x_k) + \sigma\eta_k \nabla\phi(x_k)^T d_K\big)$
```

```
        $\eta_k = \beta\eta_k$;    (backtracking step)
    END
    $x_{k+1} = x_k + \eta_k d_k$;
    END
```

3.3 Coniugate Gradient Methods

We chose the Coniugate Gradient methods for two reason: first, they are among the most useful techniques for solving large linear systems of equations, and second, they can be adapted to solve nonlinear optimization problems.

The Linear Conjugate Gradient method was proposed by Hestenes and Stiefel in the 1950s as an iterative method for solving linear systems with positive definite coefficient matrices.

The first Nonlinear Conjugate Gradient method was introduced by Fletcher and Reeves in the 1960s. It is one of the earliest known techniques for solving large-scale nonlinear optimization problems.

3.3.1 The Linear Conjugate Gradient Method

The Conjugate Gradient method is an iterative method for solving a linear system of equations:

$$Ax = b,$$

where A is an $n \times n$ symmetric positive definite matrix. The problem can be stated equivalently as the following minimization problem:

$$\min \varphi(x) = \frac{1}{2}x^T Ax - b^T x.$$

Indeed we have:

$$\nabla\varphi(x) = Ax - b$$

This equivalence will allow us to interpret the Conjugate Gradient method either as an algorithm for solving linear systems or as a technique for minimizing convex quadratic functions.

Definition 3.3.1 (A-conjugate Direction)**.** Let A be a $n \times n$ symmetric positive definite matrix. u and v are said to be conjugate with respect to A or A-conjugate if:

$$u^T A v = 0$$

We consider the following conjugate direction method. Given a startinh point $x_0 \in \mathbb{R}$ and a set of A-conjugate direction $\{p_0, p_1, \ldots, p_{n-1}\}$, let us generate the sequence $\{x_k\}$:

$$x_{k+1} = x_k + \alpha_k p_k,$$

where:

$$\alpha_k = \arg\min_{\alpha \geq 0} \varphi(x_k + \alpha p_k).$$

Let impose:

$$\frac{d}{d\alpha} \varphi(x_k + \alpha p_k)|_{\alpha = \alpha_k} = 0,$$

and we obtain:

$$\begin{aligned} 0 &= p_k^T (A(x_k + \alpha_k p_k) - b) \\ &= p_k^T A x_k + \alpha_k p_k^T A p_k - p_k^T c \\ &= \alpha_k p_k^T A p_k + p_k^T \nabla \varphi(x_k). \end{aligned} \tag{3.6}$$

and therefore:

$$\alpha_k = -\frac{\nabla \varphi^T(x_k) p_k}{p_k^T A p_k}. \tag{3.7}$$

With this we have the following result.

Theorem 3.3.1. *Let A be a $n \times n$ symmetric positive definite matrix, let $\{p_0, p_1, \ldots, p_{n-1}\}$ be a system of n directions A-conjugate and let be:*

$$\varphi(x) = \frac{1}{2} x^T A x - b^T x + e, \quad b \in \mathbb{R}^n, e \in \mathbb{R}.$$

For any starting point $x_0 \in \mathbb{R}^n$, let define the sequence $x_{k+1} = x_k + \alpha_k p_k$ with α_k defined by 3.7.

Then exist $m \leq n - 1$ such that x_{m+1} is the minimum of φ.

Without numerical errors, the conjugate direction method ends in at most n step, but it requires to known n directions mutually conjugate.

The Conjugate Gradient method is a conjugate direction method with a very special property: In generating its set of conjugate vectors, it can compute a new vector p_{k+1} by using only the previous vector p_k. It does not need to know all the previous elements $p_0, p_1, \ldots, p_{k-1}$ of the conjugate set; p_{k+1} is automatically conjugate to these vectors. This remarkable property implies that the method requires little storage and computation.

In the conjugate gradient method, each direction p_k is chosen to be a linear combination of the negative residual:

$$-r_{k+1} = b - Ax_{k+1} = -\nabla\varphi(x_{k+1}),$$

and the previous direction p_k:

$$p_{k+1} = -r_{k+1} + \beta_{k+1} p_k, \tag{3.8}$$

where the scalar β_{k+1} is to be determined by the requirement that p_k and p_{k+1} must be conjugate with the respect to A. By premultiplying 3.8 by $p_k^T A$ and imposing the condition $p_k^T A p_{k+1} = 0$, we find that

$$\beta_{k+1} = \frac{r_{k+1}^T A p_k}{p_k^T A p_k}. \tag{3.9}$$

we can obtain, also, r_{k+1}:

$$r_{k+1} = \nabla\varphi(x_{k+1}) = Ax_{k+1} - b = A(x_k + \alpha_k p_k) - b = Ax_k + \alpha_k A p_k - b,$$

and therefore:

$$r_{k+1} = r_k + \alpha_k A p_k. \tag{3.10}$$

Now lets multiply by p_k both members of 3.10 and we obtain:

$$r_{k+1}^T p_k = r_k^T p_k + \alpha_k p_k^T A p_k = 0,$$

where the last equality follow from 3.7. Therefore r_{k+1} and p_k are orthogonal.

Now let's multiply by g_k the equality:

$$p_{k+1} = -r_{k+1} + \beta_{k+1} p_k,$$

and we obtain:

$$r_{k+1}^T p_{k+1} = -r_{k+1}^T r_{k+1} + \beta_{k+1} r_{k+1}^T p_k$$

where $\beta_{k+1} r_{k+1}^T p_k = 0$ (r_{k+1} and p_k are orthogonal) and therefore:

$$r_{k+1}^T p_{k+1} = -||r_{k+1}||^2, \tag{3.11}$$

then $r_k^T p_k < 0$ i.e. p_k is a descent direction.

For the Conjugate Gradient method we have the following convergence theorem.

Theorem 3.3.2. *The coniugate gradient method calculate in at most n steps the minimum of:*

$$\varphi(x) = \frac{1}{2} x^T A x - b^T x + e, \quad A > 0.$$

In particular exist $m \leq n-1$ such that for $i = 1, 2, \ldots, m$ we have:

$$r_i^T r_j = 0, \quad p_i^T A p_j = 0, \quad j = 0, 1, \ldots, i-1$$

and $r_{m+1} = 0$.

Remark 3.3.1. *Let's consider:*

$$r_{k+1}^T = r_{k+1}^T (r_{k+1} - r_k) = r_{k+1}^T r_{k+1} - r_{k+1}^T r_k,$$

from this, for the previous theorem, we obtain:

$$r_{k+1}^T (r_{k+1} - r_k) = ||r_{k+1}||^2. \tag{3.12}$$

Furthermore from 3.10 we obtain:

$$A p_k = \frac{r_{k+1} - r_k}{\alpha_k}, \tag{3.13}$$

and therefore from 3.7 and 3.9 and using 3.11, 3.12 and 3.13 we can rewrite α_k and β_{k+1} as:

$$\alpha_k = \frac{||r_k||^2}{p_k^T A p_k}, \quad \beta_{k+1} = \frac{||r_{k+1}||^2}{||r_k||^2}. \tag{3.14}$$

Now we can outline the Linear Conjugate Gradient method:

Algorithm 3.3.1 (Linear Conjugate Gradient method)**.**

```
chosen $x_0 \in \mathbb{R}^n$ and tol $\geq 0$,
set $r_0 = Ax_0 - b$, $p_0 = -r_0$, and $k = 0$.
WHILE $\big(||r_k||^2 \geq 0\big)$
    $\alpha_k = \frac{||r_k||^2}{p_k^T A p_k}$,
    $x_{k+1} = x_k + \alpha_k p_k$,
    $r_{k+1} = r_k + \alpha_k A p_k$,
    $\beta_{k+1} = \frac{||r_{k+1}||^2}{||r_k||^2}$,
    $p_{k+1} = -r_{k+1} + \beta_{k+1} p_k$,
    $k = k + 1$.
END
```

3.3.2 Nonlinear Conjugate Gradient Methods

We have noted that Conjugate Gradient method, can be viewed as a minimization algorithm for the convex function:

$$\min \varphi(x) = \frac{1}{2} x^T A x - b^T x.$$

Now we show how we can adapt the approach to minimize general nonlinear functions ϕ

Fletcher and Reeves showed how to extend the Conjugate Gradient method to nonlinear functions by making two simple changes in Algorithm 3.3.1. First, in place of the formula 3.7 for the step length α_k (which minimizes φ along the search direction p_k), we need to perform a line search that identifies an approximate minimum of the nonlinear function ϕ along p_k. Second, the residual r, which is simply the gradient of φ in Algorithm 3.3.1, must be replaced by the gradient of the nonlinear objective ϕ . These changes give rise to the following algorithm for nonlinear optimization.

Algorithm 3.3.2 (Fletcher-Reeves (FR)).

