A limited memory BFGS method for a nonlinear inverse problem in digital breast tomosynthesis

Published Version:
Landi, G., LOLI PICCOLOMINI, E., Nagy, J.G. (2017). A limited memory BFGS method for a nonlinear inverse problem in digital breast tomosynthesis. INVERSE PROBLEMS, 33(9), 1-21 [10.1088/1361-6420/aa7a20].

Availability:
This version is available at: https://hdl.handle.net/11585/610052 since: 2017-11-06

Published:

DOI: http://doi.org/10.1088/1361-6420/aa7a20

A Limited Memory BFGS method for a nonlinear inverse problem in digital breast tomosynthesis

G. Landi E. Loli Piccolomini J. G. Nagy

November 9, 2016

Abstract

Digital breast tomosynthesis (DBT) is an imaging technique that allows the reconstruction of a pseudo three-dimensional image of the breast from a finite number of low-dose two-dimensional projections obtained by different x-ray tube angles. An issue that is often ignored in DBT is the fact that an x-ray beam is polyenergetic, i.e. it is composed of photons with different levels of energy. The polyenergetic model requires solving a large-scale, nonlinear inverse problem, which is more expensive than the typically used simplified, linear monoenergetic model. However, the polyenergetic model is much less susceptible to beam hardening artifacts, which show up as dark streaks and cupping (i.e., background nonuniformities) in the reconstructed image. In addition, it has been shown that the polyenergetic model can be exploited to obtain additional quantitative information about the material of the object being imaged. In this paper we consider the multimaterial polyenergetic DBT model, and solve the nonlinear inverse problem with a limited memory BFGS quasi-Newton method. Regularization is enforced at each iteration using a diagonally modified approximation of the Hessian matrix, and by truncating the iterations.

1 Introduction

Digital breast tomosynthesis (DBT) is an imaging technique that allows the reconstruction of a pseudo three-dimensional image of the breast from a finite number of low-dose two-dimensional projections obtained by different x-ray tube angles. DBT is becoming increasingly popular in healthcare, since this innovative technology may address some of the long-standing limitations of conventional two-dimensional mammography such as the phenomena of summation and subtraction, potentially responsible for the production of false positive findings and for masking of true positive findings. Since DBT image acquisition allows the breast to be viewed in as a pseudo 3D volume (i.e., as slices), these phenomena are minimized and the accuracy of screening and diagnostic breast imaging is improved by ruling out overlapping structures and facilitating identification of small lesions [19]. Two crucial issues are faced when implementing a reconstruction algorithm for DBT. First, because the angular range over which projections are obtained is very limited (e.g., between 15 and 30 degrees), the mathematical problem is very ill-posed. Secondly, DBT systems generally use high resolution detectors (e.g., 2048×2048), so the transfer and storage of this data can be challenging as far as the time required for image reconstruction.

An issue that is often ignored in DBT, as well as in full computed tomography (CT) applications, is the fact that an x-ray beam is polyenergetic, i.e. it is composed of photons with different levels of energy. Instead, the image formation model usually assumes the x-ray source to be monoenergetic. The polyenergetic model requires solving a nonlinear inverse problem, whereas the monoergetic model results in a much simpler linear inverse problem. However, the unrealistic physical assumption of the monoenergetic model leads to so-called beam hardening artifacts in reconstructed images, which typically contain dark streaks and overall cupping (i.e., background nonuniformities) [1].

Although it is more computationally expensive, previous work has been done to develop algorithms that use the more accurate polyenergetic model of the source x-rays; in the case of CT, see for example [3, 8, 9, 17], and in the limited angle case of tomosynthesis, see [6, 7, 12]. In [7], a polyenergetic tomosynthesis model was described under the assumption of Poisson noise. Iterative techniques, such as gradient descent and Newton-type methods, were used for image reconstruction. In [6], the polyenergetic model was further developed by also taking into account the various materials composing the object. The model allows for computing weight fractions of the linear attenuation coefficients of the materials composing the object and reconstructing density mappings of each material, giving quantitative information about the tissue, which can facilitate clinical diagnoses. In [12], a multimaterial polyenergetic model was proposed which is based on the mass (instead of linear) attenuation coefficients of the materials composing the object and allows for the restoration of pseudo-3D images of the mass attenuation coefficients of each material. Under Gaussian noise assumption, the image reconstruction process was formulated in a nonlinear least squares framework, which was solved using the Levenberg-Marquardt method.

In this work, we consider the multimaterial polyenergetic DBT model based on the linear attenuation coefficients described in [6] and we assume the data noise to have a Gaussian distribution. We remark that the Gaussian noise assumption is not restrictive since the quality of the reconstructed images is reported to be not significantly influenced by the noise model [14]. In [6] a Poisson noise was assumed and the problem was solved by minimizing a likelihood function by a gradient descent method. In that paper, the efficiency of the numerical method was not deeply investigated, since the attention was mainly on the information and quality of the results obtained by the proposed model. As previously mentioned, computing a solution of the optimization problem is challenging because it is large-scale and because the limited angular range of projections causes the problem to be severely ill-posed. Moreover, in a clinical setting it is desirable to have a quite good reconstruction in as short a time as possible.

The aim of this work is to propose an efficient method for the solution of the nonlinear inverse problem formulated as a nonlinear least squares minimization. There are many numerical approaches to solve nonlinear least squares problems, ranging from gradient-like methods to Newton-like methods, and there are well known trade-offs to consider when deciding on which algorithm to use. For example, a gradient approach uses only first order information, and hence each iteration is less expensive, but typically exhibits much slower convergence than a Newton-like method. On the other hand, a Newton-like method uses second order information, which can result in much faster convergence than a gradient method, but each iteration is more expensive as it requires the solution of a linear system involving the Hessian (or its approximation). This additional cost per iteration can be very expensive, and often prevents from applying Newton-type methods to large-scale problems. In order to address the high dimensionality of the tomosynthesis problem together with the need of a good solution in a short time, we propose to use a Newton-like method with a limited memory BFGS strategy. The BFGS strategy[1] requires, at each iteration, the solution of a linear system whose coefficient matrix is a low-cost Hessian approximation, and the limited memory version, referred to as LBFGS, uses curvature information only from the most recent iterations [16]. Due to the ill-posedness of the continuous problem, the approximation of the Hessian matrix is ill-conditioned. To avoid magnifying errors when solving linear systems with this ill-conditioned matrix, which is required at each LBFGS iteration, we use a diagonal modification of the Hessian approximation. Two strategies are proposed and compared for the numerical inversion of this diagonally modified linear system. Regularization is also enforced by truncating the iterations of the LBFGS algorithm.

The rest of the paper is organized as follows. In Section 2, we formulate polyenergetic DBT image reconstruction as a nonlinear least squares problem. In Section 3 we describe the diagonally modified limited memory BFGS method and analyze its convergence property. The results of numerical experiments are presented in Section 4 in order to illustrate the potential of the proposed approach. Finally, conclusions are given in Section 5.

2 The mathematical model

We present in this section the nonlinear model for the image formation process considered in this paper. The nonlinear model is described in [6] and it is derived from the Beer's law [10] taking into account both the polyenergetic nature of the x-ray beam and the various materials composing the object. Before proceeding with the model description, let us fix our notation. We assume that the object being imaged is made up of a small number of known materials; for example, in the case of breast imaging, this would include adipose and glandular tissue, and possibly calcium and/or iodine (which, when injected into the blood stream can highlight blood vessels and tissues). We denote the number of materials by N_m. In particular, we suppose that the linear attenuation coefficients $\mu_{j,e}$ of each voxel j at the energy level e is approximated as a linear combination of individual materials with *unknown* weights $w_{j,m}$:

$$\mu_{j,e} = \sum_{m=1}^{N_m} w_{j,m} c_{m,e} \tag{1}$$

where $c_{m,e}$ is the linear attenuation coefficient of the m-th material at e-th energy. In addition, we denote the number of voxels in the discretized 3D volume (object) by N_v, the number of pixels in each 2D projection by N_p, the number of discrete energies by N_e, and the number of angles by N_θ. We use $a^\theta_{i,j}$ to denote the length of the x-ray beam through the voxel j, incident onto the

[1] The acronym BFGS is based on the names of Broyden, Fletcher, Goldfarb and Shannon who are credited with discovering the method.

pixel i in the detector, with source at angle θ, and b_i^θ to denote the measured projection value in the i-th pixel, with source at angle θ. Finally, we indicate by s_e the energy fluence, which is a product of the x-ray energy with the number of incident photons at that energy.

