Regularization opportunities for the Gauss-Newton and BFGS algorithms: application to the diffuse optical tomography problem

Fabien Dubot[a,b], Yann Favennec[b,*], Benoit Rousseau[b], Daniel Rousse[a]

[a] *Industrial research chair in technologies of energy and energy efficiency (T3E), École de Technologie Supérieure, 1100, rue Notre-Dame Ouest, Montréal (Qc), Canada H3C 1K3*

[b] *Laboratoire de Thermocinétique de Nantes (LTN), UMR CNRS 6607, Rue Christian Pauc, BP 50609, 44306 Nantes Cedex 3, France*

Abstract

In optical tomography, the optical properties of the medium under investigation are obtained through the solution of an inverse problem where some light is injected on a boundary and the measurement of intensity is performed elsewhere on boundary. The properties of interest are the scattering and the absorption coefficients, denoted $\sigma(\boldsymbol{x})$ and $\kappa(\boldsymbol{x})$ where $\boldsymbol{x}$ stands for the space variable in a bounded region. Such an inverse problem is solved through optimization with the help of gradient-type methods. Since it is well known that such inverse problems are ill-posed, regularization is required. This paper compares three distinct regularization strategies for two very different kinds of optimization algorithms, namely the Gauss-Newton and the L-BFGS algorithms for the two-dimensional diffuse approximation model of radiation propagation. The conclusion is that the use of Tikhonov regularization is absolutely compulsary when considering optimization algorithms that rely on matrix inversion. Moreover, combining this regularization with appropriate parameterization enhances the quality of reconstructions. However, the Tikhonov regularization does not bring much improvements when considering optimizers that do not rely on matrix inversion, while the combination of an appropriate parameterization of the control space and the use of Sobolev gradients brings much more improvements. Finally, a parallel BFGS algorithm is proposed and applied to the three-dimensional diffuse approximation model.

Keywords: inverse problem, regularization, optical tomography, adjoint state method, BFGS, Gauss-Newton

1. Introduction

In Optical Tomography (OT), the volumetric optical properties of the medium under investigation are obtained through the solution of an inverse problem where some light is injected on a boundary and the measurement of transmitted and/or reflected intensity is performed elsewhere on boundary [1, 2]. The properties of interest are mainly the scattering and the absorption coefficients, denoted $\sigma(\boldsymbol{x})$ and $\kappa(\boldsymbol{x})$ where $\boldsymbol{x}$ is the location in a three-dimensional

*Corresponding author

Email address: Yann.Favennec@univ-nantes.fr (Yann Favennec)

bounded region in general. Such an inverse problem is solved minimizing a $L_2(\partial\mathcal{D})$-based norm between the prediction and the measurement. It is well known that such an inverse problem is ill-posed, thus regularization is mandatory. In this paper, three regularization tools are combined: (1) a proper parameterization of the control space, (2) the ordinary Tikhonov penalization, and, (3) for the Broyden-Fletcher-Goldfarb-Shanno (BFGS) optimizer, the use of the so-called Sobolev-gradient [3]. This latter acts as a pre-conditionner within the optimization algorithm, smoothing the cost function gradient which contains the noise due to experimental set-up.

In the following setting, let φ be the state solution of some partial differential equation, $\mathcal{J}(\varphi)$ a cost function that explicitly measures the misfit $\upsilon(\varphi - \breve{\varphi})$ between the prediction φ and the corresponding measurement $\breve{\varphi}$, and $j(\gamma)$ the reduced cost function implicitly expressed in terms of the parameter function(s) $\gamma(\boldsymbol{x}) = (\kappa, \sigma)(\boldsymbol{x}) \in \mathcal{K} \times \mathcal{S} \subset [L_2(\mathcal{D})]^2$ which is actually the function to be minimized. One thus sets by definition $j(\gamma) := \mathcal{J}(\varphi)$ and searches:

$$\bar{\gamma} = \arg \min_{\gamma \in \mathcal{K} \times \mathcal{S}} j(\gamma) \tag{1}$$

In order to solve this optimization problem, the forward model to predict φ is solved with finite elements. The "continuous" control variable γ is also approximated through a finite element projection so that its dimension becomes finite, making possible the use of optimization algorithms. Choosing a single finite element space for both coefficients with $\Lambda \subset \mathcal{K}$ and $\Lambda \subset \mathcal{S}$ shall later on simplify the calculations, though it is not compulsary to impose such a condition. Due to the relatively high dimension of the control parameter space in our applications (from 1 000 up to 10 000 and more), gradient-type optimization algorithms are chosen rather than gradient-free algorithms. To the best knowledge of the authors, the nonlinear conjugate-gradients and relative algorithms give quite fair reconstructions in space-distributed parameters recovering, but the BFGS algorithm and its limited memory version, i.e. the L-BFGS [4], yield much better reconstructions at lower computational price. The Gauss-Newton (GN) and relative algorithms (Levenberg-Marquardt, etc.) are much more used in practice but they need quite a lot of regularizations in order to stabilize the ill-posed inverse problem character. On the contrary, the L-BFGS is much less sensitive to the ill-posed character; regularization may be not compulsory, although its use may however enhance regularity of the solutions.

From [3], several regularization opportunities can be dealt with, namely the choice of the norm involved in the cost function definition, the choice of the inner product involved in the adjoint identity, and the choice of the inner product involved in the cost function gradient extraction.

In this paper, the authors rather introduce the following regularization opportunities:

– the discrete version of the control parameter space in a suitable way is chosen. Indeed, it has been shown elsewhere [5] that the use of a much coarser mesh than the one used for the states and adjoints can enhance regularity in the reconsruction. Such use of distinct meshes for states and control parameters, also known as "dual meshing" has been used efficiently by [6]. Moreover, added to dual-meshing, choosing an appropriate functional space or finite element parameterization for the control parameters may also affect the regularity of the solutions. Following previous works performed by some of the authors [7], the piecewise linear continuous element is chosen instead of the most common piecewise constant functions per element;

– an alternate inner product when extracting the cost function gradient is selected, considering especially the Sobolev gradient that dramatically enhances regularity in the reconstruction. Depending on the situations (discussed in the numerical results section of the paper), the Sobolev gradients may either smooth on the cost function gradient which somehow inherits the noise present within data or, in particular situations, precondition the optimization problem [3] yielding to allow for solutions that would not be possible otherwise;

– a Tikhonov penalization is added to the cost function, which then becomes $j^+(\gamma) = \mathcal{J}(\varphi) + \mathcal{J}^+(\gamma)$. Though this regularization is not compulsory while using first-order gradient-type methods (contrary to GN-type algorithms which yield to non-invertible matrix systems if no regularization is used), its use may also enhance regularity on the solution.

The paper, which constitutes an extension of a previous work [8] is organized as follows. Section 2 shortly summarizes the models used in the area of optical tomography, namely the ones based on the full radiative transfer equation and the related diffuse approximation model. The optical tomography problem is then written in terms of an optimization problem. Section 3 brings mathematical tools (1) on property adimensionalization, (2) on specific complex and real inner products, (3) on directional derivatives of some function for order one and two, and (4) on finite element parameterization. Section 4 briefly presents the way the cost function gradient and approximate hessian are computed in terms of state sensitivities. Section 5 gives the relationship of the cost function gradient when the adjoint-state method is used; a more detailed derivation of such gradient is given in Appendix A. Section 6 then presents the results for a two-dimensional bounded domain. Specially, both the Gauss-Newton and the BFGS algorithms are concerned with the use of several regularization techniques: (1) the ordinary Tikhonov penalization, (2) an appropriate parameterization for the control space, and (3) the Sobolev inner product when extracting the cost function gradient. The main conclusions are as follows. The efficiency of Gauss-Newton algorithms does rely on the use of regularization. On the other hand, the BFGS algorithm can be used without any regularization while the use of some may enhance regularity of the solution. Overall, the BFGS algorithm provides better reconstructions than those provided by the Gauss-Newton method at the cost of a lower computational effort. These results somehow corroborate some recent results from [9]. These results also clearly show that reconstructions are made possible through the use of a suitable parameterization of the control variables and Sobolev gradients. In addition, these two regularization techniques have cumulative effects in the sense that the best reconstructions are obtained using both together. Next, section 7 exploits main results from section 6 applying them to a three-dimensional reconstruction. This section also briefly presents some strategies to parallelize the code in order to make it computationally efficient. This paper then recalls the main conclusions and gives extension to future work.

2. Problem statement

The forward model commonly used in OT is the Radiative Transfer Equation (RTE) written in the frequency domain, which consists in the following integro-differential equation [10]:

$$\left[\vec{\Omega}\cdot\nabla+\frac{2\pi\mathrm{i}\nu}{c}+\kappa(\boldsymbol{x})+\sigma(\boldsymbol{x})\right]\Phi(\boldsymbol{x},\vec{\Omega})=\frac{\sigma(\boldsymbol{x})}{4\pi}\int_{4\pi}f(\vec{\Omega}',\vec{\Omega})\Phi(\boldsymbol{x},\vec{\Omega}')\,\mathrm{d}\vec{\Omega}' \tag{2}$$

where $\vec{\Omega}$ is the propagation direction of the prescribed radiation, Φ is the radiative intensity per unit solid angle per unit area at the spatial location $\boldsymbol{x}$ in the direction $\vec{\Omega}$, and κ and σ are the absorption and scattering coefficients that depend on $\boldsymbol{x}$ and that have to be retrieved from measurements. $f(\vec{\Omega}',\vec{\Omega})$ is the scattering phase function that is often described by the Henyey-Greenstein phase function [10].