```
Given $x_0$;
Evaluate $\phi_0 = \phi(x_0), \quad \nabla\phi_0 = \nabla\phi(x_0)$;
Set $p_0 = -\nabla\phi_0, \quad k = 0$;
WHILE $(\nabla\phi_k \neq 0)$
    Compute $\alpha_k$ and set $x_{k+1} = x_k + \alpha_k p_k$;
    Evaluate $\nabla\phi_{k+1}$;
```

$$\beta_{k+1}^{FR} = \frac{\nabla\phi_{k+1}^T \nabla\phi_{k+1}}{\nabla\phi_k^T \nabla f_k}; \tag{3.15}$$

$$p_{k+1} = -\nabla\phi_{k+1} + \beta_{k+1}^{FR} p_k; \tag{3.16}$$

$$k = k + 1; \tag{3.17}$$

```
END
```

If we choose ϕ to be a strongly convex quadratic function and α_k to be the exact minimizer, this algorithm reduces to the Linear Conjugate Gradient method.

We need to be more precise about the choice of line search parameter α_k. Because of the second term in 3.16 the search direction p_k may fail to be a descent direction unless α_k satisfies certain condition.

By taking the inner product of 3.16 with the gradient vector $\nabla\phi_k$, we obtain:

$$\nabla\phi_k^T p_k = -||\nabla\phi_k||^2 + \beta_k^{FR} \nabla\phi_k^T p_{k-1}. \tag{3.18}$$

If the line search is exact, so that α_{k-1} is a local minimizer of ϕ along the direction p_{k-1}, we have that $\nabla\phi_k^T p_{k-1} = 0$. In this case we have from 3.18 that $\nabla\phi_k^T p_{k-1} < 0$, then p_k is a descent direction. If the search line is not exact, however, the second term in 3.18 may dominate the first term, and we may have $\nabla\phi_k^T p_{k-1} > 0$, implying tha p_k is actually a direction of ascent. We can avoid this situation by requiring the step length α_k to satisfy the *strong* Wolfe conditions:

$$\begin{aligned} \phi(x_k + \alpha_k p_k) &\leq \phi(x_k) + c_1 \alpha_k \nabla\phi_k^T p_k, \\ |\nabla\phi(x_k + \alpha_k p_k)^T p_k| &\leq -c_2 \nabla\phi_k^T p_k, \end{aligned} \tag{3.19}$$

where $0 < c_1 < c_2 < \frac{1}{2}$. We can show that conditions 3.19 implies that 3.18 is negative, therefore, we conclude that any line search procedure that yields an α_k satisfying 3.19 will ensure that all direction p_k are descent direction for the function ϕ.

Unlike the Linear Conjugate Gradient method, whose convergence properties are well understood and which is known to be optimal as described above, Nonlinear Conjugate Gradient methods possess surprising, sometimes bizarre, convergence properties. We now present a few of the main results known for the Fletcher-Reeves.

For this purpose we make the following (nonrestrictive) assumptions on the objective function.

1. The level st $\mathcal{L} := \{x|\phi(x) \leq \phi(x_0)\}$ is bounded.

2. In some open neighborhood $\mathcal{N}$ of $\mathcal{L}$, the objective function ϕ is Lipschitz continuosly differentiable.

Under this assumption we can build a global convergence result for the FR method:

Theorem 3.3.3 (Al-Baali)**.** *Suppose that the previous assumption hold and Algorithm 3.3.2 is implemented with a line search that satisfied the strong Wolfe conditions 3.19, with* $0 < c_1 < c_2 < \frac{1}{2}$. *Then:*

$$\liminf_{k\to\infty} ||\nabla\phi_k|| = 0. \tag{3.20}$$

In general Al-Baali show that if exist α_k satisfying the strong Wolfe condition, then for some $c > 0$ we have:

$$\cos\theta_k \geq c\frac{||\nabla\phi_k||}{||p_k||}, \tag{3.21}$$

where

$$\cos\theta_k = \frac{-\nabla\phi_k^T p_k}{||\nabla\phi_k||||p_k||}. \tag{3.22}$$

Therefore substituting 3.22 in 3.21, we obtain:

$$\nabla\phi_k^T p_k \leq -c||\nabla\phi_k||^2,$$

consequently p_k is a descent direction for ϕ in x_k.

Despite the good convergence property Fletcher-Reeves method has a weakness. Suppose that p_k is a poor search direction, in the sense that it makes an angle of nearly 90° with $-\nabla\phi_k$, that is, $\cos\theta_k \approx 0$. From this we can show that:

$$||\nabla\phi_k|| \ll ||p_k||.$$

Since p_k is almost orthogonal to the gradient, it is likely that the step from x_k to x_{k+1} is tiny, that is, $x_{k+1} \approx x_k$. If so, we have $\nabla\phi_{k+1} \approx \nabla\phi_k$, and therefore:

$$\beta_{k+1}^{FR} \approx 1,$$

by 3.15. Finally using this approximation together with $||\nabla\phi_{k+1}|| \approx ||\nabla\phi_k|| \ll ||p_k||$ in 3.16, we obtain:

$$p_{k+1} \approx p_k,$$

so the new search direction will improve little (if at all) on the previous one. It follows that if the condition $\cos\theta_k \approx 0$ holds at some iteration k and if the subsequent step is small, a long sequence of unproductive iterates will follow.

An important variant, proposed by Polyak, Polak and Ribiere, defines this parameter as follows:

$$\beta_{k+1}^{PPR} = \frac{\nabla\phi_{k+1}^T(\nabla\phi_{k+1} - \nabla\phi_k)}{||\nabla\phi_k||^2} \tag{3.23}$$

The algorithm, therefore, become:

Algorithm 3.3.3 (Polyak-Polak-Ribiere (PPR))**.**

```
*Given* $x_0$*;*
*Evaluate* $\phi_0 = \phi(x_0), \quad \nabla\phi_0 = \nabla\phi(x_0)$;
*Set* $p_0 = -\nabla\phi_0, \quad k = 0$;
*WHILE* $(\nabla\phi_k \neq 0)$
    *Compute* $\alpha_k$ *and set* $x_{k+1} = x_k + \alpha_k p_k$;
    *Evaluate* $\nabla\phi_{k+1}$;
```

$$\beta_{k+1}^{PPR} = \frac{\nabla\phi_{k+1}^T(\nabla\phi_{k+1} - \nabla\phi_k)}{||\nabla\phi_k||^2}$$

$$p_{k+1} = -\nabla\phi_{k+1} + \beta_{k+1}^{PPR} p_k; \tag{3.24}$$

$$k = k + 1;$$

END

It is identical to Algorithm 3.3.2 when ϕ is a strongly convex quadratic function and the line search is exact, since by Theorem 3.3.2 the gradients are mutually orthogonal, and so $\beta_{k+1}^{PPR} = \beta_{k+1}^{FR}$.

When applied to general nonlinear functions with inexact line searches, however, the behavior of the two algorithms differs markedly. Numerical experience indicates that Algorithm 3.3.3 tends to be the more robust and efficient of the two.

Remark 3.3.2. *If the search direction p_k satisfies $\cos\theta_k \approx 0$ for some k, and if the subsequent step is small, it follow by sobstituting $\nabla\phi_k \approx \nabla\phi_{k+1}$ into 3.23 that $\beta_{k+1}^{PPR} \approx 0$. From the formula 3.24, we find that the new search direction p_{k+1} will be close to the steepest descent direction $-\nabla\phi_{k+1}$, and $\cos\theta_{k+1}$ will be close to 1. Therefore, Algorithm PPR essentially perform a restart after it encounters a bad direction.*

The PPR method contrary to FR method, is very efficient but have some difficullty of convergence in the general case.

Suppose ϕ is strict convex, then PPR method with exact line search converge.

Proposition 3.3.1. *Let be $\phi : \mathbb{R}^n \to \mathbb{R}$ twice differentiable, with continuos derivate, in an open convex set D containing the compact level set $\mathcal{L}$ (i.e. $\{x|\phi(x) \leq \phi(x_0)\}$) . Suppose, furthermore, that exist $0 < \delta_1 \leq \delta_2$ such that:*

$$\delta_1||h||^2 \leq h^T\nabla^2\phi(x)h \leq \delta_2||h||^2, \quad \forall x \in \mathcal{L}, \forall h \in \mathbb{R}^n.$$

let be $\{x_k\}_{k\in\mathbb{N}}$ the sequence generated by PPR method with $\nabla\phi_k \neq 0$ and

$$\alpha_k = \arg\min_{\alpha\geq 0} \phi(x_k + \alpha p_k).$$

Then the sequence $\{x_k\}$ converge to the minimum of ϕ in $\mathbb{R}^n$.

In general case Powell, in 1981, showed that the PPR method can cycle infinitely without approaching a solution point. to guarantee convergence in the general case we can use inexact line search or modify β_{k+1}.