Using this notation, and following [6], we have the following discrete image formation model

$$b_i^\theta = \sum_{e=1}^{N_e} s_e \exp\left(-\sum_{j=1}^{N_v} a_{i,j}^\theta \sum_{m=1}^{N_m} w_{j,m} c_{m,e} \right) + \eta_i^\theta, \quad \begin{cases} i = 1, \dots N_p, \\ \theta = 1, \dots N_\theta, \end{cases} \tag{2}$$

where η_i^θ represents noise measured at the detector, which can include x-ray scatter and electronic noise and which is assumed to follow a Gaussian distribution. We also assume that the weight fractions for each voxel add to 1, i.e:

$$\sum_{m=1}^{N_m} w_{j,m} = 1, \quad j = 1, \dots N_v. \tag{3}$$

We refer the reader to [6, 7] for a more detailed presentation of the multi-material model (2) for polyenergetic digital breast tomosynthesis.

The equations given in (2) can be rewritten in matrix-vector form as

$$\mathbf{b} = \exp(-\mathbf{A}\mathbf{W}\mathbf{C}^T)\mathbf{s} + \boldsymbol{\eta} \tag{4}$$

where $\mathbf{C}$ is an $N_e \times N_m$ matrix with entries $c_{e,m}$, $\mathbf{W}$ is an $N_v \times N_m$ matrix with entries $w_{j,m}$, and $\mathbf{s}$ is a vector with entries s_e. The matrix $\mathbf{A}$ and the vectors $\mathbf{b}$ and $\boldsymbol{\eta}$ are defined in terms of blocks that depend on the angle θ. Specifically, if we define the $N_p \times N_v$ matrices $\mathbf{A}^{(\theta)}$ with entries $a_{i,j}^\theta$ and the N_p vectors $\mathbf{b}^{(\theta)}$ and $\boldsymbol{\eta}^{(\theta)}$ with, respectively, entries b_i^θ and η_i^θ, then

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}^{(1)} \\ \mathbf{A}^{(2)} \\ \vdots \\ \mathbf{A}^{(N_\theta)} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}^{(1)} \\ \mathbf{b}^{(2)} \\ \vdots \\ \mathbf{b}^{(N_\theta)} \end{bmatrix}, \quad \boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{\eta}^{(1)} \\ \boldsymbol{\eta}^{(2)} \\ \vdots \\ \boldsymbol{\eta}^{(N_\theta)} \end{bmatrix}.$$

We remark that, usually, the noise vector $\boldsymbol{\eta}$ is not available while both $\mathbf{s}$ and $\mathbf{C}$ are known since an accurate estimate of the x-ray energy distribution can be obtained by using well-known spectra models and the linear attenuation coefficients $c_{e,m}$ can be derived by taking x-ray transmission measurements of objects with known dimension, density and material composition. Moreover, although we can use modestly sized problems for numerical simulations, in actual clinical applications typical values are $N_v = n_x n_y n_z$ with $n_x = 1280$, $n_y = 2048$ and $n_z = 50$, $N_p = n_x n_y$, $N_e \leq 30$ and $N_m \leq 4$. Such large scale problems require very efficient algorithms, and efficient implementation of computational kernels, such as operations to perform matrix-vector multiplication with $\mathbf{A}$ (i.e., projection) and with $\mathbf{A}^T$ (i.e., back projection).

Assuming model (4), DBT reconstruction is a large-scale nonlinear inverse problem where the weight fractions matrix $\mathbf{W}$ has to be determined, given the ray tracing matrix $\mathbf{A}$, the noisy projections vector $\mathbf{b}$, the x-ray energies vector $\mathbf{s}$ and the linear attenuation coefficients matrix $\mathbf{C}$.

3 Numerical solution of the inverse problem

Under a Gaussian noise assumption, the maximum likelihood solution to (4) is obtained by solving the nonlinear least squares problem

$$\min_{\mathbf{W}} \frac{1}{2}\|\mathbf{b} - \exp(-\mathbf{A}\mathbf{W}\mathbf{C}^T)\mathbf{s}\|^2. \tag{5}$$

By imposing the constraint (3) on problem (5) and by employing variable substitution for

$$w_{j,1} = 1 - \sum_{m=2}^{N_m} w_{j,m}, \tag{6}$$

we obtain the nonlinear least squares problem

$$\min_{\mathbf{X}} f(\mathbf{X}) = \frac{1}{2}\|\mathbf{r}(\mathbf{X})\|^2 \tag{7}$$

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

$$\mathbf{X} = [\mathbf{w}_2 \mid \mathbf{w}_3 \mid \ \dots \ \mid \mathbf{w}_{N_m} \ ]. \tag{8}$$

If $\mathbf{w}_m$ and $\mathbf{x}_m$ denote the m-th column of $\mathbf{W}$ and $\mathbf{X}$, respectively, then the residual $\mathbf{r}(\mathbf{X})$ has the form

$$\mathbf{r}(\mathbf{X}) = \mathbf{b} - \exp\left(\mathbf{A}\left[\ 1 - \sum_{m=1}^{N_m-1}\mathbf{x}_m \ \middle|\ \mathbf{X}\ \right]\mathbf{C}^T\right)\mathbf{s}. \tag{9}$$

If we define the $N_e \times (N_m - 1)$ matrix $\widehat{\mathbf{C}}$ as

$$\widehat{\mathbf{C}} = [\mathbf{c}_2 - \mathbf{c}_1 \mid \mathbf{c}_3 - \mathbf{c}_1 \mid \ \dots \ \mid \mathbf{c}_{N_m} - \mathbf{c}_1 \ ] \tag{10}$$

where $\mathbf{c}_\ell$ denotes the ℓ-th column of $\mathbf{C}$, then equation (9) can be written component wise as

$$\begin{aligned} r_i^\theta &= b_i^\theta - \sum_{e=1}^{N_e} s_e \exp\left(-\sum_{j=1}^{N_v} a_{i,j}^\theta \left(c_{1,e} + \sum_{m=2}^{N_m} x_{j,m}(c_{m,e} - c_{1,e})\right)\right) \quad (11) \\ &= b_i^\theta - \sum_{e=1}^{N_e} s_e \exp\left(-\sum_{j=1}^{N_v} a_{i,j}^\theta \left(c_{1,e} + \sum_{m=1}^{N_m-1} x_{j,m}\widehat{c}_{m,e}\right)\right). \quad (12) \end{aligned}$$

3.1 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 (7). The gradient of $f(\mathbf{X})$ is expressed in terms of the Jacobian $\mathbf{J}(\mathbf{X})$ of $\mathbf{r}(\mathbf{X})$:

$$\nabla f(\mathbf{X}) = \mathbf{J}(\mathbf{X})^T\mathbf{r}(\mathbf{X}) \tag{13}$$

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

$$\begin{aligned}
\{\mathbf{J}(\mathbf{X})\}_{i,\theta,j,m} &= \frac{\partial}{\partial x_{j,m}} r_i^\theta \\
&= \frac{\partial}{\partial x_{j,m}} \sum_{e=1}^{N_e} s_e \exp\left(-\sum_{\ell=1}^{N_v} a_{i,\ell}^\theta \sum_{n=1}^{N_m-1} x_{\ell,n}\widehat{c}_{n,e} \right) \\
&= \sum_{e=1}^{N_e} s_e \exp\left(-\sum_{\ell=1}^{N_v} a_{i,\ell}^\theta \sum_{n=1}^{N_m-1} x_{\ell,n}\widehat{c}_{n,e} \right) \left(-a_{i,j}^\theta \widehat{c}_{m,e}\right) \\
&= -a_{i,j}^\theta \sum_{e=1}^{N_e} \exp\left(-\sum_{\ell=1}^{N_v} a_{i,\ell}^\theta \sum_{n=1}^{N_m-1} x_{\ell,n}\widehat{c}_{n,e} \right) s_e \widehat{c}_{m,e} \\
&= -a_{i,j}^\theta \sum_{e=1}^{N_e} \left\{ \exp\left(-\mathbf{A}\mathbf{X}\widehat{\mathbf{C}}\right) \right\}_{i,\theta,e} (\mathbf{s} \odot \widehat{\mathbf{c}}_m)_e
\end{aligned} \tag{14}$$

for $i = 1, \ldots, N_p$, $\theta = 1, \ldots, N_\theta$, $j = 1, \ldots, N_v$ and $m = 1, \ldots, N_m$, where $\odot$ denotes component wise multiplication. Equation (14) shows that the (ℓ, n)-th entry of $\mathbf{J}(\mathbf{X})$, for $\ell = 1, \ldots, N_pN_\theta$ and $n = 1, \ldots, N_vN_m$, is the scalar product of the ℓ-th row of $\exp(-\mathbf{A}\mathbf{X}\widehat{\mathbf{C}})$ and the vector $-a_{i,j}^\theta(\mathbf{s} \odot \widehat{\mathbf{c}}_m)$.