The RTE is an integro-differential equation and thus heavy computation is needed to get accurate solutions. The computation may become highly time consuming when dealing with inverse problems for which solutions require large numbers of iterations. Alternatively, the Diffuse Approximation (DA) provides a simple equation governing the evolution of the photon density, say φ, within the medium. The photon density is linked to the radiative intensity through the formula: $\varphi(\boldsymbol{x})=\int_{4\pi}\Phi(\boldsymbol{x},\vec{\Omega})\,\mathrm{d}\vec{\Omega}$. This approximation is used when Φ is assumed to be quasi-isotropic everywhere in the medium. A detailed description to get the DA model from the general RTE is given in [1]. It is well accepted that the DA model is a reasonably good approximation of the RTE as soon as the medium under consideration is highly diffusing and satisfies $0\ll\kappa\ll\sigma$. The DA model is written as, with $\varphi:\ \mathcal{D}\mapsto\mathbb{C}$:

$$\begin{aligned}&-\nabla\cdot\left[D(\boldsymbol{x})\nabla\varphi(\boldsymbol{x})\right]+\left[\kappa(\boldsymbol{x})+\frac{2\pi\mathrm{i}\nu}{c}\right]\varphi(\boldsymbol{x})=0, &&\forall\boldsymbol{x}\in\mathcal{D}\\&\varphi(\boldsymbol{\zeta})+\frac{A}{2\gamma}D(\boldsymbol{\zeta})\nabla\varphi(\boldsymbol{\zeta})\cdot\boldsymbol{n}=\frac{I}{\gamma}\mathbb{1}_{[\boldsymbol{\zeta}\in\partial\mathcal{D}_s]}(\boldsymbol{\zeta}) &&\forall\boldsymbol{\zeta}\in\partial\mathcal{D}\end{aligned} \tag{3}$$

with $D=(n_{\mathcal{D}}(\kappa+\sigma'))^{-1}$ is the macroscopic scattering coefficient (expressed in m^{-1}), $n_{\mathcal{D}}$ depicts the dimension of $\mathcal{D}$ and κ, σ' are respectively the absorption and reduced scattering coefficients (both expressed in m^{-1}). Remark that the latter coefficient is deduced from the scattering coefficient σ involved in (2) and from the phase funtion f (or rather the anisotropic coefficient) following developments of [1]. Next, the parameter A, which characterizes the reflection at the boundary, involved in the Robin-type boundary condition, is a parameter that can be derived from the Fresnel laws if specular reflection is considered [11] or from experimental set-ups. I denotes the prescribed light intensity, and $\mathbb{1}_{[\cdot]}$ denotes the indicator function. With $\varphi\in\hat{H}^1(\mathcal{D})$ and κ, $\sigma'\in L^\infty(\mathcal{D})$, the existence and unicity of the DA model can be easily demonstrated through the complex version of the Lax-Milgram theorem [12].

Note at this stage that many authors consider a cost function $\mathcal{J}(\varphi):\mathbb{C}\mapsto\mathbb{R}$ but since some gradient-type algorithms shall be introduced later on, and since $\mathcal{J}'(\varphi)$ does not exist, i.e. this function is not holomorphic (the complex derivative of a real function does not exist, except for constant real functions), the differentiable cost function $\mathcal{J}(\varphi_r,\varphi_i):\mathbb{R}\times\mathbb{R}\mapsto\mathbb{R}$ that depends explicitly on both the real and imaginary parts of the density $\varphi=\varphi_r+\mathrm{i}\varphi_i$ is introduced. Taking into account of such issue, with no loss of generality, this yields:

$$j(\gamma):=\mathcal{J}(\varphi_r,\varphi_i):=\oiint_{\partial\mathcal{D}_d}\Upsilon(\varphi_r,\varphi_i)\,\mathrm{d}\boldsymbol{\zeta} \tag{4}$$

with $\oint\!\!\!\!\!\!\int_{\partial\mathcal{D}_d} = \sum_{k=1}^{K} \int_{\partial\mathcal{D}_d}$, K is the number of distinct sources, $\partial\mathcal{D}_d$ represents the sensor location, and Υ is a function being assumed to be quadratic in φ_r and φ_i in order to solve the optimization problem (1).

3. Mathematical settings

The two radiative properties to be retrieved from experiments are indeed different in nature, and their order of magnitude also differ. As a consequence, the cost function gradient parts associated to both optical properties also differ by roughly the same order of magnitude. In order to speed-up the iterative convergence to the local minimum, it is usual to follow [13] performing a scaling on the computed cost function gradient parts such as:

$$\tilde{\nabla}_\kappa j(\gamma) = \chi_\kappa \nabla_\kappa j(\gamma),\ \tilde{\nabla}_{\sigma'} j(\gamma) = \chi_{\sigma'} \nabla_{\sigma'} j(\gamma) \tag{5}$$

with coefficients χ_κ and $\chi_{\sigma'}$ computed before the first iteration with:

$$\chi_\kappa = 0.05 \frac{\|\kappa\|_{L^\infty(\mathcal{D})}}{\|\nabla_\kappa j(\gamma)\|_{L^\infty(\mathcal{D})}},\ \chi_{\sigma'} = 0.05 \frac{\|\sigma'\|_{L^\infty(\mathcal{D})}}{\|\nabla_{\sigma'} j(\gamma)\|_{L^\infty(\mathcal{D})}}. \tag{6}$$

In this paper, the proposed strategy is different in the sense that the scaling is performed at the beginning of the optimization problem on the parameters themselves rather than on the cost function gradient. Choosing an a priori function for each optical properties, one searches parameters that fluctuate about this prior. This adimensionalization leads to recover both $\varkappa(\boldsymbol{x}) = \kappa(\boldsymbol{x})/\kappa_{ap}(\boldsymbol{x})$ and $\varsigma(\boldsymbol{x}) = \sigma'(\boldsymbol{x})/\sigma'_{ap}(\boldsymbol{x})$ in respectively $\mathcal{K}$ and $\mathcal{S}$ and for which magnitude is of order one approximately for both coefficients. Though the use of such assumed-to-be-known coefficients, $\kappa_{ap}(\boldsymbol{x}) > 0$ and $\sigma'_{ap}(\boldsymbol{x}) > 0$ for all $\boldsymbol{x} > 0$ in $\mathcal{D}$, has not been found in literature to the best of our knowledge, this results, in the authors opinion, in a much elegant and proper way for performing the scaling.

Some real and complex inner products are also to be defined before calculations, integrating on the whole domain of interest $\mathcal{D}$ or only on the boundaries $\partial\mathcal{D}_d$ where the cost is integrated:

$$(\gamma, \eta)_{\hat{\mathcal{X}}} := \oint\!\!\!\!\!\!\int_{\partial\mathcal{D}_d} \bar{\gamma}\eta \, \mathrm{d}\boldsymbol{\zeta} \quad ; \quad (\gamma, \eta)_{\hat{\mathcal{Y}}} := \oint\!\!\!\!\!\!\int_{\mathcal{D}} \bar{\gamma}\eta \, \mathrm{d}\boldsymbol{x} \tag{7}$$

with γ and η: $\mathcal{D} \mapsto \mathbb{C}$ and $\bar{\gamma} : \mathcal{D} \mapsto \mathbb{C}$ is the complex conjugate of γ. In the sequel, the related real inner products are written, for simplicity, with no hat, i.e. $(\gamma, \eta)_{\mathcal{X}} := \oint\!\!\!\!\!\!\int_{\partial\mathcal{D}_d} \gamma\eta \, \mathrm{d}\boldsymbol{\zeta}$, $(\gamma, \eta)_{\mathcal{Y}} := \oint\!\!\!\!\!\!\int_{\mathcal{D}} \gamma\eta \, \mathrm{d}\boldsymbol{x}$, with γ and η: $\mathcal{D} \mapsto \mathbb{R}$.

The cost function, eq. (4), is rewritten in terms of the $\hat{\mathcal{X}}$-inner product as $\mathcal{J}(\varphi_r, \varphi_i) = \frac{1}{2}(\upsilon(\varphi), \upsilon(\varphi))_{\hat{\mathcal{X}}}$ where implicitly $\varphi = \varphi_r + \mathrm{i}\varphi_i$ is used such that this cost function is actually written with a real inner product in order to make the differentiation possible. Next, the error function is written as $\upsilon(\varphi) = \frac{\varphi - \breve{\varphi}}{\breve{\varphi}}$ so that all different orders of magnitude are equivalently weighted within the cost function integration process. The cost function to be minimized, eq. (4), thus reads:

$$j(\gamma) = \frac{1}{2}\left(\frac{1}{|\breve{\varphi}|^2}(\varphi - \breve{\varphi}), \varphi - \breve{\varphi}\right)_{\hat{\mathcal{X}}} \tag{8}$$

One also needs to define the directional derivatives $j'(\gamma; \eta)$ and $\varphi'(\gamma; \eta)$ at the point $\gamma \in \Lambda^2$ and towards the direction $\eta \in \Lambda^2$ as [14]:

$$j'(\gamma; \eta) := \lim_{\varepsilon \to 0^+} \frac{j(\gamma + \varepsilon\eta) - j(\gamma)}{\varepsilon} \quad ; \quad \varphi'(\gamma; \eta) := \lim_{\varepsilon \to 0^+} \frac{\varphi(\gamma + \varepsilon\eta) - j(\gamma)}{\varepsilon} \tag{9}$$

and extend this definition to the second order, according to [15], with also $\zeta \in \Lambda^2$:

$$j''(\gamma;\eta,\zeta) := \lim_{\varepsilon\to 0^+} \frac{j'(\gamma+\varepsilon\zeta;\eta) - j'(\gamma;\eta)}{\varepsilon} \tag{10}$$

The cost function being assumed to be twice differentiable at point γ, it can be shown that the directional derivative exists along the direction η and the derivative to gradient and second derivative to Hessian relationships are then:

$$j'(\gamma;\eta) = (\nabla j(\gamma),\eta)_{L_2(\mathcal{D})} \quad ; \quad j''(\gamma;\eta,\zeta) = \left(\nabla^2 j(\gamma)\eta,\zeta\right)_{L_2(\mathcal{D})} \tag{11}$$

In order to use efficient optimization algorithms to solve eq. (1), the control parameter space must be approached in order to be finite. Often, the finite element projection is used, so that one searches $\varkappa$ and ς belonging to $V_h(\mathcal{M}_h,\mathcal{G})$ with $\mathcal{M}_h$ the finite element triangulation dedicated to the control parameter space, and $\mathcal{G}$ the shape functions. In this study, the prescriptions found in [7] are followed in the sense that the linear Lagrangian interpolating functions for $\mathcal{G}$ are chosen. Hence, $V_h(\mathcal{M}_h,\mathcal{G})$ is the function space of piecewise linear continuous functions.

Thus, the finite element projection may be written as:

$$\begin{pmatrix}\kappa\\ \sigma'\end{pmatrix}(\boldsymbol{x}) = \begin{pmatrix}\kappa_{ap}\\ \sigma'_{ap}\end{pmatrix}(\boldsymbol{x}) \odot \sum_{\xi=1}^{\Xi} \psi_\xi(\boldsymbol{x}) \begin{pmatrix}\varkappa\\ \varsigma\end{pmatrix}(\boldsymbol{x}_\xi) \tag{12}$$

with $\xi \in \mathbb{N}_+^*$, $\Xi \in \mathbb{N}_+^*$, and κ_{ap} and σ'_{ap} denote the known "a priori" values for both properties κ and σ', and $\odot$ is the element-wise vector product. Next, $\psi_\xi(\boldsymbol{x})$ denotes the ξ^{th} finite element basis function, and $\Xi < \infty$ is the dimension of the finite element space V_h.