The following proposition show some convergence condition that can be see as requirement for the line search.

Proposition 3.3.2. *Let be $\phi : \mathbb{R}^n \to \mathbb{R}$ Lipschitz continuosly differentiable in an open convex set D containing the compact level set $\mathcal{L}$. Let be $\{x_k\}_{k\in\mathbb{N}}$ the sequence generated by PPR method with $\nabla\phi_k \neq 0$ and α_k that satisfy:*

- $x_k \in \mathcal{L}$;
- $\lim_{k\to\infty} \frac{|\nabla\phi_k^T p_k|}{||p_k||} = 0$;
- $\lim_{k\to\infty} ||\alpha_k p_k|| = 0$.

Then exist an accumulation point of $\{x_k\}$ that is a stationary point for ϕ.

To satisfy the requirement for the previous proposition we can modify the Armijo algorithm with backtraking:

Algorithm 3.3.4 (Modified Armijo)**.**

Choose $0 < \rho_1 < \rho_2, \quad \gamma \in (0,1), \quad \delta \in [0,1), \quad \theta \in (0,1)$;
set $\tau_k = \frac{|\nabla\phi_k^T p_k|}{||p_k||^2}$ *and chose* $\Delta_k \in [\rho_1\tau_k, \rho_2\tau_k]$,
evaluate $\alpha_k = \max_{\{j=0,1,\ldots\}}\{\theta^j\Delta_k\}$ *such that:*

$$x_{k+1} = x_k + \alpha_k p_k$$

$$p_{k+1} = -\nabla\phi_{k+1} + \beta_{k+1}p_k$$

satisfy

- $\phi(x_{k+1}) \leq \phi(x_k) + \gamma\alpha_k\nabla\phi_k^T p_k$,
- $\nabla\phi_{k+1}^T p_{k+1} \leq -\delta||\nabla\phi_{k+1}||^2$.

Now using β_{k+1}^{PPR} with the choice of α_k with the previous alghorithm we assure the convergence.

Proposition 3.3.3. *Let be $\phi : \mathbb{R}^n \to \mathbb{R}$ Lipschitz continuosly differentiable in an open convex set D containing the compact level set $\mathcal{L}$. Let be $\{x_k\}_{k\in\mathbb{N}}$ the sequence generated by PPR method with $\nabla\phi_k \neq 0$ and α_k calculated using the Armijo modified Algorithm. Then exist an accumulation point of $\{x_k\}$ that is a stationary point for ϕ.*

We show, also, a version of the PPR method that converge thanks to the change of β_{k+1}^{PPR}. Powell, demonstrate the following result:

Proposition 3.3.4. *Let be $\phi : \mathbb{R}^n \to \mathbb{R}$ Lipschitz continuosly differentiable in an open convex set D containing the compact level set $\mathcal{L}$. Let be $\{x_k\}_{k\in\mathbb{N}}$ the sequence generated by PPR method with $\beta_{k+1}^{+} = \max\{0, \beta_{k+1}^{PPR}\}$, $\nabla f_k \neq 0$ and α_k satisfy:*

- *the Zoutendijk condition:*

$$\sum_{k=0}^{\infty} ||\nabla\phi_k||^2 \cos^2\theta_k < \infty;$$

- *the sufficient descent condition*

$$\nabla\phi_k^T p_k \leq -c||\nabla_k||^2$$

for some $c > 0$.

Then exist an accumulation point of $\{x_k\}$ that is a stationary point for ϕ.

If for all iteration of PPR method was $0 \leq \nabla\phi_{k+1}^T\nabla\phi_K \leq ||\nabla\phi_{k+1}||^2$, then would be valid the same demonstration made for the convergence of the FR method. This condition, however, isn't always satisfyed, but, is equal to $0 \leq \beta_{k+1}^{PPR} \leq \beta_{k+1}^{FR}$. So we can consider an *Hybrid Conjugate Gradient Method* with:

$$\beta_{k+1} = \begin{cases} \beta_{k+1}^{PPR} & \text{if} \quad 0 \leq \beta_{k+1}^{PPR} \leq \beta_{k+1}^{FR} \\ \beta_{k+1}^{FR} & \text{otherwise.} \end{cases} \tag{3.25}$$

This method was proposed by Touati-Ahmed and Storey in 1990.

Remark 3.3.3. *If the algorithm generate $x_k \approx x_{k+1}$, then, $\beta_{k+1}^{PPR} \approx 0$ and $\beta_{k+1}^{FR} \approx 1$, therefore β_{k+1} will be updated with the PPR method and not with the FR one, that would generate tiny step. In this way we obtain the better from both algorithm:*

- *convergence from FR method;*

- *efficiency from PPR method.*

The Hybrid Conjugate Gradient Method has global convergence, indipendently of the fact that α_k would be find with exact or inexact line search. A more efficient method is given by:

$$\beta_{k+1} = \max\{0, \min\{\beta_{k+1}^{FR}, \beta_{k+1}^{PPR}\}\}$$

that assure $\beta_{k+1} \geq 0$.

Capitolo 4

Numerical Results

In this chapter we are going to present and comment the results of the experiments for breast imaging reconstruction, that is, the solution of the non-linear least square problem 2.8.

In our test problems for a simulated breast imaging reconstruction we used one simuleted three-dimesional phantom object, of size $31 \times 31 \times 7$ and $65 \times 65 \times 7$, made of four ellipses consisting of a tissue mixture with varying precentages of glandur and adipose tissue, while the background is made of a mixture of 50% adipose and 50% glandular tissue. We can see an example of the central slice in Figure 4.1.

We, first, used the phantom of size $31 \times 31 \times 7$ to search the optimal regularization parameter λ. We have tested, for both Total Variation regularization (TV) and 1-Norm regularization (1N), two different values of the noise: $\eta = 10^{-3}$ and $\eta = 5 \cdot 10^{-4}$.

For these tests we chose to consider 20 values of λ, from 0 to 0.4 for TV and from 0 to 1 for 1N uniformly, and to use these two stop conditions:

1. Iteration $(k) \leq 2000$;
2. $\frac{||x_k - x_{k-1}||_{fro}}{||x_{k-1}||_{fro}} < 10^{-n}$;

with $n = 5$ for TV regularization and $n = 4$ for 1N regularization.

We, then, chose λ using two criteria. As first criterion, the parameter that minimize the relative error:

$$\frac{||x_{ex} - x||}{||x_{ex}||}.$$

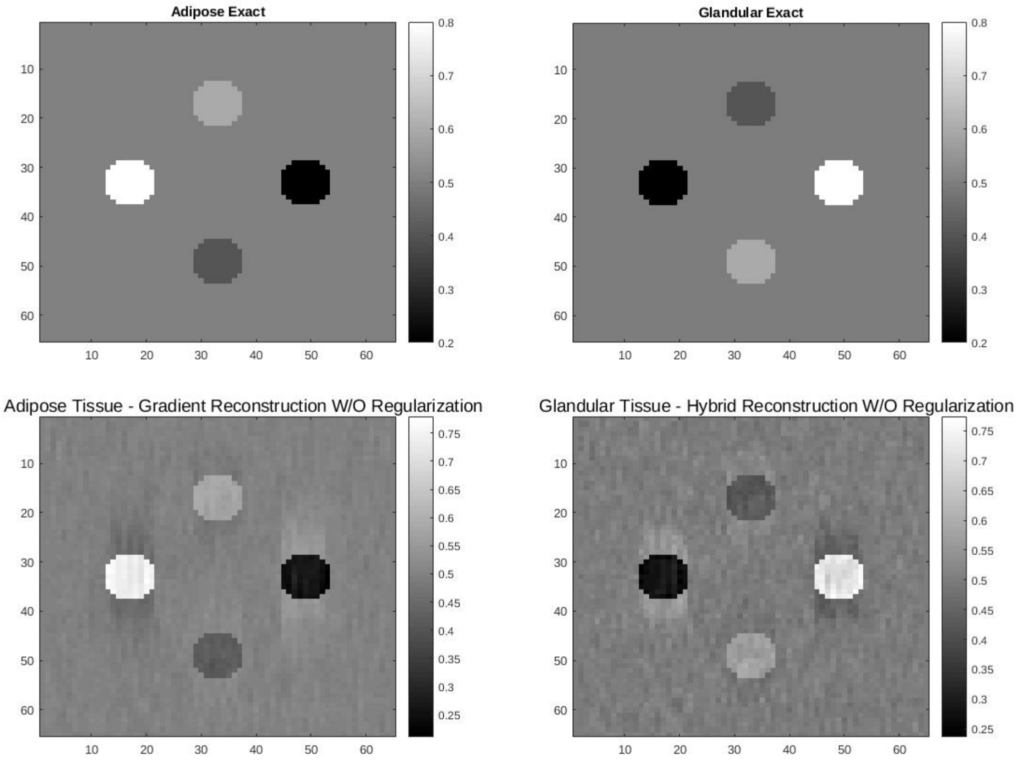


Figura 4.1: On the top left: exact adipose tissue; on the top right: exact glandular tissue; on the bottom left: reconstructed adipose tissue with Gradient method without regularization; on the bottom right: reconstructed glandular tissue with Hybrid method without regularization;

We could use this criterion only for TV regularization because we obtain reliable graphics with a clear minimum, as we can see in Figure 4.2.