However, even if it is possible to explicitly calculate the first partial derivatives of $\mathbf{r}(\mathbf{X})$ making up $\mathbf{J}(\mathbf{X})$, computer memory may be insufficient to store the full Jacobian matrix. In order to avoid the storage of $\mathbf{J}(\mathbf{X})$, we compute directly the n-th partial derivative of $f(\mathbf{X})$, $n = 1, \ldots, N_vN_m$, as the scalar product of the n-th column of $\mathbf{J}(\mathbf{X})$ by the residual vector $\mathbf{r}(\mathbf{X})$. This results in a considerable reduction of the memory demand for the gradient $\nabla f(\mathbf{X})$. However, since we do not have access to $\mathbf{J}(\mathbf{X})$, in order to solve (7) we can not use standard algorithms for nonlinear least squares problems such as the Gauss-Newton and Levenberg-Marquardt methods. Therefore, we propose to use two alternative algorithms requiring only the gradient of the objective function to be supplied at each iteration.

3.2 Newton method with diagonally modified Hessian matrix

Before describing the proposed algorithms, let us briefly recall that Newton's method uses the iterative updating strategy

$$\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} + \alpha_k \mathbf{p}^{(k)} \tag{15}$$

where $\mathbf{p}^{(k)}$ is the step direction, obtained by solving the linear system

$$\nabla^2 f(\mathbf{X}^{(k)})\mathbf{p} = -\nabla f(\mathbf{X}^{(k)}), \tag{16}$$

and α_k is the step-length. For the line search, we consider a standard approach where α_k is initially set to 1, and a back tracking strategy is employed until α_k satisfies the Armijo condition [16].

We remark that because the tomosynthesis image reconstruction problem is ill-posed the Hessian matrix $\nabla^2 f(\mathbf{X}^{(k)})$ can be severely ill-conditioned. In order to stabilize the computation for the step direction, we employ Lavrentyev method [2] replacing (16) with the regularized linear system

$$(\nabla^2 f(\mathbf{X}^{(k)}) + \mu_k \mathbf{I})\mathbf{p} = -\nabla f(\mathbf{X}^{(k)}) \tag{17}$$

where the positive parameter μ_k controls the amount of regularization. In addition, to reduce the computational cost of working with the Hessian, we replace the Hessian with an approximation based on the limited memory BFGS (LBFGS) formula [15, 13].

In the following subsections, we first discuss an efficient strategy for the choice of the regularization parameter μ_k. Then we describe two algorithms, which use the LBFGS updating strategy, to efficiently compute the step direction. Although the LBFGS strategy results in slower convergence compared to Newton's method, the advantage of LBFGS over a full Newton algorithm is that each iteration can be computed more efficiently and substantially reduces the storage requirements.

3.3 Choice of the regularization parameter μ_k

The performance of modified Newton method (17) depends on the value of the regularization parameter μ_k both in terms of reconstruction quality and method efficiency. In our implementation, at each iteration, we use the following criterion for the choice of the regularization parameter:

$$\mu_k = \max\left(\mu_{\text{inf}}, \min\left(\mu_{\text{sup}}, \|\nabla f(\mathbf{X}^{(k)})\|\right)\right) \tag{18}$$

where $\mu_0 = \mu_{\text{sup}}$. The regularization parameter μ_k is then bounded both above and below:

$$\mu_{\text{inf}} \leq \mu_k \leq \mu_{\text{sup}}. \tag{19}$$

In section 4 we report the results of some tests for tuning the values of μ_{inf} and μ_{sup}. We have done extensive numerical experiments on tomosynthesis applications, and have found that criterion (18) is quite robust in preventing ill-conditioning and speeding up convergence when the gradient norms are small.

3.4 Algorithm 1: a quasi-Newton method with diagonally modified LBFGS matrix

In the first algorithm, hereinafter referred to as Algorithm 1, we consider the LBFGS updating formula to obtain a positive definite approximation $\mathbf{B}^{(k)}$ to the Hessian matrix $\nabla^2 f(\mathbf{X}^{(k)})$. The LBFGS method uses information only from the most recent iterations in order to obtain $\mathbf{B}^{(k)}$. Specifically, at iteration k, the method uses only at most a fixed number M of vectors $\mathbf{d}_i$ and $\mathbf{y}_i$ defined as

$$\mathbf{d}_i = \mathbf{X}^{(i+1)} - \mathbf{X}^{(i)}, \quad \mathbf{y}_i = \nabla f(\mathbf{X}^{(i+1)}) - \nabla f(\mathbf{X}^{(i)}), \quad i = k-M, \dots, k-1 \tag{20}$$

to construct $\mathbf{B}^{(k)}$ as follows

$$\mathbf{B}^{(k)} = \mathbf{B}_0^{(k)} + \sum_{i=k-M}^{k-1} \left(\mathbf{z}_i \mathbf{z}_i^T - \mathbf{a}_i \mathbf{a}_i^T\right) \tag{21}$$

where

$$\mathbf{a}_i = \frac{\mathbf{B}_i^{(k)}\mathbf{d}_i}{\sqrt{\mathbf{d}_i^T\mathbf{B}_i^{(k)}\mathbf{d}_i}}, \quad \mathbf{z}_i = \frac{\mathbf{y}_i}{\sqrt{\mathbf{y}_i^T\mathbf{d}_i}}, \quad i = k-M, \dots, k-1 \tag{22}$$

$$\mathbf{B}^{(0)} = \gamma_k^{-1}\mathbf{I} \tag{23}$$

with γ_k a positive constant. The number M of vector pairs $\{\mathbf{d}_i, \mathbf{y}_i\}$ stored at each iteration is usually very small.

The general iteration of Algorithm 1 has therefore the form

$$\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} + \alpha_k\mathbf{p}^{(k)} \tag{24}$$

where the step $\mathbf{p}^{(k)}$ solves the diagonally modified LBFGS system

$$(\mathbf{B}^{(k)} + \mu_k\mathbf{I})\mathbf{p} = -\nabla f(\mathbf{X}^{(k)}). \tag{25}$$

Algorithm 1 can therefore be viewed as a *quasi-Newton method with diagonally modified LBFGS matrix.* Following the convergence analysis for the LBFGS method [13], global convergence of Algorithm 1 can be proved with μ_k satisfying (18).

Proposition 3.1. *Assume that the objective function f is twice continuously differentiable and that there exist positive constants c_1 and c_2 such that*

$$c_1\|\mathbf{z}\|^2 \leq \mathbf{z}^T\nabla^2 f(\mathbf{X})\mathbf{z} \leq c_2\|\mathbf{z}\|^2 \tag{26}$$

for all $\mathbf{z}$ and for all $\mathbf{X} \in \{\mathbf{X} \mid f(\mathbf{X}) \leq f(\mathbf{X}^{(0)})\}$ and assume that the matrices $\mathbf{B}_0^{(k)}$ are chosen so that $\{\|\mathbf{B}_0^{(k)}\|\}$ and $\{\|(\mathbf{B}_0^{(k)})^{-1}\|\}$ are bounded. Then, the eigenvalues of $\mathbf{B}^{(k)} + \mu_k\mathbf{I}$ are bounded above and bounded away from zero. That is, there are positive scalars d_1 and d_2 such that

$$d_1\|\mathbf{z}\|^2 \leq \mathbf{z}^T(\mathbf{B}^{(k)} + \mu_k\mathbf{I})\mathbf{z} \leq d_2\|\mathbf{z}\|^2 \quad \textit{for all } k. \tag{27}$$