With the dimensionless parameterization (12), the discrete versions of eqs. (11) become:

$$j'(\gamma;\eta) = \eta^t \nabla j(\gamma) \quad ; \quad j''(\gamma;\eta;\zeta) = \eta^t \nabla^2 j(\gamma)\zeta \tag{13}$$

with $\gamma \in \mathbb{R}_+^{2\Xi}$, η and $\zeta \in \mathbb{R}^{2\Xi}$, $\nabla j(\gamma) \in \mathbb{R}^{2\Xi}$ and $\nabla^2 j(\gamma) \in \mathbb{R}^{2\Xi} \times \mathbb{R}^{2\Xi}$. In the sequel, we consider:

$$\nabla j(\gamma) = \begin{pmatrix}\nabla_\varkappa j(\gamma)\\ \nabla_\varsigma j(\gamma)\end{pmatrix} \quad ; \quad \nabla^2 j(\gamma) = \begin{pmatrix}\nabla^2_{\varkappa\varkappa} j(\gamma) & \nabla^2_{\varkappa\varsigma} j(\gamma)\\ \nabla^2_{\varsigma\varkappa} j(\gamma) & \nabla^2_{\varsigma\varsigma} j(\gamma)\end{pmatrix} \tag{14}$$

4. Optimization algorithms

In the field of OT based on the DA model, the Gauss-Newton (GN) method and related methods (such as the Levenberg-Marquardt and others) are mostly used [1, 16]. These optimization methods approximate the cost function Hessian and are therefore quasi-Newton-type optimization methods. More precisely, the cost function Hessian is approximated assuming that second-order state derivatives are negligible in comparison with the product of first-order state derivatives. The Newton matrix system is then written as:

$$\nabla^2 j(\gamma)\delta\gamma = -\nabla j(\gamma) \tag{15}$$

The gradient and approximate Hessian of the cost function are derived as follows:

$$[\nabla j]_{\xi_1} = \mathrm{Re}\left(\frac{1}{|\breve{\varphi}|^2} \left(\varphi - \breve{\varphi}\right), \varphi'(\gamma; \alpha'_{\xi_1}) \right)_{\hat{\mathcal{X}}} \tag{16}$$

$$\left[\nabla^2 j\right]_{\xi_1,\xi_2} = \mathrm{Re}\left(\frac{1}{|\breve{\varphi}|^2} \varphi'(\gamma; \alpha'_{\xi_2}), \varphi'(\gamma; \alpha'_{\xi_1}) \right)_{\hat{\mathcal{X}}} \tag{17}$$

with α'_{ξ_1} and α'_{ξ_2} spanning the whole finite element basis for both $\varkappa$ and ς.

Though some Krylov-based methods can be used to solve iteratively the GN matrix system deriving from Newton's one (15) without computing explicitly (16)-(17) [17, 16], the basic Gauss-Newton and related algorithms rely on the computation of the states derivatives in order to assemble the sensitivity matrices before solving the linear system for one optimization iteration. As it shall be shown in the numerical results dedicated part, the cost associated to the computation of the sensitivity matrix may become prohibitive as soon as the dimension Ξ (see (12)) becomes high. The other drawback is that the matrix involved in eq. (15) may be severely ill-conditionned yielding large errors in $\delta\gamma$ if no specific regularization is used to damp the matrix system.

The alternative consists in using first-order gradient-type algorithms such as the conjugate gradients or other quasi-Newton algorithms. In our experience, the limited memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) [4] is particularly efficient in view of its low computation requirements. Since theses algorithms only rely on the cost function gradient $\nabla j(\gamma)$ at each iteration of the optimization process, the reconstruction cost becomes very low when the adjoint-state method detailed hereafter is used.

5. State derivatives, adjoint states, jacobian and cost gradient

The state derivative $\varphi'(\gamma; \eta)$ involved in the cost function gradient and Hessian is computed differentiating the DA problem (3), for direction $\mathrm{k} \in \Lambda$ related to $\varkappa$:

$$\begin{aligned} &-\nabla\cdot(D\nabla\varphi') + (\kappa + \frac{2\pi i\nu}{c})\varphi' + \nabla\cdot(n_{\mathcal{D}}D^2\kappa_{ap}\mathrm{k}\nabla\varphi) + \kappa_{ap}\mathrm{k}\varphi = 0, \quad && \forall \boldsymbol{x} \in \mathcal{D} \\ &\varphi' + \frac{A}{2\gamma}D\nabla\varphi'\cdot n - \frac{A^c}{2\gamma}n_{\mathcal{D}}D^2\kappa_{ap}\mathrm{k}\nabla\varphi\cdot\boldsymbol{n} = 0 && \forall \boldsymbol{\zeta} \in \partial\mathcal{D} \end{aligned} \tag{18}$$

and for direction $\mathrm{s} \in \Lambda$ related to ς:

$$\begin{aligned} &-\nabla\cdot(D\nabla\varphi') + (\kappa + \frac{2\pi i\nu}{c})\varphi' + \nabla\cdot(n_{\mathcal{D}}D^2\sigma'_{ap}\mathrm{s}\nabla\varphi) = 0, \quad && \forall \boldsymbol{x} \in \mathcal{D} \\ &\varphi' + \frac{A}{2\gamma}D\nabla\varphi'\cdot n - \frac{A^c}{2\gamma}n_{\mathcal{D}}D^2\sigma'_{ap}\mathrm{s}\nabla\varphi\cdot\boldsymbol{n} = 0 && \forall \boldsymbol{\zeta} \in \partial\mathcal{D} \end{aligned} \tag{19}$$

In a fully discretized setting, with the parameterization introduced in eq. (12), one has to perform Ξ runs of both models eqs. (18) and (19) to access the whole cost function gradient expressed in the whole canonical basis for the parameters as well as the whole cost function Hessian. Doing so and at this price, the jacobian matrix, and thus the Gauss-Newton linear system (15) can be solved.

On the other hand, if first-order gradient-type algorithms that only rely on the cost function gradient are dealt with, the adjoint-state method yields to access the cost function gradient at the low cost of only one additional adjoint problem. It can be shown that the directional derivatives of the cost function are equal to:

$$\begin{aligned} j'(\varkappa;\eta) &= \left(\mathrm{Re}\left(\kappa_{ap}\varphi\bar{\varphi}^* - n_{\mathcal{D}}D^2\kappa_{ap}\nabla\varphi\cdot\nabla\bar{\varphi}^*\right),\eta\right)_{\mathcal{Y}} \quad \forall\eta\in L_2(\mathcal{D}) \\ j'(\varsigma;\eta) &= \left(\mathrm{Re}\left(-n_{\mathcal{D}}D^2\sigma'_{ap}\nabla\varphi\cdot\nabla\bar{\varphi}^*\right),\eta\right)_{\mathcal{Y}} \quad \forall\eta\in L_2(\mathcal{D}) \end{aligned} \tag{20}$$

in which φ^* is the solution to the adjoint problem:

$$\begin{aligned} &-\nabla\cdot(D\nabla\varphi^*) + (\kappa - \frac{2\pi i\nu}{c})\varphi^* = 0, && \forall\boldsymbol{x}\in\mathcal{D} \\ &\frac{2\gamma}{A}\varphi^* + D\nabla\varphi^*\cdot\boldsymbol{n} = -\frac{1}{|\check{\varphi}|^2}(\varphi-\check{\varphi})\,\mathbb{1}_{[\zeta\in\partial\mathcal{D}_d]}(\zeta) && \forall\boldsymbol{\zeta}\in\partial\mathcal{D} \end{aligned} \tag{21}$$

Therefore, because the L_2 inner product is used to extract the cost function gradient in eq. (11), one deduces the expressions of the cost function gradients such that:

$$\begin{aligned} \nabla_{\varkappa} j &= \textstyle\sum_{k=1}^{K}\mathrm{Re}\left(\kappa_{ap}\varphi\bar{\varphi}^* - n_{\mathcal{D}}D^2\kappa_{ap}\nabla\varphi\cdot\nabla\bar{\varphi}^*\right) \\ \nabla_{\varsigma} j &= \textstyle\sum_{k=1}^{K}\mathrm{Re}\left(-n_{\mathcal{D}}D^2\sigma'_{ap}\nabla\varphi\cdot\nabla\bar{\varphi}^*\right) \end{aligned} \tag{22}$$

where φ and φ^* are implicitly linked to the source number k. The reader is invited to consult the Appendix A for a demonstration of this result (including the presence of the Tikhonov penalization term, eq. (23), in the cost function).

6. Mixing regularization tools for GN and L-BFGS algorithms

The two-dimensional domain $\mathcal{D}$ is open bounded with a Cassini-type curve such that the largest diameter is approximately equal to 10 cm. 5 sources and 5 sensors 4 mm large are equally spaced on the boundary $\partial\mathcal{D}$.

Two inclusions are embedded within the medium. One is a 0.5 cm radius disk centered at $x = -2$ cm, $y = -0.75$ cm. The other is a 1 cm large square centered on $x = 2$ cm, $y = 0.75$ cm. The optical properties of the background and inclusions are given in Table 1. The sources are prescribed in sequence, all modulated at frequency $\nu = 100$ MHz with intensity $I = 0.01$ W m^{-2}. The dimensionless properties are initialized to unity, i.e. $\varkappa(\boldsymbol{x}) = \varsigma(\boldsymbol{x}) = 1$ $\forall\boldsymbol{x}\in\mathcal{D}$, while the a priori functions κ_{ap} and σ'_{ap} are fixed to the background values, i.e. $\kappa_{ap}(\boldsymbol{x}) = 0.08$ and $\sigma'_{ap}(\boldsymbol{x}) = 20$ $\forall\boldsymbol{x}\in\mathcal{D}$.

When considering the L-BFGS optimizer, provided by the IPOpt software package [18] for this bidimensional numerical simulation, the adjoint-state method is used to compute the cost function gradient at the low cost of single additionnal adjoint problem integration.

On the other hand, when considering the GN optimizer, the direct differentiation method is used. This means that both eqs. (18) and (19) are solved for each degree of freedom $\xi\in\Xi$. The cost associated to this GN optimization method is thus roughly $2\times\Xi$ times the one associated to the BFGS method, for a single iteration of the optimization algorithm. One should note that even in the case where the Hessian is not explicitly computed when solving the GN matrix system [16], the GN optimizer remains computationally expensive compared to the L-BFGS method [9].

Several regularization strategies have then been considered: the well-known Tikhonov method, the appropriate reparameterization of the control space and the use of Sobolev inner products when extracting the cost function gradient. This latter is used when combined with the L-BFGS optimizer. The maximum discrepancy principle [19], which is also considered as a regularization tool, is implicitely used within all algorithms involved herein.

For each optimizer, the optimization stops according to the maximum discrepancy principle (when the cost $\mathcal{J}$ reaches approximately 10^{-7} for a 30 dB random gaussian noise and five 4 mm sensors), and also with a limit of 100 iterations. Note that the scale accompanying the target in Figure 1 is used to build all the reconstruction maps presented hereafter.

	κ (cm^{-1})	σ' (cm^{-1})
Background	0.08	20
Inclusion 1	0.10	10
Inclusion 2	0.06	30

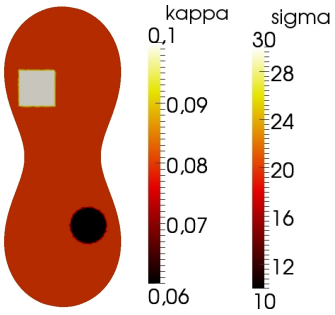


Figure 1: Optical property target maps associated with the iso-value scales for both the absorption and the reduced scattering coefficients.