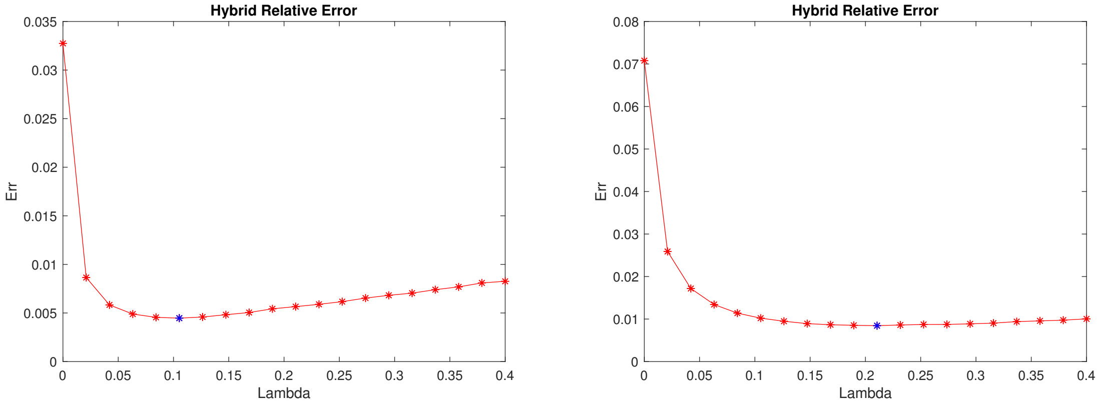


Figura 4.2: On the left: relative error vary depending on λ for Hybrid method with TV and $\eta = 5 \cdot 10^{-4}$; On the right: relative error vary depending on λ for Hybrid method with TV and $\eta = 1 \cdot 10^{-3}$

As second criterion, for the 1N regularization, the parameter that obtain the better image. Indeed, due to the semiconvergence of the 1N regularization we obtain unreliable graphics.

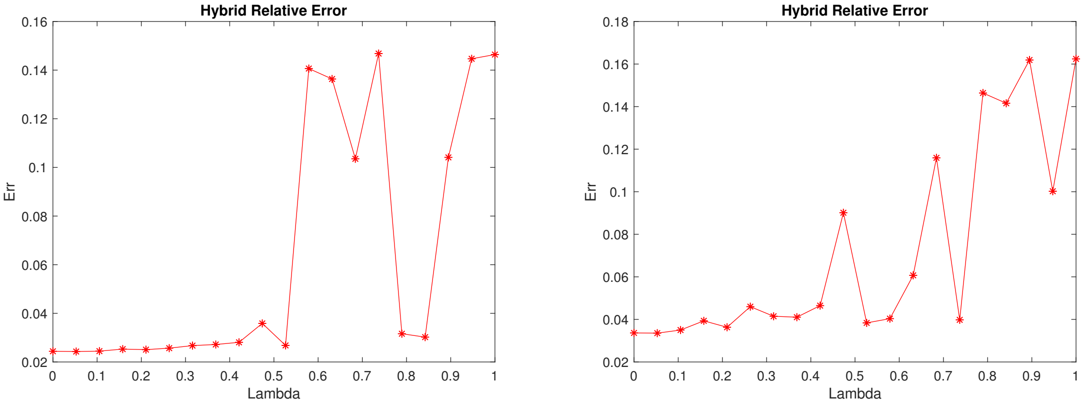


Figura 4.3: On the left: relative error vary depending on λ for Hybrid method with 1N and $\eta = 5 \cdot 10^{-4}$; On the right: relative error vary depending on λ for Hybrid method with 1N and $\eta = 1 \cdot 10^{-3}$

After these tests we found the parameter λ for all, method and noise, and the results are summarised in the following table:

Method	Regularization	Noise η	λ
Gradient	TV	$5 \cdot 10^{-4}$	0.1
Gradient	TV	$1 \cdot 10^{-3}$	0.2
Gradient	1N	$5 \cdot 10^{-4}$	0.4
Gradient	1N	$1 \cdot 10^{-3}$	0.5
Hybrid Conjugate Gradient	TV	$5 \cdot 10^{-4}$	0.1
Hybrid Conjugate Gradient	TV	$1 \cdot 10^{-3}$	0.2
Hybrid Conjugate Gradient	1N	$5 \cdot 10^{-4}$	0.4
Hybrid Conjugate Gradient	1N	$1 \cdot 10^{-3}$	0.5

We can see that, TV regularization requires a lower parameter λ and therefore it obtains a closer solution to the real solution. Furthermore, as expected, we need a greater values of λ if we set a greater values for the noise.

Once we've set the parameter λ, we started the simulations of the reconstructions, first with TV regularization and then with 1N regularization, for both Gradient and Hybrid Conjugate Gradient algorithms, previosly described in Section 3.2 and 3.3.

4.1 Numerical Results for Total Variation Regularization

In this section we'll comment the result obtained using TV regularization. We begin the analysis considering the problem with a level of noise $\eta = 5 \cdot 10^{-4}$

In figure 4.4 we can see that both methods obtain cleaner and more precise images using the TV regularization than the same methods without regularization. Is it enough to look at these images to understand the usefulness and superiority of methods that use TV regularization. Therefore, from here on out, we analyse only the two methods with the regularization.

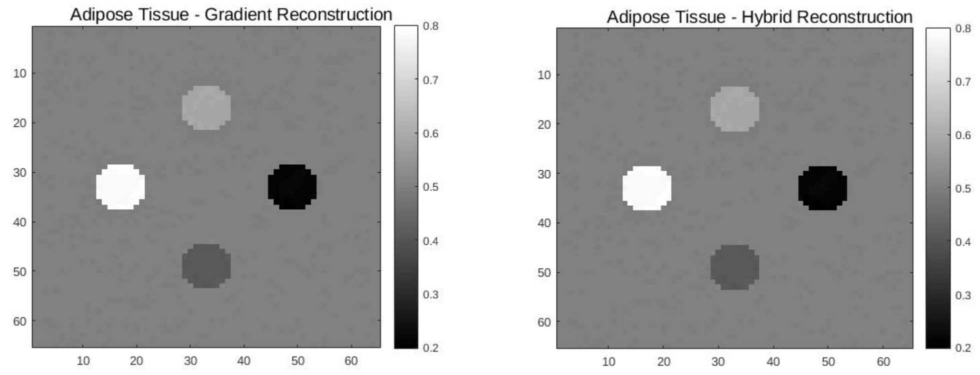


Figura 4.4: On the left: reconstructed image with Gradient method with TV and $\eta = 5 \cdot 10^{-4}$; On the right: reconstructed image with Hybrid method with TV and $\eta = 5 \cdot 10^{-4}$

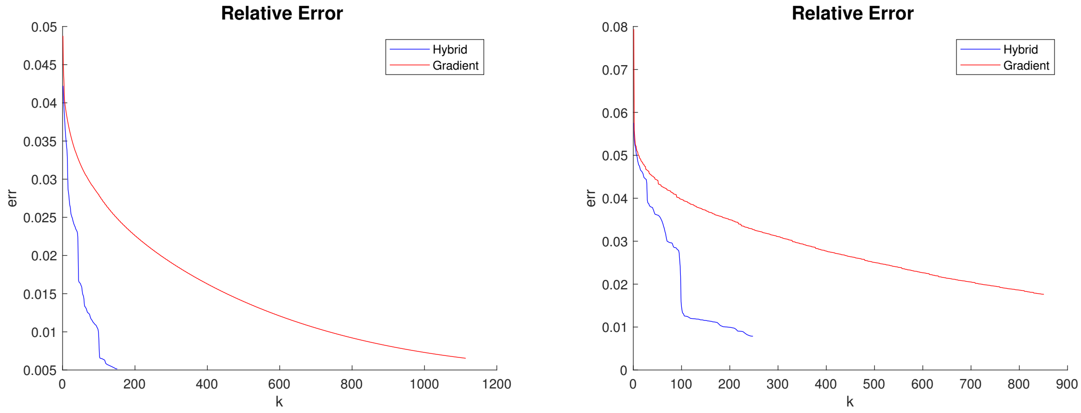


Figura 4.5: On the left: relative error vary depending on iteration for Hybrid and Gradient method with TV, $\eta = 5 \cdot 10^{-4}$ and dimension $31 \times 31 \times 7$; on the right: relative error vary depending on iteration for Hybrid and Gradient method with TV, $\eta = 5 \cdot 10^{-4}$ and dimension $65 \times 65 \times 7$.