Proof. The proof follows [13, Theorem 6.1] very closely. If we define

$$\overline{G}^{(k)} = \int_0^1 \nabla^2 f(\mathbf{X}^{(k)} + \tau\mathbf{d}_k)d\tau \tag{28}$$

then we have

$$\mathbf{y}_k = \overline{G}^{(k)}\mathbf{d}_k. \tag{29}$$

From (26) and (29), we obtain

$$c_1\|\mathbf{d}_k\|^2 \leq \mathbf{y}_k^T\mathbf{d}_k \leq c_2\|\mathbf{d}_k\|^2 \tag{30}$$

and

$$\frac{\|\mathbf{y}_k\|^2}{\mathbf{y}_k^T\mathbf{d}_k} = \frac{\mathbf{d}_k^T(\overline{G}^{(k)})^2\mathbf{d}_k}{\mathbf{d}_k^T\overline{G}^{(k)}\mathbf{d}_k} \leq c_2. \tag{31}$$

From the definition of $\mathbf{B}^{(k)}$, the boundedness of $\{\|\mathbf{B}_0^{(k)}\|\}$ and (31), we have

$$\mathrm{tr}(\mathbf{B}^{(k)}) \leq \mathrm{tr}(\mathbf{B}_0^{(k)}) + \sum_{i=k-M}^{k-1}\frac{\|\mathbf{y}_i\|^2}{\mathbf{y}_i^T\mathbf{d}_i} \leq \mathrm{tr}(\mathbf{B}_0^{(k)}) + c_2M \leq L \tag{32}$$

where $\mathrm{tr}(\mathbf{B}^{(k)})$ denotes the trace of $\mathbf{B}^{(k)}$ and L is some positive constant. Then, it follows that a constant $L > 0$ exists such that the largest eigenvalue of $\mathbf{B}^{(k)}$ is less than L. We have that

$$\mu_{\inf}\|\mathbf{z}\|^2 \leq \mathbf{z}^T(\mathbf{B}^{(k)} + \mu_k\mathbf{I})\mathbf{z} \leq (L + \mu_{\sup})\|\mathbf{z}\|^2. \tag{33}$$

The proof follows by setting $d_1 = \mu_{\inf}$ and $d_2 = L + \mu_{\sup}$. □

Since the positive definite matrices $\mathbf{B}^{(k)}+\mu_k\mathbf{I}$ satisfy the bounded eigenvalues condition (27), the sequence of directions $\{\mathbf{p}^{(k)}\}$ is gradient related to $\{\mathbf{X}^{(k)}\}$ [4, pp. 35] and [4, Proposition 1.2.1] guarantees that $\{\mathbf{X}^{(k)}\}$ converges to a stationary point.

The computation of the step $\mathbf{p}^{(k)}$ requires the inversion of the diagonally modified LBFGS matrix $\mathbf{B}^{(k)} + \mu_k\mathbf{I}$ in (17). In [11], an efficient and effective algorithm, based on the Sherman-Morrison-Woodbury formula, is presented for the recursive computation of $(\mathbf{B}^{(k)} + \mu_k\mathbf{I})^{-1}$ with complexity $O(M^2 N_v N_m)$.

The following theorem shows that $(\mathbf{B}^{(k)} + \mu_k\mathbf{I})^{-1}$ can be computed recursively, if $\gamma_k\mu_k$ is bounded away from zero.

Theorem 3.1 ([11])**.** *Let $\gamma_k > 0$ and $\mu_k > 0$ with $\gamma_k\mu_k > \epsilon$ for some $\epsilon > 0$. Let $\mathbf{G} = (\mathbf{B}^{(0)} + \mu_k\mathbf{I}) = (\gamma_k^{-1} + \mu_k)\mathbf{I}$ and let $\mathbf{H} = \sum_{i=0}^{2\widetilde{M}-1}\mathbf{E}_i$ where*

$$\begin{aligned}
\mathbf{E}_0 &= -\mathbf{a}_0\mathbf{a}_0^T, & \mathbf{E}_1 &= -\mathbf{z}_0\mathbf{z}_0^T, \\
&\vdots & &\vdots \\
\mathbf{E}_{2\widetilde{M}-2} &= -\mathbf{a}_{\widetilde{M}-1}\mathbf{a}_{\widetilde{M}-1}^T, & \mathbf{E}_{2\widetilde{M}-1} &= -\mathbf{z}_{\widetilde{M}-1}\mathbf{z}_{\widetilde{M}-1}^T.
\end{aligned}$$

Then, $(\mathbf{B}^{(k)} + \mu_k\mathbf{I})^{-1} = (\mathbf{G} + \mathbf{H})^{-1}$ is given by

$$(\mathbf{G} + \mathbf{H})^{-1} = \mathbf{C}_{M-1}^{-1} - v_{M-1}\mathbf{C}_{M-1}^{-1}\mathbf{E}_{M-1}\mathbf{C}_{M-1}^{-1} \tag{34}$$

together with

$$\mathbf{C}_{k+1}^{-1} = \mathbf{C}_k^{-1} - v_k\mathbf{C}_k^{-1}\mathbf{E}_k\mathbf{C}_k^{-1}, \quad v_k = \frac{1}{trace(\mathbf{C}_k^{-1}\mathbf{E}_k)}. \tag{35}$$

For a proof and an exhaustive description of the recursive formula, we refer readers to [11].

3.5 Algorithm 2: a quasi-Newton method with LBFGS update of the diagonally modified Hessian matrix

In the second algorithm, hereinafter referred to as Algorithm 2, we consider the LBFGS update formula to obtain a positive definite approximation $\widetilde{\mathbf{B}}^{(k)}$ to the diagonally modified Hessian matrix $\nabla^2 f(\mathbf{X}^{(k)}) + \mu_k I$.

The general iteration of Algorithm 2 has therefore the form

$$\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} + \alpha_k\mathbf{p}^{(k)} \tag{36}$$

where the step $\mathbf{p}^{(k)}$ solves the LBFGS system

$$\widetilde{\mathbf{B}}^{(k)}\mathbf{p} = -\nabla f(\mathbf{X}^{(k)}) \tag{37}$$

and $\widetilde{\mathbf{B}}^{(k)}$ is the LBFGS approximation to the Hessian of the function $f(\mathbf{X}) + \frac{\mu_k}{2}\|\mathbf{X}\|^2$ at $\mathbf{X}^{(k)}$.

Algorithm 2 can therefore be viewed as a *quasi-Newton method with LBFGS update of the diagonally modified Hessian matrix.* As for Algorithm 1, if the positive definite matrices $\widetilde{\mathbf{B}}^{(k)}$ satisfy the bounded eigenvalues condition [4, Proposition 1.2.1], the limit of iteration (36) is a stationary point.

Proposition 3.2. *Assume that the objective function f is twice continuously differentiable and that there exist positive constants c_1 and c_2 such that*

$$c_1\|\mathbf{z}\|^2 \le \mathbf{z}^T\nabla^2 f(\mathbf{X})\mathbf{z} \le c_2\|\mathbf{z}\|^2 \tag{38}$$

for all $\mathbf{z}$ and for all $\mathbf{X} \in \{\mathbf{X} \mid f(\mathbf{X}) \le f(\mathbf{X}^{(0)})\}$ and assume that the matrices $\widetilde{\mathbf{B}}_0^{(k)}$ are chosen so that $\{\|\widetilde{\mathbf{B}}_0^{(k)}\|\}$ and $\{\|(\widetilde{\mathbf{B}}_0^{(k)})^{-1}\|\}$ are bounded. Then, the eigenvalues of $\widetilde{\mathbf{B}}^{(k)}$ are bounded above and bounded away from zero. That is, there are positive scalars d_1 and d_2 such that

$$d_1\|\mathbf{z}\|^2 \le \mathbf{z}^T\widetilde{\mathbf{B}}^{(k)}\mathbf{z} \le d_2\|\mathbf{z}\|^2 \quad \textit{for all } k. \tag{39}$$