6.1. GN: mixing Tikhonov penalization and reparameterization

The Tikhonov regularization method, that has been used extensively these last years together with the GN and related algorithms, consists in damping the matrix by adding the penalization $\mathcal{J}^+(\gamma)$ to the cost function $\mathcal{J}(\varphi)$, so that one searches the minimum of $j^+(\gamma) = \mathcal{J}(\varphi) + \mathcal{J}^+(\gamma)$. In the particular case where two property functions are to be estimated, it is convenient to penalize dimensionless properties. The additional term involved in the cost function thus reads:

$$\mathcal{J}^+(\gamma) = \frac{\lambda}{2}\left\|\frac{\kappa-\kappa_{ap}}{\kappa_{ap}}\right\|^2_{L_2(\mathcal{D})} + \frac{\lambda}{2}\left\|\frac{\sigma'-\sigma'_{ap}}{\sigma'_{ap}}\right\|^2_{L_2(\mathcal{D})} \tag{23}$$

After basic calculations, this implies that the quantity $\lambda\frac{\kappa-\kappa_{ap}}{\kappa_{ap}}$ is added to the cost function gradient $\nabla_\varkappa j(\gamma)$, and the quantity $\lambda\frac{\sigma'-\sigma'_{ap}}{\sigma'_{ap}}$ is added to the cost function gradient $\nabla_\varsigma j(\gamma)$. Moreover, for the GN algorithm, the quantity λ is added to $\mathrm{diag}\nabla^2 j(\gamma)$.

Up to here the results found from numerical experiments are in concordance with literature in the sense that the matrix involved in eq. (15) is not invertible without Tikhonov regularization and also for $\lambda \to 0$; on contrary, for $\lambda \gg 0$ the optimization process rapidly stabilizes to functions close to priors $\kappa_{ap}(\boldsymbol{x})$ and $\sigma'_{ap}(\boldsymbol{x})$. Indeed, as it is well-known, that the optimal Tikhonov parameter $\bar{\lambda}$ for GN may be found through the L-curve construction. In the tests presented here, such a curve was built after the first GN iteration and $\bar{\lambda} \approx 10^{-1}$ was found.

Next, the functional space of control variables is chosen such that regularity of the solution is enhanced. To do so, within a finite elements setting, piecewise linear continuous functions are chosen instead of the most common

piecewise constant functions per element. Indeed, it has been shown in [7] that continuous finite elements approximations of the optical properties introduce some implicit regularization on the inversion by smoothing the results. The degree of refinement of the finite elements mesh used for the projection of the control variable is also to be controlled. Indeed, most often, the same finite elements mesh is used for both the states (forward and adjoint) and for the control variables. But, the control variable can be searched within a different functional space. The regularization thus consists in lowering the dimension Ξ of the finite version of the control variable coarsening the related mesh.

Figure 2 presents the numerical reconstructions for the four different meshes. The Tikhonov penalization is always needed for all meshes to damp the GN matrix system and get an invertible matrix. Thus, it can be argued that though the use of Tikonov penalization is of first importance at least to damp the matrix and make it invertible, the reparameterization does not bring much improvement when combined with the GN-type algorithms.

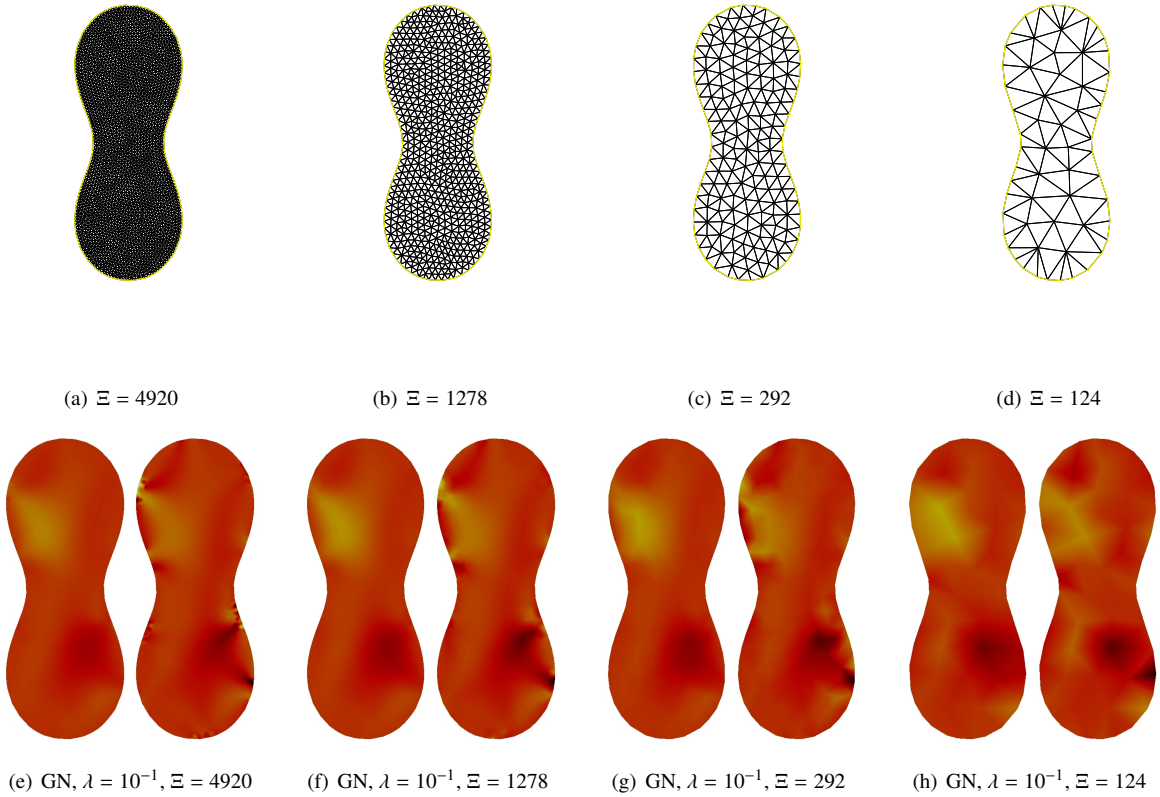
Moreover, and this is a very important result, the optimal Tikhonov parameter determined through the L-curve does not depend on the dimension Ξ. This means that from a practical point of view, and because the GN algorithm is very time consuming, the optimal Tikhonov parameter should be identified based on a coarse mesh for the control space, even though a fine enough mesh is to be used afterwards for better reconstructions.

6.2. L-BFGS: reparameterization

This section is concerned with the reparameterization of the control space. At this stage, there is no Tikhonov penalization, i.e. the Tikhonov parameter λ involved in (23) is set to zero. Figure 3 presents the results with the same four distinct meshes than those used previously. From Figure 3 , it can be concluded that the better results are obtained with the two coarsest meshes. Indeed, the results obtained with the finer mesh (i.e. the mesh that is used to project the forward state and the adjoint state) are highly noisy. This comes from the fact that the noise propagates from the data to the reconstructions through the following path: noisy adjoint states are obtained because the source $\frac{1}{|\breve{\varphi}|^2}(\varphi - \breve{\varphi})$ involved in the adjoint model (21) contains the data that is indeed noisy. Then, noisy cost function gradients (20) are obtained because they depend on noisy adjoint states. Eventually the re-actualization process of the properties contains (at least implicitely) the noise present within the data. In other words, the reparameterization, which consists in projecting onto a coarser mesh the variables involved in eq. (20), has the effect of regularizing implicitly the inverse problem by cutting off the high frequencies inherent to the gradient fluctuations due to noise. The first conclusion is that reparameterization of the control space is somehow a regularization tool in the sense that it enhances regularity of the solutions. Moreover, its use also yields solutions that are much closer to the targets.

6.3. L-BFGS: mixing reparameterization and Tikhonov penalization

The next step consists in using the previous result, i.e. use an appropriate parameterization of the control space, and combine this first regularization tool with the ordinary Tikhonov penalization. Note that the Tikhonov penalization does not play the same role with algorithms that do not rely on matrix inversion. Indeed, when combined with L-BFGS, the first-order Tikhonov penalization, eq. (23), does improve regularity of the solution but in a too diffuse way



(a) $\Xi = 4920$ (b) $\Xi = 1278$ (c) $\Xi = 292$ (d) $\Xi = 124$

(e) GN, $\lambda = 10^{-1}$, $\Xi = 4920$ (f) GN, $\lambda = 10^{-1}$, $\Xi = 1278$ (g) GN, $\lambda = 10^{-1}$, $\Xi = 292$ (h) GN, $\lambda = 10^{-1}$, $\Xi = 124$

Figure 2: Gauss-newton algorithm with Tikhonov parameter $\lambda = 10^{-1}$. Influence of the reconstructions with parameterization of the control space. Top: meshes; bottom: reconstruction of κ and σ' for the four meshes.

such that the reconstructions are too flat (see Figure 4). Though the use of such regularization is of first importance with algorithms that do rely on matrix inversion, its use is not efficient at all with algorithms that do not rely on matrix inversion.

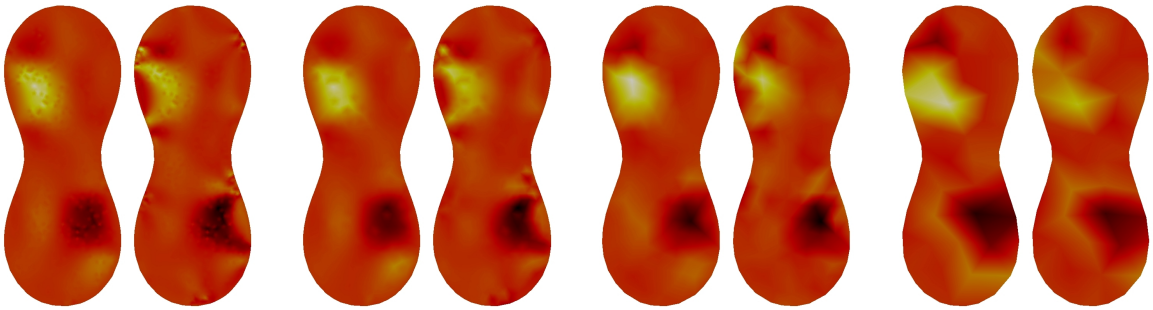
6.4. L-BFGS: mixing reparameterization and Sobolev gradients method

Another regularization consists in choosing an appropriate inner product when extracting the cost function gradient, eq. (11). Most often the $L_2(\mathcal{D})$ inner product is the one used to extract the gradient but, choosing another inner product such as the Sobolev one yields to much smoother and much closer-to-the-target reconstructions. Specially, the inner product that is used here is:

$$(\eta, \zeta)_{H^{1(\ell)}(\mathcal{D})} = (\eta, \zeta)_{L_2(\mathcal{D})} + \ell^2 \left(\nabla\eta, \nabla\zeta\right)_{L_2(\mathcal{D})} \quad \forall \eta, \zeta \in H^1(\mathcal{D}) \tag{24}$$