Looking at the graphics in Figure 4.5 we note that Hybrid method obtains the best results, regarding the relative error, in the shortest time. Indeed, in the resolution of the test problem of dimension $31 \times 31 \times 7$, the Hybrid method obtains a relative error of $5.14 \cdot 10^{-3}$ doing 151 iteration, against a relative error of $6.53 \cdot 10^{-3}$ with 1114 iteration of the Gradient method. Also in the test problem of greater dimension the Hybrid method obtains better results than the Gradient one, not only regarding the relative error and time but also in the precision of the reconstruction.

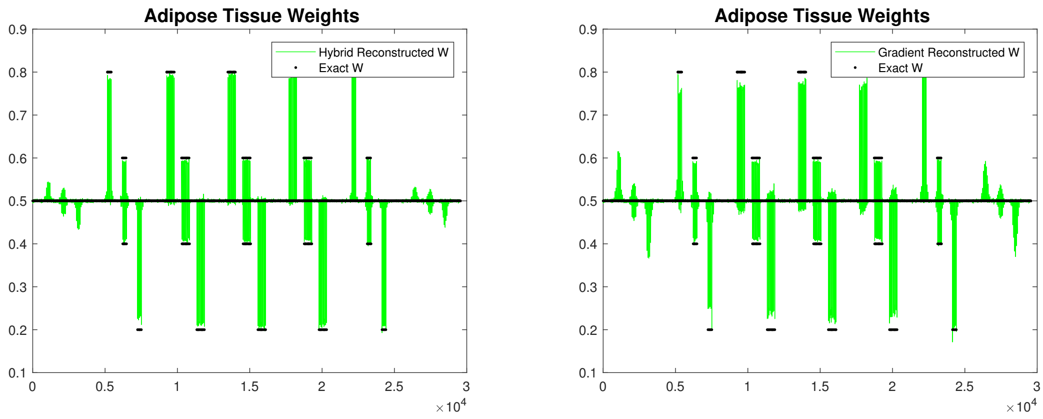


Figura 4.6: On the left: graphics of the weight for adipose tissue reconstructed with Hybrid method with TV and $\eta = 5 \cdot 10^{-4}$ and dimension $65 \times 65 \times 7$; On the right: graphics of the weight of adipose tissue reconstructed with Gradient method with TV and $\eta = 5 \cdot 10^{-4}$ $65 \times 65 \times 7$

Indeed, in Figure 4.6 we can see the graphics of the weights for adipose tissue for both methods. In black we have the exact weights while, in green the reconstructed weights. We can note that, the weights reconstructed with Hybrid method are more precise. The green columns of the Hybrid method are closer to the black line, than the green columns of the Gradient method. Both methods produce, in the first and last slice, a sort of shadow of the solutions of the inner slices. The Hybrid method is the best one, also, in this aspect. The shadow artifacts produced by the Hybrid method are more limited than those produced by Gradient method. These shadows, anyway, are very limited for both methods and in real breast imaging reconstruction it is merged with the background.

We resume the main results for Gradient and Hybrid for $\eta = 5 \cdot 10^{-4}$ in this table.

Method	Dimension	Relative Error	Time	Iteration
Gradient	$31 \times 31 \times 7$	$6.53 \cdot 10^{-3}$	$1.23 \cdot 10^{3}$	1114
Hybrid	$31 \times 31 \times 7$	$5.14 \cdot 10^{-3}$	$1.63 \cdot 10^{2}$	151
Gradient	$65 \times 65 \times 7$	$1.75 \cdot 10^{-2}$	$3.33 \cdot 10^{4}$	851
Hybrid	$65 \times 65 \times 7$	$7.85 \cdot 10^{-3}$	$9.56 \cdot 10^{3}$	248

Now, we'll analyse the results obtained with the higher value of the noise $\eta = 10^{-3}$.

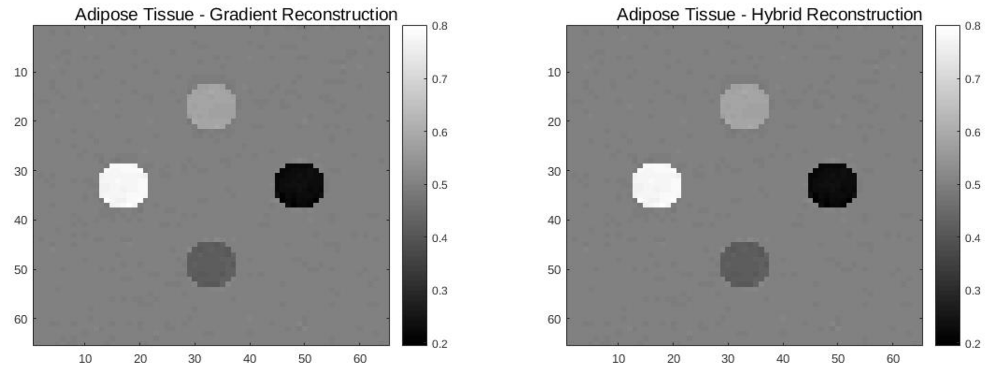

Adipose Tissue - Gradient Reconstruction
Adipose Tissue - Hybrid Reconstruction
10
20
30
40
50
60
0.2
0.3
0.4
0.5
0.6
0.7
0.8

Figura 4.7: On the left: reconstructed image with Gradient method with TV and $\eta = 1 \cdot 10^{-3}$; on the right: reconstructed image with Hybrid method with TV and $\eta = 1 \cdot 10^{-3}$

Looking at the images in Figure 4.7 we can deduce that, the two methods work in a linear way regarding the growth of the noise. The Hybrid method produces, again, the cleanest image and, we'll see, in the shortest time. The only difference, trivially, is that the images are noiser than before.

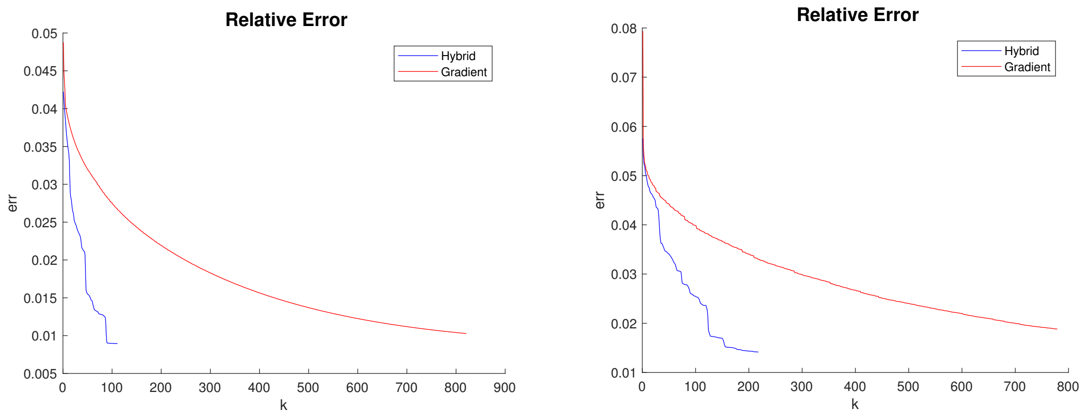


Figura 4.8: On the left: relative error vary depending on iteration for Hybrid and Gradient method with TV, $\eta = 1 \cdot 10^{-3}$ and dimension $31 \times 31 \times 7$; on the right: relative error vary depending on iteration for Hybrid and Gradient method with TV, $\eta = 1 \cdot 10^{-3}$ and dimension $65 \times 65 \times 7$.

As expected the graphics of Figure 4.8 do not differ too much from the graphics in Figure 4.5. More interesting are the graphics of the weight.

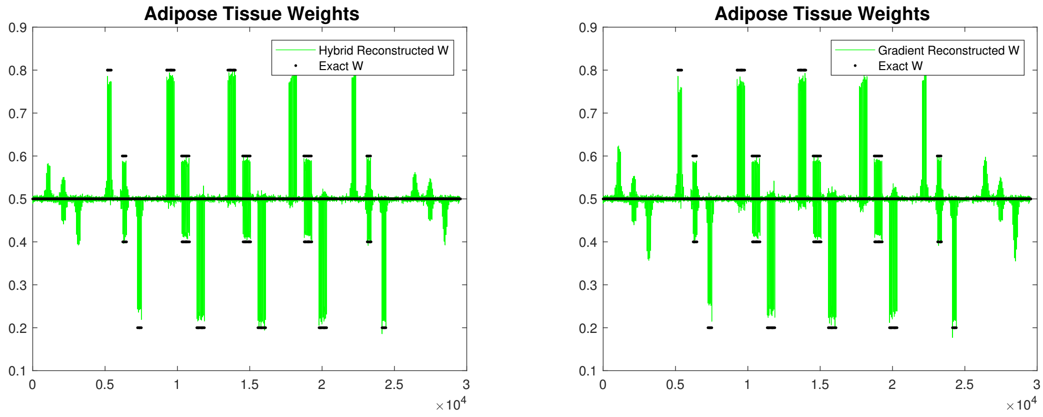


Figura 4.9: On the left: graphics of the weight for adipose tissue reconstructed with Hybrid method with TV and $\eta = 10^{-3}$ and dimension $65 \times 65 \times 7$; On the right: graphics of the weight of adipose tissue reconstructed with Gradient method with TV and $\eta = 10^{-3}$ and dimension $65 \times 65 \times 7$

In Figure 4.9 we can note that, the algorithms can't eliminate as much noise as before. Indeed, some of the weights seems to be distibuted on the background. This is the effect of the higher level of noise. Also the shadow artifacts are more intense with higher values of η.