Proof. Following the proof of Proposition 3.1, we can show that a positive constant L_1 exists such that the largest eigenvalue of $\widetilde{\mathbf{B}}^{(k)}$ is less than L_1. Since the largest eigenvalue of $\widetilde{\mathbf{B}}^{(k)}$ is less than L_1, if we consider the expression for the determinant

$$\det\left(\widetilde{\mathbf{B}}^{(k+1)}\right) = \det\left(\widetilde{\mathbf{B}}_0^{(k)}\right)\prod_{i=k-M}^{k-1}\frac{y_i^T s_i}{s_i^T\widetilde{\mathbf{B}}^{(i)}s_i},$$

we have, using the boundness of $\{\|(\widetilde{\mathbf{B}}_0^{(k)})^{-1}\|\}$, that

$$\det\left(\widetilde{\mathbf{B}}^{(k+1)}\right) \ge \det\left(\widetilde{\mathbf{B}}_0^{(k)}\right)\left(\frac{c_1}{L}\right)^M \ge L_2$$

for some positive constant L_2 (cfr. [13, Theorem 6.1]). Let now $\widetilde{\lambda}_{\min}^{(k+1)}$ and $\widetilde{\lambda}_{\max}^{(k+1)}$ be, respectively, the smallest and largest eigenvalues of $\widetilde{\mathbf{B}}^{(k+1)}$. Thus

$$\widetilde{\lambda}_{\min}^{(k+1)}(\widetilde{\lambda}_{\max}^{(k+1)})^{N_vN_m-1} \ge \det\left(\widetilde{\mathbf{B}}^{(k+1)}\right) \ge L_2$$

and

$$\widetilde{\lambda}_{\min}^{(k+1)} \ge \frac{L_2}{(\widetilde{\lambda}_{\max}^{(k+1)})^{N_vN_m-1}} \ge \frac{M}{L^{n-1}}$$

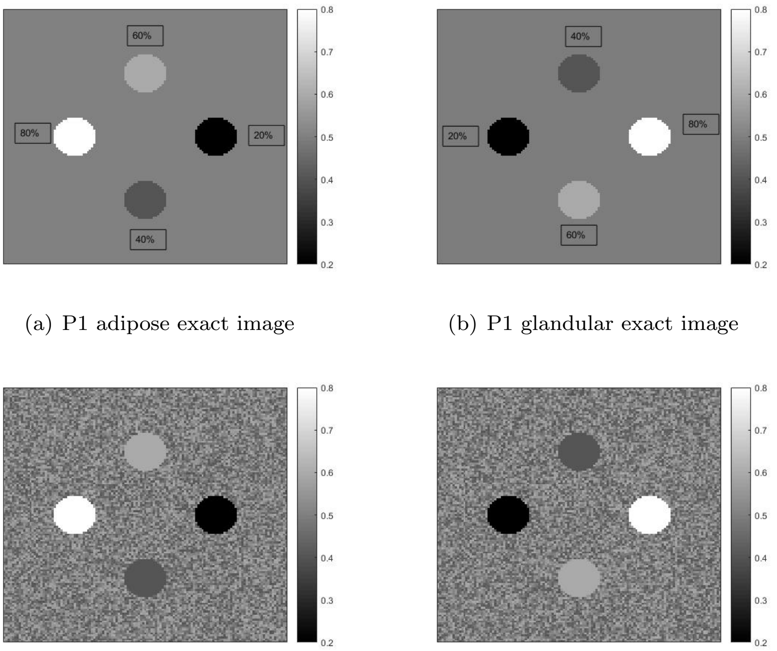
The proof follows by setting $d_1 = \frac{L_2}{L_1^{N_vN_m-1}}$ and $d_2 = L_1$. □

The two loop recursive formula described in [16] can be used to determine the inverse of $\widetilde{\mathbf{B}}^{(k)}$ and efficiently compute the step $\mathbf{p}^{(k)}$ as

$$\mathbf{p}^{(k)} = -(\widetilde{\mathbf{B}}^{(k)})^{-1}\nabla f(\mathbf{X}^{(k)}). \tag{40}$$

4 Numerical results

In this section, we present the results obtained on a simulated breast imaging problem, using MATLAB R2016b on a workstation with two 230 Ghz Intel processors equipped with 132 Gb of RAM. The experiments have been performed on two 3D digital phantoms constituted of glandular and adipose tissue ($N_m = 2$). The P1 phantom has size $N_v = 129 \times 129 \times 7$ and it has an homogeneous background made of a mixture of 50% adipose and 50% glandular tissue and four spheres with different percentages of the two materials (20%, 40%, 60% and 80%). We can visually represent the weights $w_{j,m}$ of the adipose tissue (cf. Figure 1(a)) and glandular tissue (Figure 1(b)). The P2 phantom differs from P1 only in the background, which has a pointwise variable concentration of adipose and glandular tissue (central slices are displayed). Figures 2 and 3 show the seven layers of the adipose and glandular tissue, respectively. The weights $w_{j,m}, j = 1, \dots N_v, m = 1, 2$, have been randomly generated in the interval [0.4,0.6].



(a) P1 adipose exact image

(b) P1 glandular exact image

(c) P2 adipose exact image

(d) P2 glandular exact image

Figure 1: Central slices of the exact images of P1 phantom (top row) and of P2 phantom (bottom row).

The simulated projections are obtained as:

$$\mathbf{b} = \exp(-\mathbf{A}\mathbf{W}\mathbf{C}^T)\mathbf{s} + \boldsymbol{\eta}$$

where the exact weights matrix $\mathbf{W} = [\mathbf{W}_1 \ \mathbf{W}_2]$, the vector $\mathbf{s}$ of the simulated x-ray energy levels ($N_e = 37$ different levels of energy were considered from 10 keV to 28 keV, in 0.5 keV steps) and the linear attenuation coefficients matrix

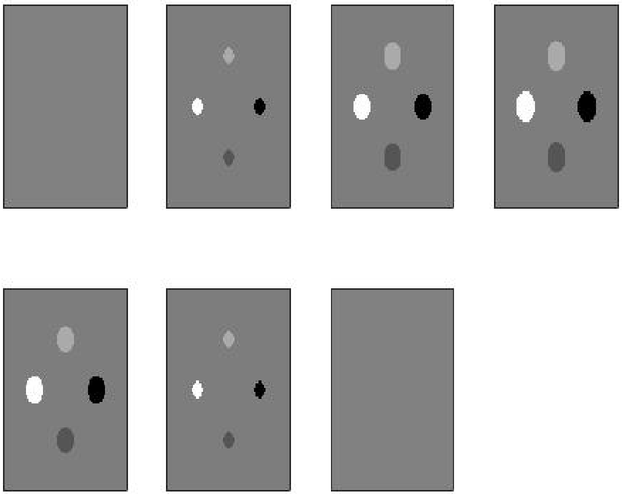


Figure 2: P1 adipose weights for slices $1 \dots 7$.

$\mathbf{C}$ are given. The matrix $\mathbf{A}$ is the projection ray tracing matrix obtained with the Siddon algorithm [18] from a geometry with $N_\theta = 15$ angles in the range $[-17, 17]$ degrees and $\boldsymbol{\eta}$ is the vector simulating Gaussian white noise of relative level

$$nl = \frac{\|\boldsymbol{\eta}\|_2}{\|\boldsymbol{b} - \boldsymbol{\eta}\|_2}.$$

The results obtained by the proposed algorithms are evaluated by measuring the relative error:

$$Er = \frac{\|\mathbf{W} - \widetilde{\mathbf{W}}\|_F}{\|\mathbf{W}\|_F}$$

where $\mathbf{W}$ is the exact weights matrix, $\widetilde{\mathbf{W}}$ is the reconstructed one and the norm is the Frobenius norm.

In order to get the *best* possible results obtained by the methods in a reasonable time, the algorithms are stopped at the iteration k as soon as one of the following conditions is reached:

- the semiconvergence condition, i.e.:

  $$Er_k \leq Er_{k+1}$$

  where Er_k is the relative error at the iteration k;

- $k \leq k_{max}$ (maximum number of iterations reached).

  In our experiments, $k_{max} = 50$ for the LBFGS methods.

The remainder of this section is divided into two parts. In the first part, we present some tests on a reduced size P1 phantom, in order to tune some parameters with many fast simulations; in the second part, we use this information in tests on the full P1 and P2 phantom data.

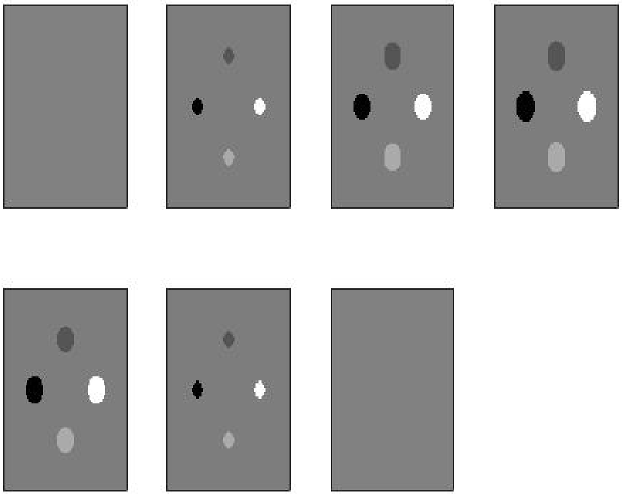


Figure 3: P1 glandular weights for slices $1 \dots 7$.