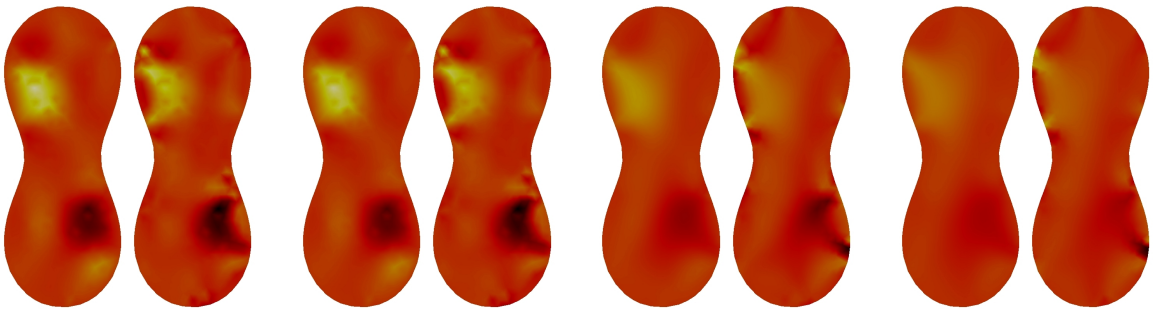
This implies that:

$$\begin{aligned} \left(\nabla^{H^{1(\ell)}}_{\varkappa} j, \eta\right)_{H^{1(\ell)}(\mathcal{D})} &= \left(\mathrm{Re}\left(\kappa_{ap}\varphi\bar{\varphi}^* - n_{\mathcal{D}} D^2 \kappa_{ap} \nabla\varphi \cdot \nabla\bar{\varphi}^*\right), \eta\right)_{\mathcal{Y}} \quad \forall \eta, \zeta \in H^1(\mathcal{D}) \\ \left(\nabla^{H^{1(\ell)}}_{\varsigma} j, \eta\right)_{H^{1(\ell)}(\mathcal{D})} &= \left(\mathrm{Re}\left(-n_{\mathcal{D}} D^2 \sigma'_{ap} \nabla\varphi \cdot \nabla\bar{\varphi}^*\right), \eta\right)_{\mathcal{Y}} \quad \forall \eta, \zeta \in H^1(\mathcal{D}) \end{aligned} \tag{25}$$



(a) L-BFGS, $\lambda = 0, \Xi = 4920$ (b) L-BFGS, $\lambda = 0, \Xi = 1278$ (c) L-BFGS, $\lambda = 0, \Xi = 292$ (d) L-BFGS, $\lambda = 0, \Xi = 124$

Figure 3: L-BFGS algorithm with reparameterization regularization. Influence of the reconstructions with parameterization of the control space.



(a) L-BFGS, $\Xi = 1278, \lambda = 0$ (b) L-BFGS, $\Xi = 1278, \lambda = 10^{-3}$ (c) L-BFGS, $\Xi = 1278, \lambda = 10^{-1}$ (d) L-BFGS, $\Xi = 1278, \lambda = 1$

Figure 4: L-BFGS algorithm with 2 regularizations. Influence of the reconstructions with the Tikhonov parameter after one reparameterization regularization.

Thus, the use of such inner product when considering the cost function gradient extraction leads to, after integration by parts:

$$\begin{aligned} \left(1 - \ell^2 \Delta\right) \nabla_{\varkappa}^{H^{1(\ell)}} j(\gamma) &= \sum_{k=1}^{K} \mathrm{Re}\left(\kappa_{ap} \varphi \bar{\varphi}^* - n_{\mathcal{D}} D^2 \kappa_{ap} \nabla\varphi \cdot \nabla\bar{\varphi}^*\right) \quad \forall \boldsymbol{x} \in \mathcal{D} \\ \nabla \nabla_{\varkappa}^{H^{1(\ell)}} j(\gamma) \cdot \vec{n} &= 0 \qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \forall \boldsymbol{\zeta} \in \partial\mathcal{D} \end{aligned} \tag{26}$$

and

$$\begin{aligned} \left(1 - \ell^2 \Delta\right) \nabla_{\varsigma}^{H^{1(\ell)}} j(\gamma) &= \sum_{k=1}^{K} \mathrm{Re}\left(-n_{\mathcal{D}} D^2 \sigma'_{ap} \nabla\varphi \cdot \nabla\bar{\varphi}^*\right) \quad \forall \boldsymbol{x} \in \mathcal{D} \\ \nabla \nabla_{\varsigma}^{H^{1(\ell)}} j(\gamma) \cdot \vec{n} &= 0 \qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \forall \boldsymbol{\zeta} \in \partial\mathcal{D} \end{aligned} \tag{27}$$

Figure 5 presents some reconstructions with the use of such Sobolev gradients combined with appropriate parameterizations of the control space. Different values for the Sobolev weighting parameter were used, from $\ell^2 = 0$ (i.e. the $L_2(\mathcal{D})$ inner product was actually used) to $\ell^2 = 1$.

Figure 5 shows that the reconstructions regularity increases with the Sobolev parameter. Actually, the use of the Sobolev gradients acts as a smoother so that the high-frequency fluctuations present in the adjoint variable are removed

or at least de-emphasized through the cost function computation. Figure 5 also shows that the ordinary $H^1(\mathcal{D})$ inner product (i.e. when $\ell = 1$) smoothes too much the cost function gradient (Figs. 5(d) and 5(h)), and that the ordinary $L_2(\mathcal{D})$ inner product yields to too fluctuating reconstructions (Figs. 5(a) and 5(e)). Choosing $\ell^2 \approx 0.1$ combined with the use of a coarse mesh for property parameterization yields much better reconstructions in the sense that the contrast is much better found in inclusions along with much better regularity of the solutions.

6.5. Analysis

It can be concluded that the use of Tikhonov-type regularization is absolutely compulsary when considering optimization algorithms that rely on matrix inversion: however the use of too small or too large weighting parameters gives fairly wrong reconstructions. However, combining this regularization technique with appropriate parameterization enhances the quality of reconstructions. Nevertheless, the Tikhonov regularization does not bring much improvements when considering optimizers that do not rely on matrix inversion, while the combination of an appropriate parameterization of the control space and the use of Sobolev gradients brings much more improvement. The numerical tests performed on an optical tomography application with the DA model could corroborate these main results. Finally, the combination of these two regularization tools makes the global inversion very elegant in the sense that there are very few parameters that have to be manually tuned, and their tuning is very easy when compared to other regularization tools like the Tikhonov one.

7. Extension to 3D

7.1. Numerical settings

Building on previous results, the three-dimensional reconstruction of the optical properties is now considered. In view of the increasing complexity of the problem, a particular effort has been done in parallelizing the numerical code. A parallelization strategy has been adopted because the most time-consuming part of the algorithm is the solutions of the forward and adjoint problems along with that of the Sobolev diffusion equation. The three subroutines related to these three problems are called within the cost function and cost function gradient routines. Moreover, both the cost function and its gradient are extensively called by both the optimizer and within the line-search process thus justifying parallelization.

Two options can be considered. The first one consists in dispatching the solutions of problems related to specific sources on specific processors (for instance with n sources and n processors, the forward and adjoint problems related to the source $i = 1 \in \{1, \ldots, n\}$ are solved on the i^{th} processor). The second option consists in computing the solutions of the linear systems associated to the forward, adjoint and Sobolev models with efficient parallel solvers. This second strategy has been chosen. Indeed, at a given iteration within the optimization process, i.e. for a given set of $\varkappa(\boldsymbol{x})$, $\varsigma(\boldsymbol{x})$, the matrices of the bilinear forms involved in both the forward and adjoint problems can be built and assembled only once, before being re-used. Doing so, the efficiency then relies on matrix system inversion, once the assembling

phase is achieved. Among numerous linear systems parallel solvers, the MUMPS solver [20] was chosen because it allows fast inversion, because it is a direct solver and because it is easily coupled with the FreeFem++ environment [21].

The optimizer has also been completely rewritten due to the incompatibility between the IPOpt package and the Message Passing Interface (MPI) routines. The original BFGS was implemented with the computation of the descent direction performed by the L-BFGS relationship [22].

Concerning the line-search process, an inexact one proposed by Fletcher [23] has been chosen because of its flexibility with respect to the accuracy of the one-dimensional optimization. Indeed, this accuracy is directly related to the computational time of the line-search procedure. Among the numerous parameters involved in the line-search algorithm, two concern more specifically the accuracy of the research and were chosen so that they make it quite inaccurate. Following [23, pp 20–23], parameters ρ and σ were chosen equal to 0.1 and 0.7, respectively. This results in a fast line search giving good results.

The original BFGS algorithm has been slightly modified in order to take into account the inexact line-search presented above: the global optimizer restarts the BFGS algorithm each time the numerical zero of the computer appears in the line-search procedure.

7.2. Test presentation and main results

The three dimensional domain $\mathcal{D}$ considered is a cube of 5 cm length. 6 sources are each located at the center of the six faces of the cube. The sources form squares of 5 mm length. There are also 4 square detectors of 5 mm large per face. The location of sources and detectors are schematically presented in Figure 6-a. This figure also presents the mesh used for the states φ and φ^* (Figure 6-b) and the mesh used for parameterizing the optical properties $\varkappa$ and ς (Figure 6-c). The dimension of the state mesh is 47 493 vertices and 245 612 tetrahedra, while it has been checked that such mesh ensures low computation errors. Following results of section 6, the dimension of the control mesh has been reduced to 22 945 vertices and 112 618 tetrahedra in order to enhance regularity. Concerning the optical properties to be identified within the cube, they were defined by:

$$\begin{aligned}
&\kappa(\boldsymbol{x}) = 0.06\,\mathrm{cm}^{-1}, \quad \sigma(\boldsymbol{x}) = 10\,\mathrm{cm}^{-1}, \quad \boldsymbol{x} \in \mathcal{D}_1, && \mathcal{D}_1 = [3.25, 4.25]^3\,\mathrm{cm}^3\\
&\kappa(\boldsymbol{x}) = 0.1\,\mathrm{cm}^{-1}, \quad \sigma(\boldsymbol{x}) = 30\,\mathrm{cm}^{-1}, \quad \boldsymbol{x} \in \mathcal{D}_2, && \mathcal{D}_2 = [0.75, 1.75]^3\,\mathrm{cm}^3\\
&\kappa(\boldsymbol{x}) = 0.08\,\mathrm{cm}^{-1}, \quad \sigma(\boldsymbol{x}) = 20\,\mathrm{cm}^{-1}, \quad \boldsymbol{x} \in \mathcal{D} \setminus (\mathcal{D}_1 \cup \mathcal{D}_2), && \mathcal{D} = [0, 5]^3\,\mathrm{cm}^3
\end{aligned}$$

As in the previous section, the inversion process begins with the cautious creation of synthetic data. These data, $\breve{\varphi}$, representing the pseudo-experimental measurements, are built on a finer mesh than that of generating the predictions φ in order to avoid the inverse crime [24]. The dimension of this finer mesh is equal to 67 103 vertices and 340 881 tetrahedra. Then, a multiplicative noise is applied to $\breve{\varphi}$ at the nodes of the sensors to simulate the noise inherent to experimental devices.