As before, we resume the main results in the following table:

Method	Dimension	Relative Error	Time	Iteration
Gradient	$31 \times 31 \times 7$	$1.02 \cdot 10^{-2}$	$9.06 \cdot 10^{2}$	821
Hybrid	$31 \times 31 \times 7$	$8.93 \cdot 10^{-3}$	$1.18 \cdot 10^{2}$	111
Gradient	$65 \times 65 \times 7$	$1.87 \cdot 10^{-2}$	$3.03 \cdot 10^{4}$	779
Hybrid	$65 \times 65 \times 7$	$1.41 \cdot 10^{-2}$	$8.72 \cdot 10^{3}$	218

We can conclude that, using TV regularization, we obtain, with both algorithms, better results, regarding both the clearness of the images and the relative error, than using the same methods without regularization. Furthermore, the Hybrid method obtains better results than the Gradient method for both values of noise. The Gradient method, however, seems to bear the growth of the noise, at least for greater dimension, better than the Hybrid one. The relative error of the Gradient method, indeed, goes from

$1.75 \cdot 10^{-2}$ to $1.87 \cdot 10^{-2}$, while, for the Hybrid method goes from $7.85 \cdot 10^{-3}$ to $1.41 \cdot 10^{-2}$.

4.2 Numerical results for 1-Norm Regularization

We begin to comment, as in the previous section, the results obtained with noise $\eta = 5 \cdot 10^{-4}$.

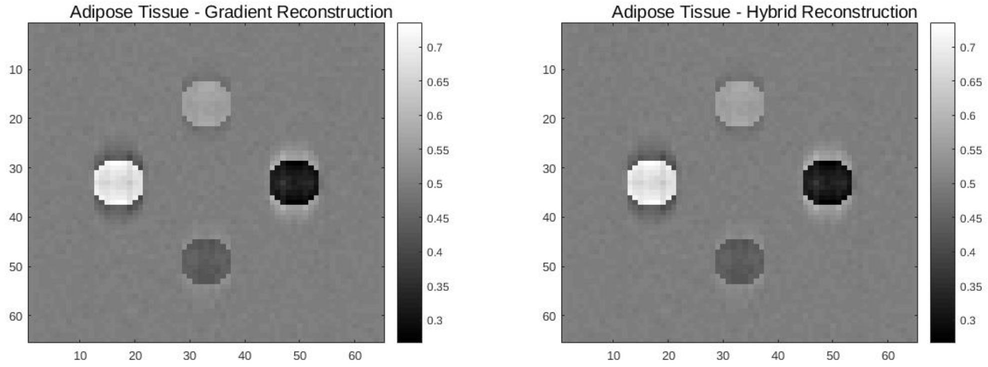

Adipose Tissue - Gradient Reconstruction
Adipose Tissue - Hybrid Reconstruction
10
20
30
40
50
60
0.7
0.65
0.6
0.55
0.5
0.45
0.4
0.35
0.3

Figura 4.10: On the left: reconstructed image with Gradient method with 1N and $\eta = 5 \cdot 10^{-4}$; On the right: reconstructed image with Hybrid method with 1N and $\eta = 5 \cdot 10^{-4}$

Looking at the Figure 4.10 we can note some artifacts all around the ellipses. These artifacts are typical of this regularization for breast imaging reconstruction problem, indeed, we've found them in all of our test. Furthermore, comparing the colorbar of this Figure with the Figure 4.4, we note that, the reconstructions with 1N are less precise than the one with TV. The reconstructions are also less precise than the one obtained without regularization, but, the images are clearer. This type of regularization has an effect of deblurring on the reconstructed images.

The Gradient method results slower than the Hybrid one, also for the 1N regularization. From Figure 4.11 seems that, both methods end much earlier than those using TV regulariziation. But, we've to recall that, for algorithms using this type of regularization we've imposed an higher stopping tollerance due to the semiconvergence of these methods.

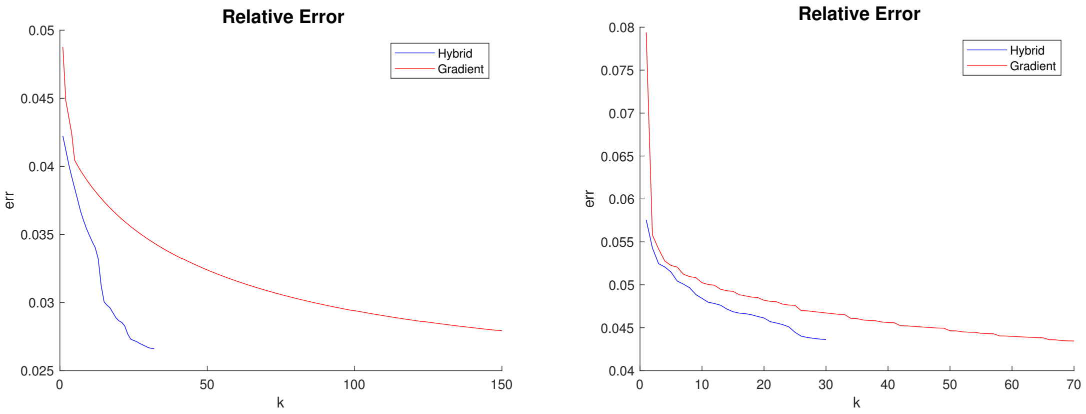


Figura 4.11: On the left: relative error vary depending on iteration for Hybrid and Gradient method with 1N, $\eta = 5 \cdot 10^{-4}$ and dimension $31 \times 31 \times 7$; on the right: relative error vary depending on iteration for Hybrid and Gradient method with 1N, $\eta = 5 \cdot 10^{-4}$ and dimension $65 \times 65 \times 7$.

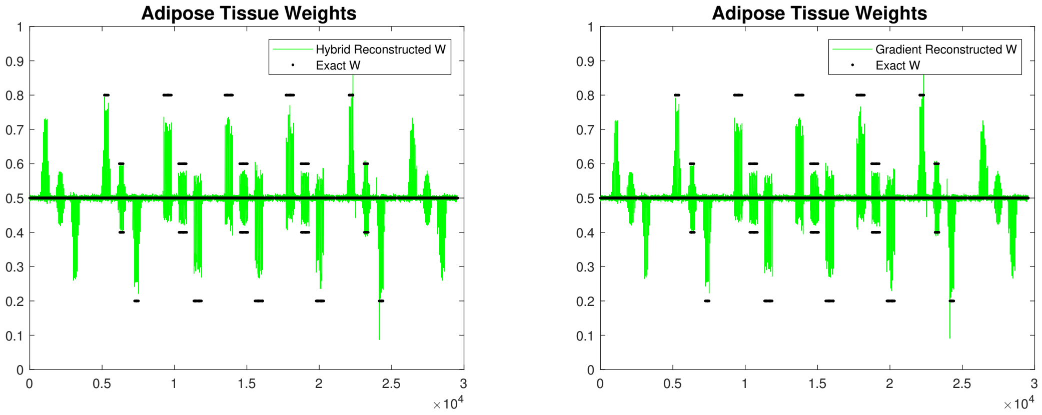


Figura 4.12: On the left: graphics of the weight for adipose tissue reconstructed with Hybrid method with 1N and $\eta = 5 \cdot 10^{-4}$ and dimension $65 \times 65 \times 7$; On the right: graphics of the weight of adipose tissue reconstructed with Gradient method with 1N and $\eta = 5 \cdot 10^{-4}$ and dimension $65 \times 65 \times 7$

Looking at the weights in Figure 4.12 we clearly note that the 1-Norm regularization doesn't remove the noise as the TV regularization. Indeed, we can see the noise all along the slices also for small values of η. Furthermore, the shadow artifacts in the first and last slices are more intense than those obtained with TV. There aren't important difference between the two algorithms, the two reconstructions are more or less the same.