JN COMMENT: I separated these parts into formal subsections, because I think there is a lot of information in them. But if you prefer to not use subsections, I don't have a very strong opinion about it.

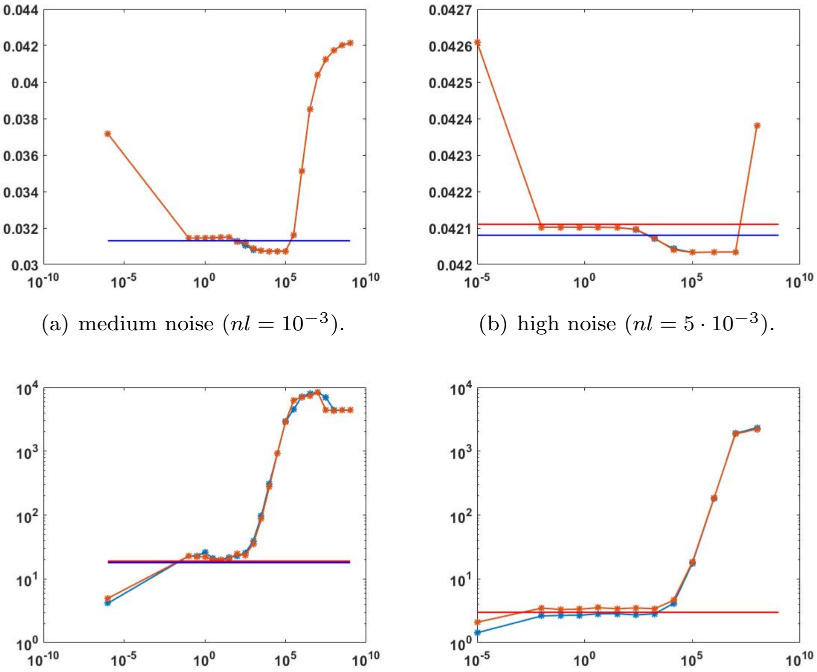
4.1 Analysis of the sensitivity of the algorithms to the parameter μ and comparison with other methods.

In this subsection we present tests performed on a small P1 phantom of size $N_v = 31 \times 31 \times 7$, $N_p = 31 \times 31$ and $N_\theta = 15$. We added noise of different levels $5 \cdot 10^{-4} \leq nl \leq 5 \cdot 10^{-3}$ to the exact projections $\mathbf{b}$.

In order to analize the sensitivity of the proposed LBFGS algorithms with respect to μ, in Figures 4(a) and 4(b) we plot the relative error values obtained with the LBFGS methods when $nl = 10^{-3}$ and $nl = 5 \cdot 10^{-3}$ as a function of μ varying with logarithmic distribution in the interval $[10^{-6}, 10^{9}]$ and $[10^{-5}, 10^{9}]$, respectively. The blue lines represent LBFGS1 (i.e., Algorithm 1) while the red lines are related to LBFGS2 (i.e., Algorithm 2). The plots have a convex shape, with a flat minimum around a nonzero value of μ that depends on the level of noise on the data. This means that regularization is necessary, showing the ill-conditioning of the problem. In Figure 4(c) and 4(d) we plot the computational times in seconds versus μ. From these plots we can deduce that when the value of μ is small the solution is very noisy and the iterations terminate due to semiconvergence after few iterations. On the other hand, when the value of μ is large the methods perform more iterations.

On this small size phantom, the two proposed implementations of the LBFGS strategy produce comparable results, both in terms of precision and computational cost.

In each subfigure, the horizontal (dashed) lines have a y-coordinate corresponding to the relative error (Figures 4(a) and 4(b)) and computational times (Figures 4(c) and 4(d)) obtained by choosing μ with the automatic rule (18), where μ_{inf} and μ_{sup} are heuristically fixed from the previous plots. They confirm that the proposed rule to automatically choose the value of the parameter μ gives excellent results, since we obtain lower errors in shorter times than with the best heuristic choices of a fixed value of μ. Moreover, the boundary values μ_{inf} and μ_{sup} can be easily fixed: for all the experimets $\mu_{\text{inf}} = 10^{-1}$ is a good value, while μ_{sup} is set to 10^2 or 10^3 depending on the noise intensity.



(a) medium noise ($nl = 10^{-3}$).

(b) high noise ($nl = 5 \cdot 10^{-3}$).

(c) medium noise ($nl = 10^{-3}$).

(d) high noise ($nl = 5 \cdot 10^{-3}$).

Figure 4: Results obtained with the LBFGS1 (blue line) and LBFGS2 (red line) methods with μ varying. In the top row the relative errors and in the bottom row the computational times versus μ.

We now consider numerical tests with the aim of showing that the greater efficiency of the LBFGS methods in terms of memory requirements and computational times with respect to other known numerical methods for the solution of nonlinear least squares problems is not at the expense of the precision in the results. Specifically, we compare the proposed diagonally modified LBFGS algorithms with the Levenberg-Marquardt (LM) method and an accelerated gradient descent (GR) method.

The LM method is a Gauss-Newton like method [16] and we consider here a variable step-length implementation whose general iteration is defined by:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$$

and the direction $\mathbf{p}_k$ is the solution of the linear system

$$(\mathbf{J}_k^T \mathbf{J}_k + \mu_k I)\mathbf{p} = -\mathbf{J}_k^T \mathbf{r}_k$$

where μ_k is chosen with the rule (18) and the Jacobian matrix $\mathbf{J}$ is computed by (14). The step-length is determined by the backtracking Armijo procedure with initial unit step-length. Since the LM method requires the computation and storage of the Jacobian matrix at each iteration, it cannot readily be used in real applications because the memory requirements can become prohibitive, and solving the Jacobian system can become too time consuming.

The accelerated GR method is a gradient method with a Barzilai-Borwein based rule for the choice of the steplength. The reader can refer to [5] for more details about the algorithm (in the present work we use the GR method [5] without the projection onto the positive orthant).

In Table (1) we show the results obtained by the LBFGS1, LBFGS2, LM and GR methods in terms of relative error values (column 3), number k of iterations (column 4) and computational time in seconds (column 5). In these experiments we choose μ_k with the automatic rule (18), where the values $\mu_{\inf}$ and $\mu_{\sup}$ are reported in the first column between square brackets.

nl($[\mu_{\inf}, \mu_{\sup}]$)	meth	erel	k	time
$5 \cdot 10^{-4}$ $[10^{-1}, 10^{2}]$	LBFGS1	0.0246	53	43
	LBFGS2	0.0244	61	46
	LM	0.0236	4	62
	GR	0.0238	1004	858
10^{-3} $[10^{-1}, 10^{2}]$	LBFGS1	0.0313	23	18
	LBFGS2	0.0314	25	19
	LM	0.0306	3	49
	GR	0.0307	190	163
$2 \cdot 10^{-3}$ $[10^{-1}, 10^{3}]$	LBFGS1	0.0385	9	7
	LBFGS2	0.0378	13	10
	LM	0.0370	3	49
	GR	0.0373	41	35
$5 \cdot 10^{-3}$ $[10^{-1}, 10^{3}]$	LBFGS1	0.0420	3	3
	LBFGS2	0.0421	2	3
	LM	0.0420	1	25
	GR	0.0440	4	4

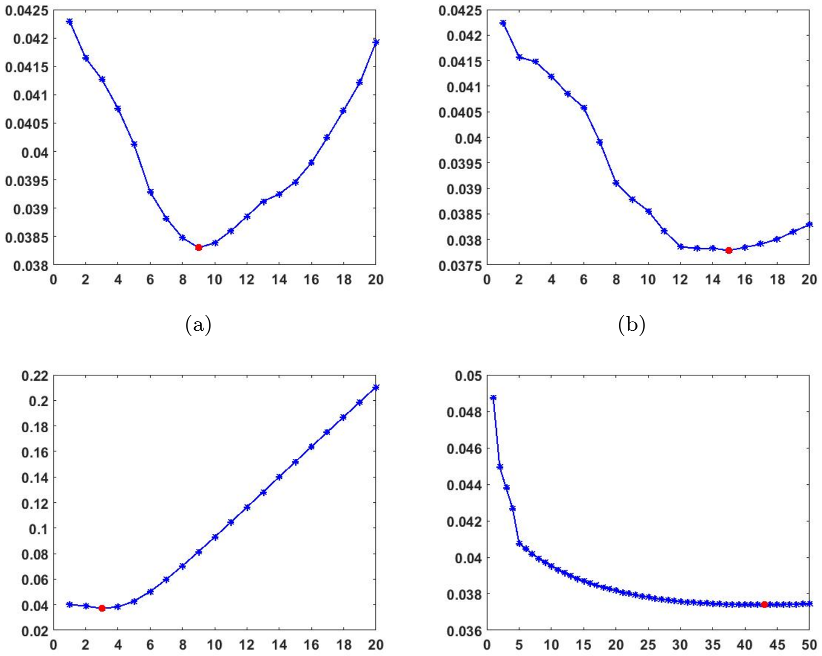
Table 1: Results on the small phantom obtained with the different methods with four levels of noise with μ varying in $[\mu_{\inf}, \mu_{\sup}]$.