The control variables have been initialized to unity while values taken by a priori functions κ_{ap} and σ'_{ap} are those of the background, as for the two-dimensional test. Similarly, a noise of 30 dB was applied to the synthetic data

and the global optimizer was set to stop when the cost function reached the parameter resulting from the Morozov's discrepancy principle [19] or when the total number of BFGS iterations reached a critical value equal to 100 in this case.

Figure 7 presents the reconstructions with an appropriate reparameterization, and without any Tikhonov regularization. The first reconstruction is performed with $\ell^2 = 0$ and the second reconstruction is performed with $\ell^2 = 1.10^{-2}$. It is seen that the Sobolev gradient smooths the reconstruction and yields to fair reconstruction when compared to the use of "ordinary" gradients that yield to fluctuations in the reconstructions.

8. Conclusion

In the first part of this paper, the Gauss-Newton and L-BFGS algorithms were tested on the bi-dimensional diffuse-approximation-based optical tomography problem. Several regularization techniques were used: the reparameterization of the control variables and the Tikhonov penalization for each of the two algorithms, and the Sobolev gradients method specifically for the L-BFGS algorithm. The conclusions of this study can be summarized as follows:

- The reparameterization improves the quality of the reconstructions for both algorithms.
- The Tikhonov penalization is required for the GN method while being useless for the L-BFGS algorithm.
- The L-BGFS method provides the best reconstructions through the use of a suitable parameterization of the control variables and Sobolev gradients. These two regularization techniques have cumulative effects in the sense that the best reconstructions are obtained using both together.
- The Gauss-Newton method is globally less efficient than the L-BFGS method because of its requirements in terms of computational time and storage requirements. Moreover, the Gauss-Newton method is much less stable than the BFGS method. Indeed, the algorithm of Gauss-Newton necessarily needs the Tikhonov regularization with an appropriate, well-selected regularization parameter which in turn can be time consuming to determine.

In the second part, an inverse method based on the BFGS optimizer with an inexact line-search was presented for solving the three-dimensional diffuse optical tomography problem. To cope with the increase in computation time due to the addition of the third space coordinate, the optimization algorithm has been parallelized when solving for the partial differential equations involved in the inverse method. From the numerical simulations, it comes out that the conclusions of the first part concerning the L-BFGS algorithm remain true in the three-dimensional case.

Future research will focus on improving the reconstructions obtained by use of the 3D-code presented in the second part of this paper, for instance by coupling the inverse method with genetic algorithms.

9. Acknowledgements

The authors would like to thank ...

Appendix A. Calculation of the cost function gradient with the adjoint state method

The optimal method, for large-scale optimization problems, of the adjoint state method is used in this appendix to calculate the cost function gradient. In order not to complicate the developments which follow, the experiment has been assumed to involve a single light source.

Let us first introduce the adjoint variable $\varphi^* = \varphi_r^* + \mathrm{i}\varphi_i^*$ to the state one $\varphi = \varphi_r + \mathrm{i}\varphi_i$. The equation (3) implies that φ_r and φ_i are respectively solutions of the following partial differential equations:

$$-\nabla \cdot (D\nabla\varphi_r) + \kappa\varphi_r - \frac{2\pi\nu}{c}\varphi_i = 0 \quad \forall \boldsymbol{x} \in \mathcal{D} \tag{A.1}$$

$$\varphi_r + \frac{A}{2\gamma} D\nabla\varphi_r \cdot n = \frac{I}{\gamma}\mathbb{1}_{[x\in\partial\mathcal{D}_s]}(x) \quad \forall \boldsymbol{\zeta} \in \partial\mathcal{D} \tag{A.2}$$

$$-\nabla \cdot (D\nabla\varphi_i) + \kappa\varphi_i + \frac{2\pi\nu}{c}\varphi_r = 0 \quad \forall \boldsymbol{x} \in \mathcal{D} \tag{A.3}$$

$$\varphi_i + \frac{A}{2\gamma} D\nabla\varphi_i \cdot n = 0 \quad \forall \boldsymbol{\zeta} \in \partial\mathcal{D} \tag{A.4}$$

Remembering that $\kappa = \kappa_{ap}\varkappa$, $\sigma' = \sigma'_{ap}\varsigma$, $D = (n_{\mathcal{D}}(\kappa + \sigma'))^{-1}$ and introducing the residues of the forward models

$$\mathcal{R}(\kappa, \sigma', \varphi_r, \varphi_i) = -\nabla \cdot (D\nabla\varphi_r) + \kappa\varphi_r - \frac{2\pi\nu}{c}\varphi_i \tag{A.5}$$

$$\mathcal{S}(\kappa, \sigma', \varphi_r, \varphi_i) = -\nabla \cdot (D\nabla\varphi_i) + \kappa\varphi_i + \frac{2\pi\nu}{c}\varphi_r \tag{A.6}$$

and the Lagrange function

$$\mathcal{L}(\varkappa, \varsigma, \varphi_r, \varphi_i, \varphi_r^*, \varphi_i^*) = \mathcal{J}(\varphi_r, \varphi_i) + \mathcal{J}^+(\kappa, \sigma') + (\mathcal{R}, \varphi_r^*)_{L_2(\mathcal{D})} + (\mathcal{S}, \varphi_i^*)_{L_2(\mathcal{D})}, \tag{A.7}$$

it can be observed that $\mathcal{L}(\varkappa, \varsigma, \varphi_r, \varphi_i, \varphi_r^*, \varphi_i^*) = \mathcal{J}(\varphi_r, \varphi_i) + \mathcal{J}^+(\kappa, \sigma') = j(\varkappa, \varsigma)$ when φ_r and φ_i are solutions of eq. (A.1) and eq. (A.3), respectively. One can then deduces that for all $\delta\varkappa \in L^2(\mathcal{D})$ and for all φ_r^*, $\varphi_i^* \in H^1(\mathcal{D})$:

$$j'(\varkappa; \delta\varkappa) := (\nabla^{\varkappa} j, \delta\varkappa)_{L_2(\mathcal{D})} \tag{A.8}$$

$$= \frac{\partial\mathcal{L}}{\partial\varkappa}\delta\varkappa + \frac{\partial\mathcal{L}}{\partial\varphi_r}\frac{\partial\varphi_r}{\partial\varkappa}\delta\varkappa + \frac{\partial\mathcal{L}}{\partial\varphi_i}\frac{\partial\varphi_i}{\partial\varkappa}\delta\varkappa \tag{A.9}$$

where the directionnal derivative $j'(\varkappa; \delta\varkappa)$ is related to the partial differential sum of $\mathcal{L}(\varkappa, \varsigma, \varphi_r, \varphi_i, \varphi_r^*, \varphi_i^*)$. Eq. (A.9)

can be calculated by the sum of the following nine terms:

$$\left(\frac{\partial \mathcal{J}}{\partial \varphi_r}, \varphi_r'\right)_{L_2(\partial \mathcal{D}_d)} = \left(\frac{\varphi_r - \breve{\varphi}_r}{|\breve{\varphi}|^2}, \varphi_r'\right)_{L_2(\partial \mathcal{D}_d)} \tag{A.10}$$

$$\left(\frac{\partial \mathcal{J}}{\partial \varphi_i}, \varphi_i'\right)_{L_2(\partial \mathcal{D}_d)} = \left(\frac{\varphi_i - \breve{\varphi}_i}{|\breve{\varphi}|^2}, \varphi_i'\right)_{L_2(\partial \mathcal{D}_d)} \tag{A.11}$$

$$\left(\frac{\partial \mathcal{R}}{\partial \varphi_r}\varphi_r', \varphi_r^*\right)_{L_2(\mathcal{D})} = (-\nabla \cdot (D\nabla \varphi_r') + \kappa \varphi_r', \varphi_r^*)_{L_2(\mathcal{D})} \tag{A.12}$$

$$\left(\frac{\partial \mathcal{R}}{\partial \varphi_i}\varphi_i', \varphi_r^*\right)_{L_2(\mathcal{D})} = \left(-\frac{2\pi\nu}{c}\varphi_i', \varphi_r^*\right)_{L_2(\mathcal{D})} \tag{A.13}$$

$$\left(\frac{\partial \mathcal{S}}{\partial \varphi_r}\varphi_r', \varphi_i^*\right)_{L_2(\mathcal{D})} = \left(\frac{2\pi\nu}{c}\varphi_r', \varphi_i^*\right)_{L_2(\mathcal{D})} \tag{A.14}$$

$$\left(\frac{\partial \mathcal{S}}{\partial \varphi_i}\varphi_i', \varphi_i^*\right)_{L_2(\mathcal{D})} = (-\nabla \cdot (D\nabla \varphi_i') + \kappa \varphi_i', \varphi_i^*)_{L_2(\mathcal{D})} \tag{A.15}$$

$$\left(\frac{\partial \mathcal{R}}{\partial \varkappa}\delta\varkappa, \varphi_r^*\right)_{L_2(\mathcal{D})} = \left(\nabla \cdot \left(n_{\mathcal{D}}\kappa_{ap} D^2 \delta\varkappa \nabla \varphi_r\right) + \kappa_{ap}\varphi_r \delta\varkappa, \varphi_r^*\right)_{L_2(\mathcal{D})} \tag{A.16}$$

$$\left(\frac{\partial \mathcal{S}}{\partial \varkappa}\delta\varkappa, \varphi_i^*\right)_{L_2(\mathcal{D})} = \left(\nabla \cdot \left(n_{\mathcal{D}}\kappa_{ap} D^2 \delta\varkappa \nabla \varphi_i\right) + \kappa_{ap}\varphi_i \delta\varkappa, \varphi_i^*\right)_{L_2(\mathcal{D})} \tag{A.17}$$

$$\left(\frac{\partial \mathcal{J}^+}{\partial \varkappa}, \delta\varkappa\right)_{L_2(\mathcal{D})} = \left(\lambda\left(\frac{\kappa - \kappa_{ap}}{\kappa_{ap}}\right), \delta\varkappa\right)_{L_2(\mathcal{D})} \tag{A.18}$$