As for the TV regularization we now summarise the main results in the following table:

Method	Dimension	Relative Error	Time	Iteration
Gradient	$31 \times 31 \times 7$	$2.79 \cdot 10^{-2}$	$1.63 \cdot 10^{2}$	150
Hybrid	$31 \times 31 \times 7$	$2.65 \cdot 10^{-2}$	$3.41 \cdot 10^{1}$	32
Gradient	$65 \times 65 \times 7$	$4.33 \cdot 10^{-2}$	$2.73 \cdot 10^{3}$	70
Hybrid	$65 \times 65 \times 7$	$4.34 \cdot 10^{-2}$	$1.13 \cdot 10^{3}$	30

This table confirms what we said before: the Gradient method is slower than the Hybrid one for both dimensions of the test problems; the two recostructions are very close, in fact the relative errors are similar. For the higher dimension, however, we can note that, for the first time, Hybrid method obtains a relative error greater than the one obtained by the Gradient method. This happen due to the semiconvergence typical of this regularization. The semiconvergence is faster in the Hybrid method and our stop criteria aren't always enough to prevent it.

We, now, analyse the results for $\eta = 10^{-3}$.

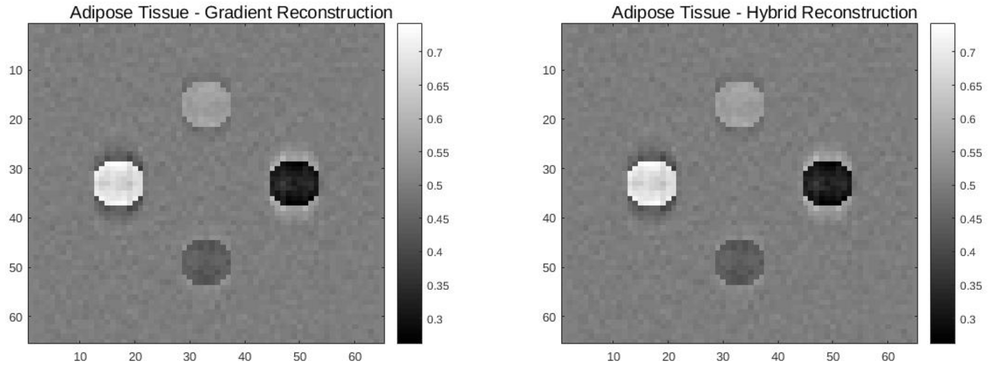

Adipose Tissue - Gradient Reconstruction
Adipose Tissue - Hybrid Reconstruction
10
20
30
40
50
60
0.7
0.65
0.6
0.55
0.5
0.45
0.4
0.35
0.3

Figura 4.13: On the left: reconstructed image with Gradient method with 1N and $\eta = 1 \cdot 10^{-3}$; on the right: reconstructed image with Hybrid method with 1N and $\eta = 1 \cdot 10^{-3}$

The two images in Figure 4.13 are, obviously, noiser and the artifacts around the ellipses are more intense. The two reconstructions seem to be more or less the same, but they hide some important differences that mark the two algorithms. Let's analyse the error graphics for more details.

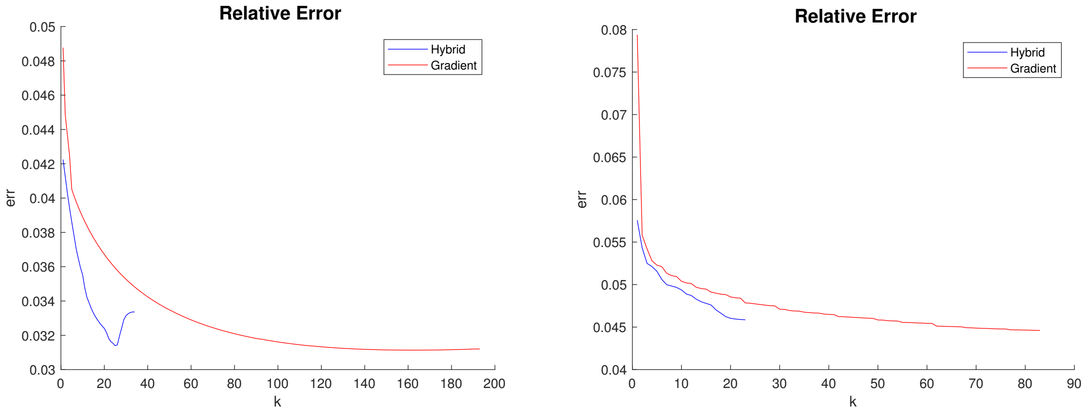


Figura 4.14: On the left: relative error vary depending on iteration for Hybrid and Gradient method with 1N, $\eta = 1 \cdot 10^{-3}$ and dimension $31 \times 31 \times 7$; on the right: relative error vary depending on iteration for Hybrid and Gradient method with 1N, $\eta = 1 \cdot 10^{-3}$ and dimension $65 \times 65 \times 7$.

In Figure 4.14 we can see the first reason why Hybrid method obtains the worst results, regarding relative error, with 1N regularization. After about $20 - 25$ iterations Hybrid algorithm goes against an high effect of semiconvergence, while, the Gradient one, thanks to its slowness, ends before the semiconvergence. The Hybrid method would be the better one, if we can stop it before the semiconvergence. Therefore, one of the future upgrade of this method could be a better stop criterion.

Analysing the graphics in Figure 4.15 painstakingly, we can note that, the reconstructions with Hybrid method are, generally, less noisy than those obtained with Gradient method. They are, however, less precise, regarding the recontruction of the ellipses. This is the second reason why Hybrid method has higher relative error. Indeed, an error in the reconstruction of the ellipses has a greater impact on the relative error than an error in the reconstruction of the background.

Lastly, we resume the main results in the usual table.

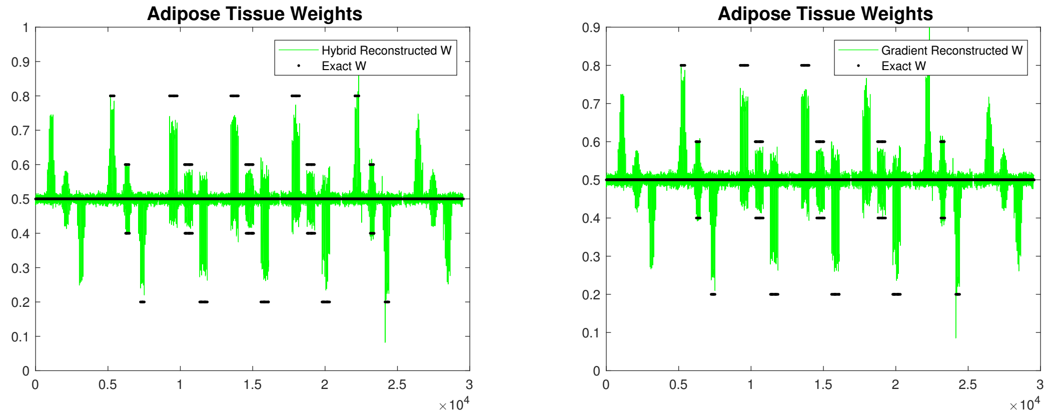


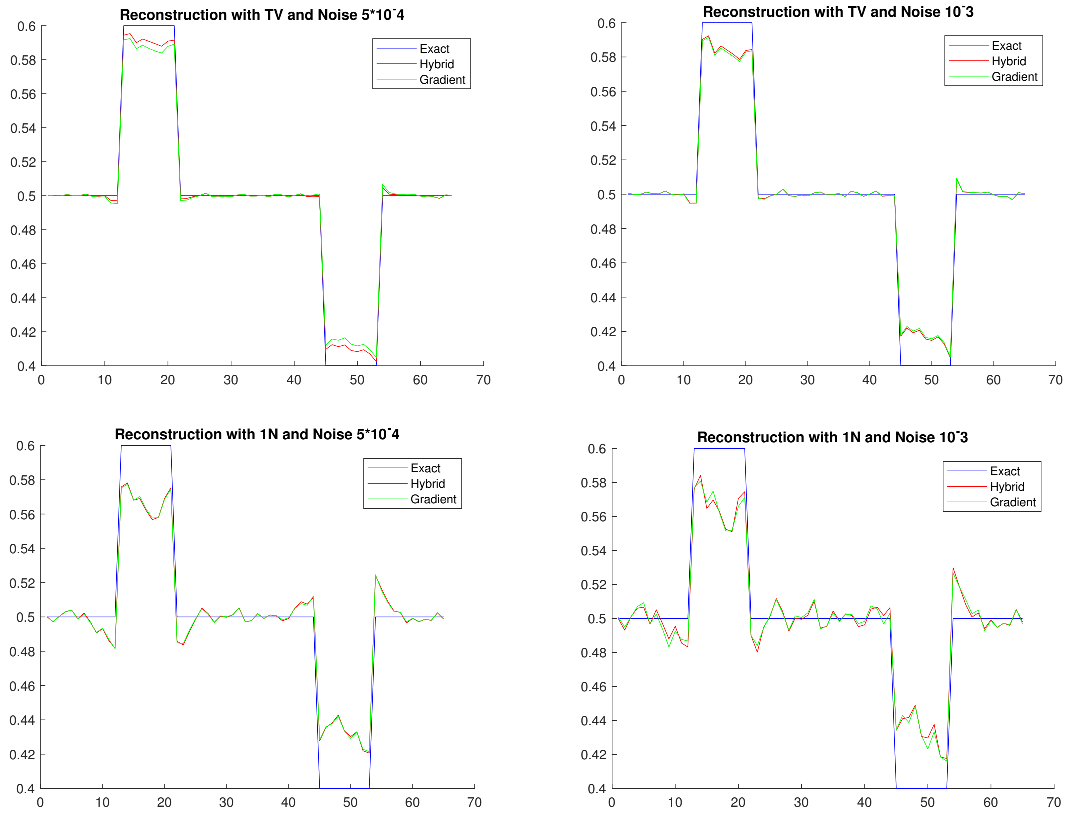
Figura 4.15: On the left: graphics of the weight for adipose tissue reconstructed with Hybrid method with 1N and $\eta = 10^{-3}$ and dimension $65 \times 65 \times 7$; On the right: graphics of the weight of adipose tissue reconstructed with Gradient method with 1N and $\eta = 10^{-3}$ and dimension $65 \times 65 \times 7$

Method	Dimension	Relative Error	Time	Iteration
Gradient	$31 \times 31 \times 7$	$3.11 \cdot 10^{-2}$	$2.10 \cdot 10^{2}$	193
Hybrid	$31 \times 31 \times 7$	$3.33 \cdot 10^{-2}$	$3.66 \cdot 10^{1}$	34
Gradient	$65 \times 65 \times 7$	$4.44 \cdot 10^{-2}$	$3.17 \cdot 10^{3}$	83
Hybrid	$65 \times 65 \times 7$	$4.57 \cdot 10^{-2}$	$9.40 \cdot 10^{2}$	23

The Hybrid method obtains the worst results, regarding the relative error, for both dimensions. But, this doesn't mean that Hybrid method is worse than Gradient method. We, always, have to remember that, relative error, for these types of problem, is often not too reliable. Indeed, looking at the reconstructed images in Figure 4.13, one can't say that, one image is, clearly, better than the other. The only thing we can say is that, Hybrid method works better with TV regularization than with the 1N regularization.