The table shows that the four methods have comparable errors, but they differ in terms of computational time required. It is immediate to see that the GR method is the slowest one; only in the case of $nl = 5 \cdot 10^{-3}$ the GR is as fast as LM and LBFGS methods, but semiconvergence occurs in only a few iterations, and the recovered image is of poorer quality.

Finally, Figure 5 plots the relative errors of the four considered methods as a function of the computational time in the case of noise with $nl = 2 \cdot 10^{-3}$.

These plots show the semiconvergence behavior of the algorithms due to the ill-conditioning of the problem. It is evident that it is easier to find a suitable stopping iteration for the LBFGS methods, whose semiconvergence curve is smooth, than for the GR method. ???

JN QUESTION: Figure 5 is a nice plot, but is it possible to put these all on the same axes? For example, if you use the same initial guess for each algorithm, then include as first data point the relative error for the initial guess. Then all plots will start at the same point.



(a)

(b)

(c)

(d)

Figure 5: Semiconvergence behavior of LBFGS1 (a), LBFGS2 (b), LM (c) and GR (d) algorithms on the small phantom ($nl = 2 \cdot 10^{-3}$).

4.2 Full P1 and P2 phantom reconstructions

The following experiments are performed on the P1 and P2 phantoms of size $N_v = 129 \times 129 \times 7$, $N_p = 129 \times 129$ and $N_\theta = 15$. We report here the results obtained in the case of noise with level $nl = 2 \cdot 10^{-3}$, with the parameter μ chosen with the rule (18).

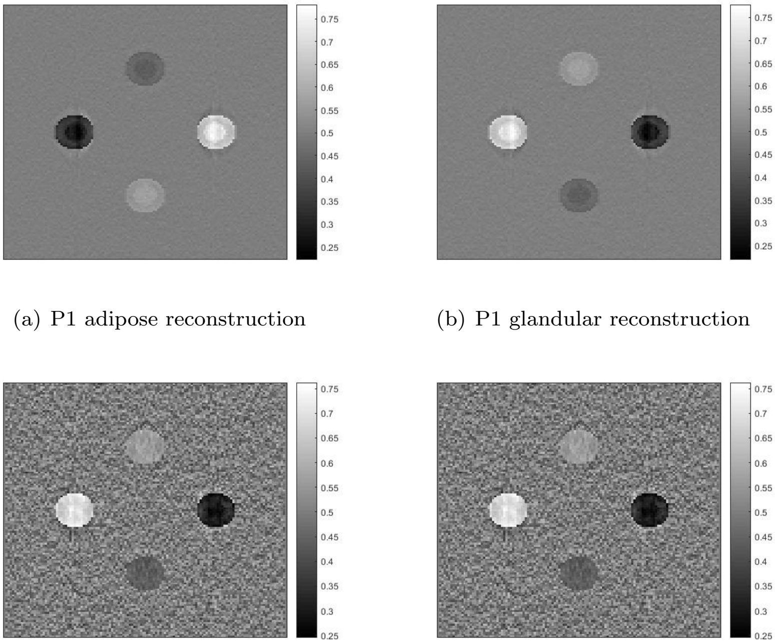
When the problem size is so large, it is impossible to store the Jacobian matrix in the LM method. Hence, only the LBFGS1, LBFGS2 and GR methods are involved in these experiments.

In Table 2 we show the results in terms of relative error, number of iterations and computational time. Since the the reconstructed images for all the three considered methods at their best are very similar, we show in Figure 6 the central slice of the reconstructions obtained with the LBFGS1 method for the

P1 and P2 phantoms. In both the spectral images, we can clearly distinguish the different concentrations of the materials inside the objects.

In Figures 7 and 8 we plot line profiles using pixels from row 64 of the adipose images in the P1 and P2 reconstructions, respectively. In each of these figures, the plot shown in (a) displays exact solution as a continuous blue line, and the LBFGS1 reconstruction in the red dashed line, and (b) displays the error between the exact and LBFGS1 reconstructed profiles.

Finally in Figure 9 we display the relative errors versus the iterations for P1 (Figure 9 (a)) and P2 (Figure 9 (b)). From the figures it is clear that in the first iterations the LBFGS algorithms approximate faster the exact solution than the GR method and can obtain good results with separate materials in few iterations. In clinical applications, where computational time is important, the LBFGS methods can be efficiently used to obtain spectral images in Digital Breast Tomosynthesis.



(a) P1 adipose reconstruction

(b) P1 glandular reconstruction

(c) P2 adipose reconstruction

(d) P2 glandular reconstruction

Figure 6: Central slices of the reconstructed images of P1 (upper row) and P2 (lower row) phantoms with the LBFGS1 method

5 Conclusions

In this paper we have presented a numerical method for the solution of an inverse nonlinear least squares problem obtained from the discretization of a polyenergetic multimaterial model in Digital Breast Tomosynthesis. In order to address the high dimensionality of the problem together with the need for

Phantom($[\mu_{\text{inf}}, \mu_{\text{sup}}]$)	`meth`	`erel`	`k`	`time`
	LBFGS1	0.0529	10	$9.9 \cdot 10^3$
P1	LBFGS2	0.0530	6	$7.26 \cdot 10^3$
$[10^{-1}, 10^2]$	GR	0.0526	17	7.6410^3
	LBFGS1	0.0676	40	$9.3 \cdot 10^3$
P2	LBFGS2	0.0676	50	$1.1 \cdot 10^4$
$[10^{-1}, 10^3]$	GR	0.0670	208	$5.76 \cdot 10^4$

Table 2: Results obtained on the P1 and P2 phantoms in the case of noise of level $nl = 2 \cdot 10^{-3}$.

a fast algorithm we proposed a limited memory BFGS quasi-Newton method, where the Hessian approximation is diagonally modified in order to enforce regularization. We considered two different implementation strategies.

The tests executed on small and medium size digital phantoms show good performance of the methods in obtaining good quality reconstructed images in very few iterations, with moderate extra-storage requirement for the Hessian approximation.

While our focus has been on Digital Breast Tomosynthesis, the proposed methods are not restricted to this particular application, and they can be used to solve other large-scale, ill-posed least squares problems, when the Hessian approximation is too large to be explicitly constructed.

References

[1] R. A. Brooks adn G. Di Chiro. Beam hardening in x-ray reconstructive tomography. *Phys. Med. Biol.*, 2(3):390–398, 1976.

[2] A. Bakushinsky and A. Goncharsky. *Ill-Posed Problems: Theory and Applications.* Kluwer, 1994.

[3] R. F. Barber, E. Y. Sidky, T. G. Schmidt, and X. Pan. An algorithm for constrained one-step inversion of spectral CT data. *Phys. Med. Biol.*, 61:3784–3818, 2016.

[4] D. Bertsekas. *Nonlinear Programming.* Athena Scientific, (2nd Edition), 1999.

[5] S. Bonettini, R. Zanella, and L. Zanni. A scaled gradient projection method for constrained image deblurring. *Inv. Probl.*, 25(1), 2009.

[6] V. Bustamante, J. Nagy, S. Feng, and I. Sechopoulus. Iterative breast tomosynthesis image reconstruction. *SIAM J. Scient. Comput.*, 135(5):S192–S208, 2013.