Applying the Green identity twice to eqs. (A.12) and (A.15) and once to eqs. (A.16) and (A.17) yields:

$$\left(\frac{\partial \mathcal{R}}{\partial \varphi_r}\varphi_r', \varphi_r^*\right)_{L_2(\mathcal{D})} = (-\nabla \cdot (D\nabla \varphi_r^*) + \kappa \varphi_r^*, \varphi_r')_{L_2(\mathcal{D})} + (D\nabla \varphi_r^* \cdot n, \varphi_r')_{L_2(\partial\mathcal{D})} - (D\nabla \varphi_r' \cdot n, \varphi_r^*)_{L_2(\partial\mathcal{D})} \tag{A.19}$$

$$\left(\frac{\partial \mathcal{S}}{\partial \varphi_i}\varphi_i', \varphi_i^*\right)_{L_2(\mathcal{D})} = (-\nabla \cdot (D\nabla \varphi_i^*) + \kappa \varphi_i^*, \varphi_i')_{L_2(\mathcal{D})} + (D\nabla \varphi_i^* \cdot n, \varphi_i')_{L_2(\partial\mathcal{D})} - (D\nabla \varphi_i' \cdot n, \varphi_i^*)_{L_2(\partial\mathcal{D})} \tag{A.20}$$

$$\left(\frac{\partial \mathcal{R}}{\partial \varkappa}\delta\varkappa, \varphi_r^*\right)_{L_2(\mathcal{D})} = -\left(n_{\mathcal{D}}\kappa_{ap} D^2 \delta\varkappa \nabla \varphi_r, \nabla \varphi_r^*\right)_{L_2(\mathcal{D})} + \left(\kappa_{ap}\varphi_r \delta\varkappa, \varphi_r^*\right)_{L_2(\mathcal{D})} + \left(n_{\mathcal{D}}\kappa_{ap} D^2 \delta\varkappa \nabla \varphi_r \cdot n, \varphi_r^*\right)_{L_2(\partial\mathcal{D})} \tag{A.21}$$

$$\left(\frac{\partial \mathcal{S}}{\partial \varkappa}\delta\varkappa, \varphi_i^*\right)_{L_2(\mathcal{D})} = -\left(n_{\mathcal{D}}\kappa_{ap} D^2 \delta\varkappa \nabla \varphi_i, \nabla \varphi_i^*\right)_{L_2(\mathcal{D})} + \left(\kappa_{ap}\varphi_i \delta\varkappa, \varphi_i^*\right)_{L_2(\mathcal{D})} + \left(n_{\mathcal{D}}\kappa_{ap} D^2 \delta\varkappa \nabla \varphi_i \cdot n, \varphi_i^*\right)_{L_2(\partial\mathcal{D})} \tag{A.22}$$

Next, the directionnal derivatives of the Robin boundary conditions for eqs. (A.1) and (A.3) give:

$$-D\nabla \varphi_r' \cdot n = \frac{2\gamma}{A}\varphi_r' - n_{\mathcal{D}}\kappa_{ap} D^2 \delta\varkappa \nabla \varphi_r \cdot n \tag{A.23}$$

$$-D\nabla \varphi_i' \cdot n = \frac{2\gamma}{A}\varphi_i' - n_{\mathcal{D}}\kappa_{ap} D^2 \delta\varkappa \nabla \varphi_i \cdot n \tag{A.24}$$

Replacing the last terms of eqs. (A.19) and (A.20) by the second members of eqs. (A.23) and (A.24), the sum of eqs. (A.19) to (A.22), Σ_{Temp}, is equal to:

$$\begin{aligned}\Sigma_{\text{Temp}} = {} & (-\nabla \cdot (D\nabla \varphi_r^*) + \kappa \varphi_r^*, \varphi_r')_{L_2(\mathcal{D})} + (D\nabla \varphi_r^* \cdot n, \varphi_r')_{L_2(\partial\mathcal{D})} + (-\nabla \cdot (D\nabla \varphi_i^*) + \kappa \varphi_i^*, \varphi_i')_{L_2(\mathcal{D})} + (D\nabla \varphi_i^* \cdot n, \varphi_i')_{L_2(\partial\mathcal{D})} \\ & - \left(n_{\mathcal{D}}\kappa_{ap} D^2 \delta\varkappa \nabla \varphi_r, \nabla \varphi_r^*\right)_{L_2(\mathcal{D})} + \left(\kappa_{ap}\varphi_r \delta\varkappa, \varphi_r^*\right)_{L_2(\mathcal{D})} - \left(n_{\mathcal{D}}\kappa_{ap} D^2 \delta\varkappa \nabla \varphi_i, \nabla \varphi_i^*\right)_{L_2(\mathcal{D})} + \left(\kappa_{ap}\varphi_i \delta\varkappa, \varphi_i^*\right)_{L_2(\mathcal{D})} \\ & + \left(\frac{2\gamma}{A}\varphi_r^*, \varphi_r'\right)_{L_2(\partial\mathcal{D})} + \left(\frac{2\gamma}{A}\varphi_i^*, \varphi_i'\right)_{L_2(\partial\mathcal{D})}\end{aligned} \tag{A.25}$$

Thereby we obtain:

$$
\begin{aligned}
j'(\varkappa;\delta\varkappa) &:= (\nabla^{\varkappa} j, \delta\varkappa)_{L_2(\mathcal{D})} \\
&= \left(\kappa_{ap}(\varphi_r\varphi_r^* + \varphi_i\varphi_i^*) - n_{\mathcal{D}}\kappa_{ap}D^2\left(\nabla\varphi_r\cdot\nabla\varphi_r^* + \nabla\varphi_i\cdot\nabla\varphi_i^*\right) + \lambda\left(\frac{\kappa-\kappa_{ap}}{\kappa_{ap}}\right), \delta\varkappa\right)_{L_2(\mathcal{D})} \\
&- \left(\nabla\cdot(D\nabla\varphi_r^*) - \kappa\varphi_r^* - \frac{2\pi\nu}{c}\varphi_i^*, \varphi_r'\right)_{L_2(\mathcal{D})} - \left(\nabla\cdot(D\nabla\varphi_i^*) - \kappa\varphi_i^* + \frac{2\pi\nu}{c}\varphi_r^*, \varphi_i'\right)_{L_2(\mathcal{D})} \\
&+ \left(\frac{2\gamma}{A}\varphi_r^* + D\left(\nabla\varphi_r^*\cdot n\right), \varphi_r'\right)_{L_2(\partial\mathcal{D})} + \left(\frac{2\gamma}{A}\varphi_i^* + D\left(\nabla\varphi_i^*\cdot n\right), \varphi_i'\right)_{L_2(\partial\mathcal{D})} \\
&+ \left(\frac{\varphi_r - \breve{\varphi}_r}{|\breve{\varphi}|^2}, \varphi_r'\right)_{L_2(\partial\mathcal{D}_d)} + \left(\frac{\varphi_i - \breve{\varphi}_i}{|\breve{\varphi}|^2}, \varphi_i'\right)_{L_2(\partial\mathcal{D}_d)}
\end{aligned}
\tag{A.26}
$$

Hence, solving the following adjoint equations for $\varphi^* = \varphi_r^* + \mathrm{i}\varphi_i^*$,

$$
-\nabla\cdot(D\nabla\varphi^*) + \left(\kappa - \frac{2\mathrm{i}\pi\nu}{c}\right)\varphi^* = 0 \qquad \forall \boldsymbol{x}\in\mathcal{D} \tag{A.27}
$$

$$
\frac{2\gamma}{A}\varphi^* + D\left(\nabla\varphi^*\cdot n\right) = \frac{\varphi - \breve{\varphi}}{|\breve{\varphi}|^2}\mathbb{1}_{[x\in\partial\mathcal{D}_d]}(x) \qquad \forall\boldsymbol{\zeta}\in\partial\mathcal{D}, \tag{A.28}
$$

it can be deduced that the cost function gradient related to $\varkappa$ is:

$$
(\nabla^{\varkappa} j, \delta\varkappa)_{L_2(\mathcal{D})} = \left(\kappa_{ap}(\varphi_r\varphi_r^* + \varphi_i\varphi_i^*) - n_{\mathcal{D}}\kappa_{ap}D^2\left(\nabla\varphi_r\cdot\nabla\varphi_r^* + \nabla\varphi_i\cdot\nabla\varphi_i^*\right) + \lambda\left(\frac{\kappa-\kappa_{ap}}{\kappa_{ap}}\right), \delta\varkappa\right)_{L_2(\mathcal{D})} \tag{A.29}
$$

$$
= \left(\mathrm{Re}\left(\kappa_{ap}\varphi\bar{\varphi}^* - n_{\mathcal{D}}\kappa_{ap}D^2\nabla\varphi\cdot\nabla\bar{\varphi}^*\right) + \lambda\left(\frac{\kappa-\kappa_{ap}}{\kappa_{ap}}\right), \delta\varkappa\right)_{L_2(\mathcal{D})} \tag{A.30}
$$

The same developments for ς lead to:

$$
(\nabla^{\varsigma} j, \delta\varsigma)_{L_2(\mathcal{D})} = \left(-n_{\mathcal{D}}\sigma'_{ap}D^2\left(\nabla\varphi_r\cdot\nabla\varphi_r^* + \nabla\varphi_i\cdot\nabla\varphi_i^*\right) + \lambda\left(\frac{\sigma'-\sigma'_{ap}}{\sigma'_{ap}}\right), \delta\varsigma\right)_{L_2(\mathcal{D})} \tag{A.31}
$$

$$
= \left(\mathrm{Re}\left(-n_{\mathcal{D}}\sigma'_{ap}D^2\nabla\varphi\cdot\nabla\bar{\varphi}^*\right) + \lambda\left(\frac{\sigma'-\sigma'_{ap}}{\sigma'_{ap}}\right), \delta\varsigma\right)_{L_2(\mathcal{D})} \tag{A.32}
$$

References

[1] S. Arridge, Optical tomography in medical imaging, Inverse Problems 15 (2) (1999) R41–R93.

[2] A. Charette, J. Boulanger, H. K. Kim, An overview on recent radiation transport algorithm development for optical tomography imaging, Journal of Quantitative Spectroscopy and Radiative Transfer 109 (17-18) (2008) 2743–2766.

[3] B. Protas, T. Bewley, G. Hagen, A computational framework for the regularization of adjoint analysis in multiscale PDE systems, Journal of Computational Physics 195 (1) (2004) 49–89.

[4] D. Liu, J. Nocedal, On the limited memory BFGS method for large scale optimization, Mathematical Programming: Series A and B 45 (3) (1989) 503–528.

[5] G. Chavent, Nonlinear least squares for inverse problems, Springer, 2009.

[6] K. D. Paulsen, P. M. Meaney, M. J. Moskowitz, J. M. Sullivan Jr, A dual mesh scheme for finite element based reconstruction algorithms, Medical Imaging, IEEE Transactions on 14 (3) (1995) 504–514.

[7] O. Balima, Y. Favennec, F. Dubot, D. Rousse, Finite elements parameterization of optical tomography with the radiative transfer equation in frequency domain, in: Journal of Physics: Conference Series, Vol. 369, IOP Publishing, 2012, p. 012022.