4.3 Conclusions

In this section we'll resume and compare all the results we have obtained.



Reconstruction with TV and Noise 5*10ˆ4

Reconstruction with TV and Noise 10ˆ3

Reconstruction with 1N and Noise 5*10ˆ4

Reconstruction with 1N and Noise 10ˆ3

Figura 4.16: On the top left: central row of the central slice of the reconstruction with TV and $\eta = 5 \cdot 10^{-4}$; on the top right: central row of the central slice of the reconstruction with TV and $\eta = 10^{-3}$; on the bottom left: central row of the central slice of the reconstruction with 1N and $\eta = 5 \cdot 10^{-4}$; on the bottom right: central row of the central slice of the reconstruction with 1N and $\eta = 10^{-3}$;

In Figure 4.16 we can see the main differences between the two tested algorithms and the two regularizations. Using TV regularization, Hybrid method obtains the best reconstruction, especially for lower level of noise. The main property of TV regularization is its effect of denoise. We can note, indeed, that the graphics have very few fluctuations.

On the contrary, looking at the graphics for 1N regularization, we have strong fluctuations and the recontructions are less precise.

Now, we resume all numerical results in the following table and then we'll try to draw some conclusions.

Method	Regularization	Noise	Dimension	Relative Error	Time	Iteration
Gradient	Total Variation	$5 \cdot 10^{-4}$	$31 \times 31 \times 7$	$6.53 \cdot 10^{-3}$	$1.23 \cdot 10^{3}$	1114
Hybrid	Total Variation	$5 \cdot 10^{-4}$	$31 \times 31 \times 7$	$5.14 \cdot 10^{-3}$	$1.63 \cdot 10^{2}$	151
Gradient	Total Variation	$5 \cdot 10^{-4}$	$65 \times 65 \times 7$	$1.75 \cdot 10^{-2}$	$3.33 \cdot 10^{4}$	851
Hybrid	Total Variation	$5 \cdot 10^{-4}$	$65 \times 65 \times 7$	$7.85 \cdot 10^{-3}$	$9.56 \cdot 10^{3}$	248
Gradient	Total Variation	10^{-3}	$31 \times 31 \times 7$	$1.02 \cdot 10^{-2}$	$9.06 \cdot 10^{2}$	821
Hybrid	Total Variation	10^{-3}	$31 \times 31 \times 7$	$8.93 \cdot 10^{-3}$	$1.18 \cdot 10^{2}$	111
Gradient	Total Variation	10^{-3}	$65 \times 65 \times 7$	$1.87 \cdot 10^{-2}$	$3.03 \cdot 10^{4}$	779
Hybrid	Total Variation	10^{-3}	$65 \times 65 \times 7$	$1.41 \cdot 10^{-2}$	$8.72 \cdot 10^{3}$	218
Gradient	1-Norm	$5 \cdot 10^{-4}$	$31 \times 31 \times 7$	$2.79 \cdot 10^{-2}$	$1.63 \cdot 10^{2}$	150
Hybrid	1-Norm	$5 \cdot 10^{-4}$	$31 \times 31 \times 7$	$2.65 \cdot 10^{-2}$	$3.41 \cdot 10^{1}$	32
Gradient	1-Norm	$5 \cdot 10^{-4}$	$65 \times 65 \times 7$	$4.33 \cdot 10^{-2}$	$2.73 \cdot 10^{3}$	70
Hybrid	1-Norm	$5 \cdot 10^{-4}$	$65 \times 65 \times 7$	$4.34 \cdot 10^{-2}$	$1.13 \cdot 10^{3}$	30
Gradient	1-Norm	10^{-3}	$31 \times 31 \times 7$	$3.11 \cdot 10^{-2}$	$2.10 \cdot 10^{2}$	193
Hybrid	1-Norm	10^{-3}	$31 \times 31 \times 7$	$3.33 \cdot 10^{-2}$	$3.66 \cdot 10^{1}$	34
Gradient	1-Norm	10^{-3}	$65 \times 65 \times 7$	$4.44 \cdot 10^{-2}$	$3.17 \cdot 10^{3}$	83
Hybrid	1-Norm	10^{-3}	$65 \times 65 \times 7$	$4.57 \cdot 10^{-2}$	$9.40 \cdot 10^{2}$	23

Comparing the results we convince ourselves that the methods using TV regularization obtain the most precise recontructions. The algorithms using 1N regularization seem to be faster, but only due to the semiconvergence effect. 1N seems to lose on every level against TV, but it has an effect of deblurring on the reconstructions. Therefore this type of regularization could have interesting applications, but it needs some more research, first of all a better stop criterion that prevent the semiconvergence effect.

If we focus ourselves on the algorithms, especially regarding those using TV regularization (the most reliable), we note the superiority of Hybrid method for both speed and precision, but, looking at the graphics in Figure 4.16, we note that Gradient method for higher values of the noise approaches the results obtained by Hybrid method. Therefore, we can suppose that, it could became the best algorithm, regarding the precision, for even higher values of η. Furthermore, we can, also, speed up the Gradient method using, for example, the adaptive rules for the choice of the step-lenght introduced by the work of Barzilai and Borwein.

Therefore, in view of the results we've obtained, we can claim that the best method is the Hybrid one, but we care to remember the potential of the Gradient method.

In conclusion, we want to give some ideas for the future researchs. We said that, TV regularization has an effect of denoising while, 1N regularization has an effect of deblurring. Therefore, why not use a method combining the two regularizations? We think that the premises are very promising.

Bibliografia

[1] J.Nocedal, S.J. Wright, *Numerical Optimization*, 2nd ed Springer (2006).

[2] Per Christian Hansen, *Rank-Deficient and Discrete Ill-Posed Problems, Numerical Aspect of Linear Inversion*, SIAM (1998).

[3] Curtis R. Vogel, *Computational Methods for Inverse Problem*, SIAM (2002).

[4] Per Christian Hansen, *Discrete Inverse Problems, Insight and Algorithms*, SIAM (2010).

[5] Julianne Chung, James G. Nagy, Ioannis Sechopoulos, *Numerical Algorithms for Polyenergetic Digital Breast Tomosynthesis Reconstruction*, SIAM (2010).

[6] Veronica Mejia Bustamante, James G. Nagy, Steve S. J. Feng, Ioannis Sechopoulos, *Iterative Breast Tomosynthesis Image Reconstruction*, SIAM (2013).

[7] G. Landi, E. Loli Piccolomini, J. G. Nagy, *A Limited Memory BFGS Method for a Nonlinear Inverse Problem in Digital Breast Tomosynthesis*, SIAM (2017).

[8] D. Touati-Ahmed, C. Storey, *Efficient Hybrid Conjugate Gradient Techniques*, J.Optim. Theory Appl., Vol. 64, No. 2 (1990).