[7] J. Chung, J. Nagy, and I. Sechopoulus. Numerical algorithms for polyenergetic digital breast tomosynthesis reconstruction. *SIAM J. Imag. Sci.*, 3(1):133, 2010.

[8] I. A. Elbakri and J. A. Fessler. Statistical image reconstruction for polyenergetic x-ray computed tomography. *IEEE Trans. Med. Imaging*, 21:89–99, 2002.

[9] I. A. Elbakri and J. A. Fessler. Segmentation-free statistical image reconstruction for polyenergetic x-ray computed tomography with experimental validation. *Phys. Med. Biol.*, 48:2453–2477, 2003.

[10] C. L. Epstein. *Introduction to the Mathematics of Medical Imaging, Second Edition*. SIAM, Philadelphia, second edition, 2007.

[11] J. Erway and R. F. Marcia. Limited-memory BFGS systems with diagonal updates. *Linear Algebra Appl.*, 437(1):333–344, 2012.

[12] G. Landi, E. Loli Piccolomini, and J. G. Nagy. Numerical solution of a nonlinear least squares problem in digital breast tomosynthesis. *Journal of Physics: Conference Series*, 657(1):012006, 2015.

[13] D.C. Liu and J. Nocedal. On the limited memory method for large scale optimization. *Math. Programming B*, 45(3):503–528, 1989.

[14] Y. Lu, H.P. Chan, J. Wei, M. Goodsitt, P.L. Carson, L. Hadjiiski, A. Schmitz, J.W. Eberhard, and B.E. Claus. A comparative study of limited angle cone-beam reconstruction methods for breats tomosynthesis. *Med. Phys.*, 33:3781–3795, 2006.

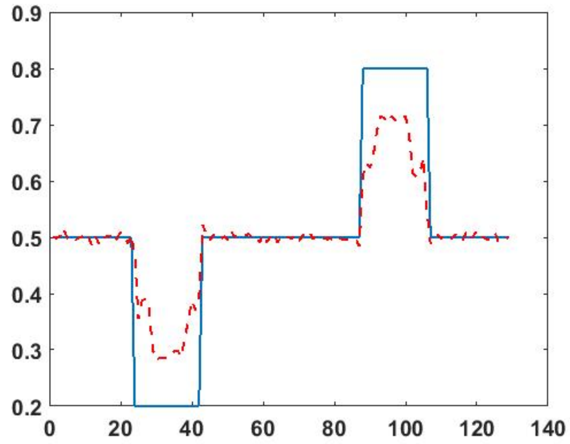
[15] J. Nocedal. Updating quasi-Newton matrices with limited storage. *Math. Comp.*, 35:773–782, 1980.

[16] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 2006.

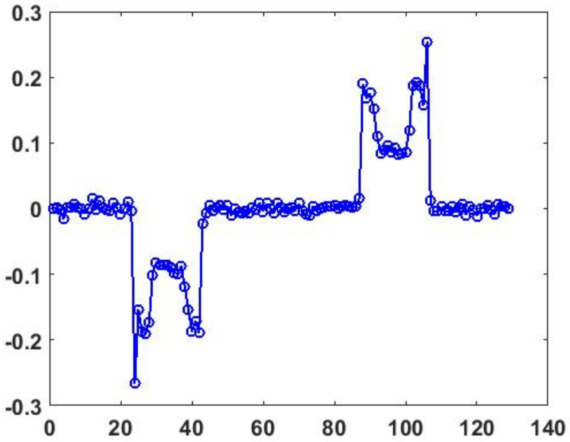
[17] J. A. O'Sullivan and J. Benac. Alternating minimization algorithms for transmission tomography. *IEEE Trans. Med. Imaging*, 26:283–297, 2007.

[18] R. L. Siddon. Fast calculation of the exact radiological path for a three-dimensional ct array. *Med. Phys.*, 12:252–255, 1985.

[19] P. Skaane, A.I. Bandos, Gullien, et al. Comparison of digital mammography alone and digital mammography plus tomosynthesis in a population-based screening program. *Radiology*, 267(1):47–56, 2013.

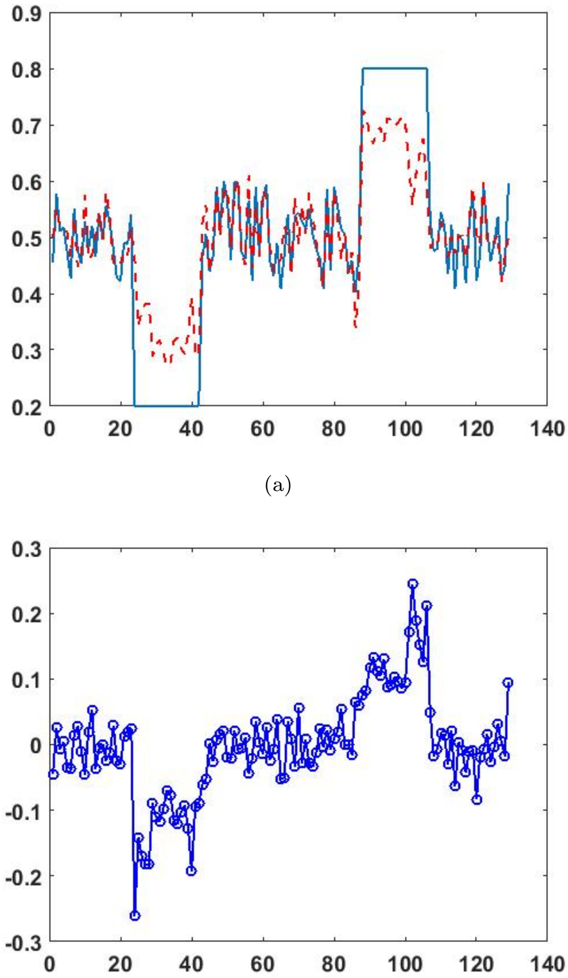


(a)



(b)

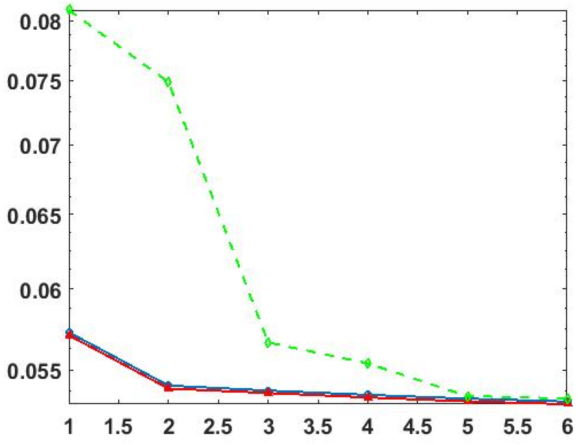
Figure 7: Plot of line 64 of P1 adipose ($[\mu_{\text{inf}}, \mu_{\text{sup}}]$ as in Table 1). (a) The exact signal (blue continuous line) and the LBFGS1 reconstruction (red dashed line);(b) the errors obtained with LBFGS1.



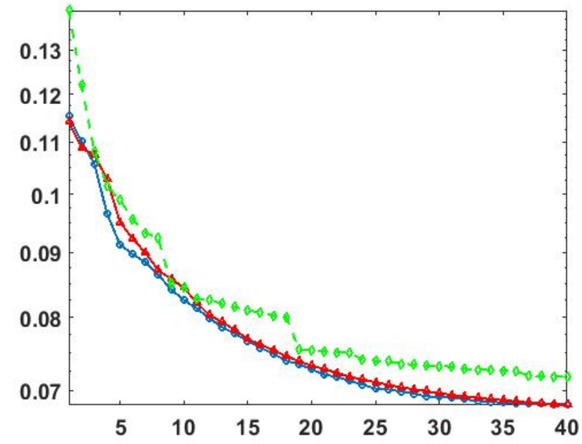
(a)

(b)

Figure 8: Plot of line 64 of P2 adipose (($[\mu_{\text{inf}}, \mu_{\text{sup}}]$ as in Table 1)). (a) The exact signal (blue continuous line) and the LBFGS1 reconstruction (red dashed line); (b) the errors obtained with LBFGS1.



(a) P1 errors



(b) P2 errors

Figure 9: Errors versus iterations for P1 and P2 phantoms obtained with the LBFGS1 (continuous blue line), LBFGS2 (continuous red line) and GR (dotted green line) methods.