[8] Y. Favennec, F. Dubot, B. Rousseau, D. Rousse, Mixing regularization tools for enhancing regularity in optical tomography applications, in: O. Fudym, J. L. Battaglia, G. S. Dulikravich (Eds.), IPDO 2013 : 4th Inverse problems, design and optimization symposium, Albi, 2013.

[9] T. Saratoon, T. Tarvainen, B. T. Cox, S. R. Arridge, A gradient-based method for quantitative photoacoustic tomography using the radiative transfer equation, Inverse Problems 29 (2013) 075006.

[10] M. F. Modest, Radiative Heat Transfer, Mechanical Engineering, McGraw Hill, 1993.

[11] H. Dehghani, S. Srinivasan, B. W. Pogue, A. Gibson, Numerical modelling and image reconstruction in diffuse optical tomography, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 367 (1900) (2009) 3073–3093.

[12] V. Brattka, A. Yoshikawa, Towards computability of elliptic boundary value problems in variational formulation, Journal of Complexity 22 (6) (2006) 858–880.

[13] A. Klose, Optical tomography based on the equation of radiative transfer, Ph.D. thesis, Department of Physiscs Freie Universität, Berlin Germany (October 2001).

[14] G. Allaire, Numerical analysis and optimization, Oxford Science Publications, 2007.

[15] J.-L. Lions, P. Faurre, Cours d'analyse numérique, Ecole polytechnique, 1982.

[16] M. Schweiger, S. R. Arridge, I. Nissilä, Gauss–newton method for image reconstruction in diffuse optical tomography, Physics in medicine and biology 50 (10) (2005) 2365.

[17] V. Akcelik, G. Biros, O. Ghattas, Parallel multiscale gauss-newton-krylov methods for inverse wave propagation, in: Supercomputing, ACM/IEEE 2002 Conference, IEEE, 2002, pp. 41–41.

[18] A. Wächter, L. T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, Mathematical programming 106 (1) (2006) 25–57.

[19] V. A. Morozov, Z. Nashed, A. Aries, Methods for solving incorrectly posed problems, Springer-Verlag New York, 1984.

[20] P. R. Amestoy, A. Guermouche, J.-Y. L'Excellent, S. Pralet, Hybrid scheduling for the parallel solution of linear systems, Parallel computing 32 (2) (2006) 136–156.

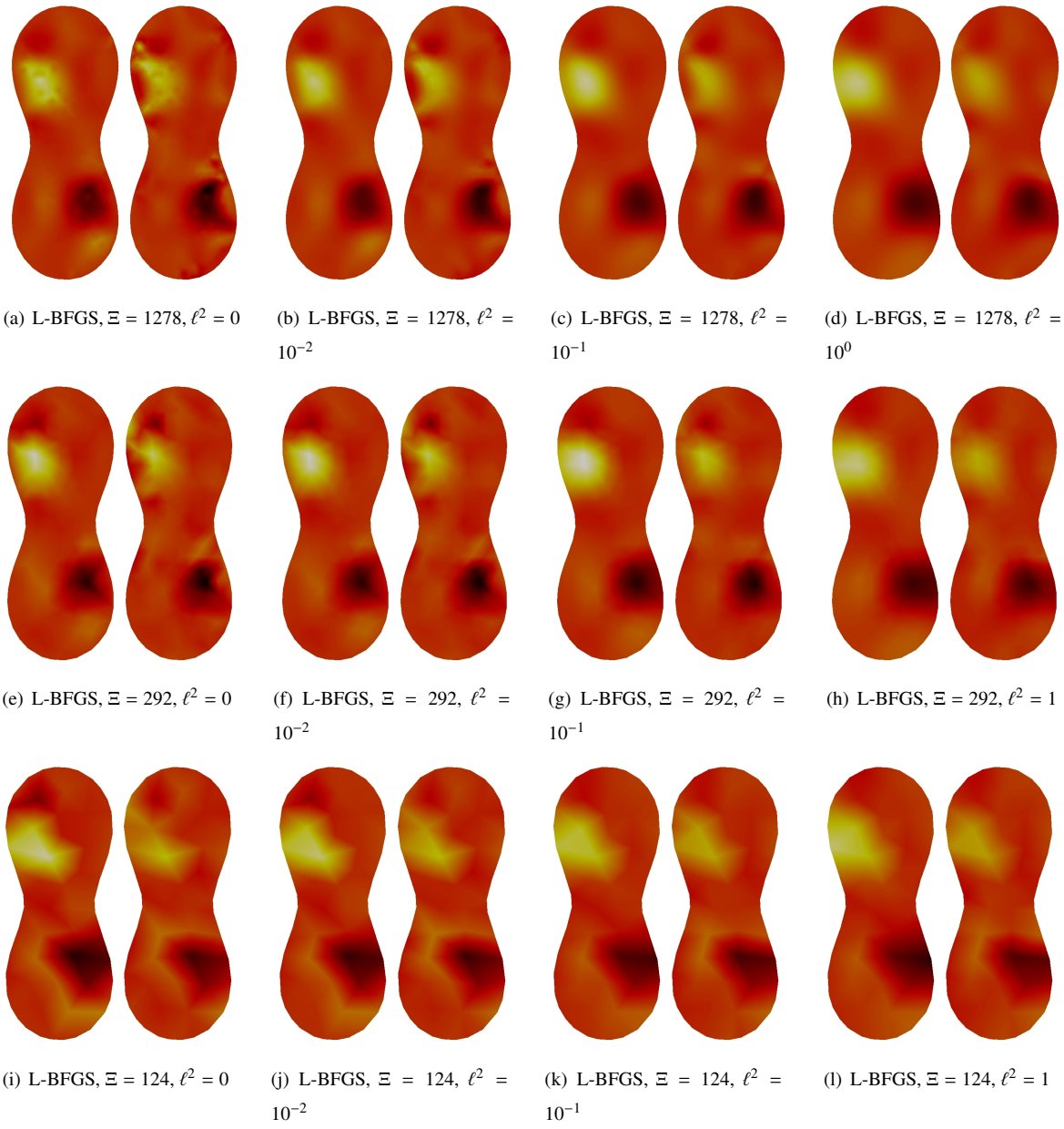
[21] F. Hecht, New development in freefem++, Journal of Numerical Mathematics 20 (3-4) (2012) 251–266.

[22] J. Nocedal, Updating quasi-newton matrices with limited storage, Mathematics of computation 35 (151) (1980) 773–782.

[23] R. Fletcher, Practical methods of optimization, vol. 1, unconstrained optimization, British Library Cataloguing in Publication Data (1980) 126.

[24] D. L. Colton, R. Kress, Inverse acoustic and electromagnetic scattering theory, Vol. 93, Springer, 1992.

[25] C. Geuzaine, J.-F. Remacle, Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities, International Journal for Numerical Methods in Engineering 79 (11) (2009) 1309–1331.

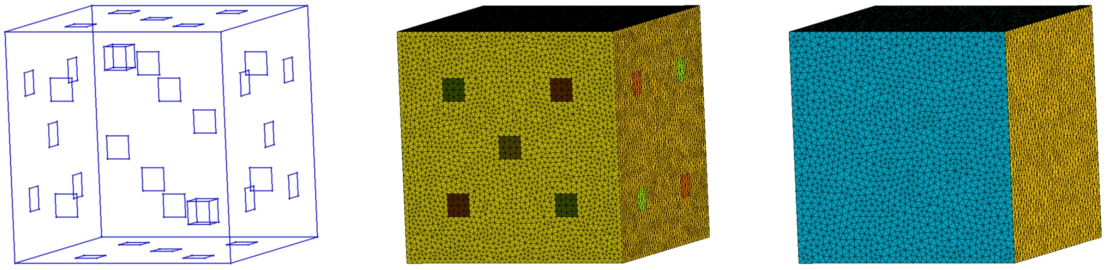


(a) L-BFGS, $\Xi = 1278$, $\ell^2 = 0$ (b) L-BFGS, $\Xi = 1278$, $\ell^2 = 10^{-2}$ (c) L-BFGS, $\Xi = 1278$, $\ell^2 = 10^{-1}$ (d) L-BFGS, $\Xi = 1278$, $\ell^2 = 10^{0}$

(e) L-BFGS, $\Xi = 292$, $\ell^2 = 0$ (f) L-BFGS, $\Xi = 292$, $\ell^2 = 10^{-2}$ (g) L-BFGS, $\Xi = 292$, $\ell^2 = 10^{-1}$ (h) L-BFGS, $\Xi = 292$, $\ell^2 = 1$

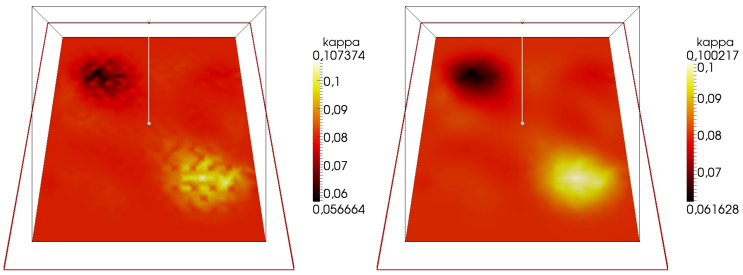
(i) L-BFGS, $\Xi = 124$, $\ell^2 = 0$ (j) L-BFGS, $\Xi = 124$, $\ell^2 = 10^{-2}$ (k) L-BFGS, $\Xi = 124$, $\ell^2 = 10^{-1}$ (l) L-BFGS, $\Xi = 124$, $\ell^2 = 1$

Figure 5: L-BFGS algorithm with 2 regularizations. Influence of the reconstructions with the Sobolev parameter after one reparameterization regularization.

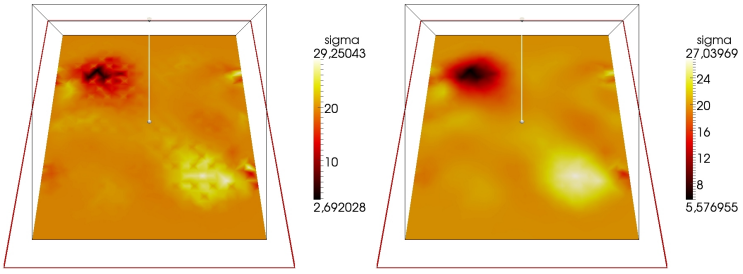


(a) Synthetic data mesh (b) State mesh (c) Control mesh

Figure 6: Geometry used for defining the synthetic data mesh with its two embedded inclusions and plot of the state and control meshes. The meshes have been created with the Gmsh software [25].



(a) From left to right: κ with $\ell^2 = 0$, κ with $\ell^2 = 1.10^{-2}$



(b) From left to right: σ with $\ell^2 = 0$, σ with $\ell^2 = 1.10^{-2}$

Figure 7: Reconstruction of the optical properties with 30 dB of noise and after 100 iterations.