

# Constrained optimization in seismic reflection tomography: an SQP augmented Lagrangian approach

F. Delbos<sup>1</sup>, J. Ch. Gilbert<sup>2</sup>, R. Glowinski<sup>3</sup>, D. Sinoquet<sup>1</sup>

<sup>1</sup> Institut Français du Pétrole, 1 & 4 avenue de Bois-Préau, 92852 Reuil-Malmaison, France

<sup>2</sup> Institut National de la Recherche en Informatique et en Automatique, BP 105, 78153 Le Chesnay Cedex, France

<sup>3</sup> University of Houston, 4800 Calhoun Rd, Houston, TX 77204-3476, USA

Received 2004 July; in original form 2004 July

## SUMMARY

Seismic reflection tomography is a method for determining a subsurface velocity model from the traveltimes of seismic waves reflecting on geological interfaces. From an optimization viewpoint, the problem consists in minimizing a nonlinear least-squares function measuring the mismatch between observed traveltimes and those calculated by ray tracing in this model. The introduction of a priori information on the model is crucial to reduce the under-determination. The contribution of this paper is to introduce a technique able to take into account geological a priori information in the reflection tomography problem expressed as constraints in the optimization problem. This technique is based on a Gauss-Newton sequential quadratic programming approach. At each Gauss-Newton step, a solution to a convex quadratic optimization problem subject to linear constraints is computed thanks to an augmented Lagrangian algorithm. Our choice for this optimization method is motivated and its original aspects are described. The efficiency of the method is demonstrated on a 2D OBC real data set and on a 3D real data set: the introduction of constraints coming both from well logs and from geological knowledge allows us to reduce the under-determination of both inverse problems.

**Key words:** seismic reflection tomography, ray tracing, a priori information, least-squares approach, constrained optimization, SQP algorithm, augmented Lagrangian.

## 1 INTRODUCTION

Geophysical methods for imaging a complex geological subsurface in petroleum exploration requires the determination of an accurate wave propagation velocity model. Seismic reflection tomography turns out to be an efficient method for doing this: it determines the seismic velocity distribution from the traveltimes associated with the seismic waves reflecting on geological surfaces. This inverse problem requires the solution to the specific forward problem, which consists in computing these traveltimes for a given subsurface model by a ray tracing method (based on a high frequency approximation of the wave equation, see Červený (1987) and Jurado et al. (1998)). The inverse problem is formulated as the minimization of the least-squares function that measures the mismatch between traveltimes calculated by ray tracing and the observed traveltimes.

The main interests of reflection tomography are

- its flexibility for handling various types of traveltime data simultaneously (primary reflections but also multiple reflections, traveltimes associated with converted waves -PS data-, surface seismic, well seismic), provided

that the ray tracing allows the computation of such data,

- the low computational time of the forward operator, in comparison with the time needed for the calculation of the wave equation solutions,
- the reduced number of local minima of the inverse problem, in comparison with the seismic inversion based on the wave equation simulation,
- its ability to integrate a priori geological information (via a least-squares formulation).

This method has been successfully applied to numerous real data sets (Ehinger et al. (2001) and Broto et al. (2003)). Nevertheless, the under-determination of the inverse problem generally requires the introduction of additional information to reduce the number of admissible models. Penalty terms modeling this information (as mentioned previously) can be added to the seismic terms in the objective function but the tuning of the penalty weights may be tricky. The standard methodology to invert complex subsurface structures (model composed of several velocity fields and reflectors), a top-down layer-stripping approach, may be inadequate.

quate. This approach consists in inverting separately each velocity layers (with its associated reflectors) starting from the upper layer to the lower one. To limit bad data fitting for deeper layers, a global inversion approach, which consists in simultaneously inverting all the velocity layers and interfaces of the subsurface model, is recommended. But, this method is often discarded due to its convergence troubles: because of the underlying under-determination, a global inversion of complex subsurface structures often leads to a bad subsurface model on which the ray tracing method fails to compute the traveltimes. Additional constraints on the model (for instance, on layer thicknesses to avoid non-physical interface intersections) are necessary to avoid those non admissible models.

We believe that the possibility to introduce constraints in the optimization process can overcome part of those difficulties. Equality and inequality constraints can indeed model many different types of a priori information. An optimization approach that can face these constraints efficiently will then discharge the final user of the inversion seismic software from the cumbersome task of tuning the weights associated with the additional penalty terms in the objective function.

The goals of this paper are twofold. The first one is to present our constrained nonlinear optimization method and to motivate its appropriateness to constrained reflection tomography. The second objective is to show the contribution of constraints in reflection tomography and the numerical efficiency of the constrained optimization method thanks to its application on a 2D PP/PS real data set and then on a 3D PP real data set.

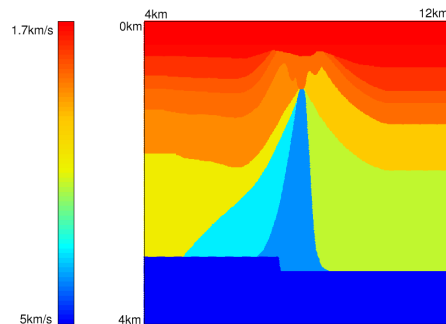
We recall the problem of interest and introduce the notation in Section 2. In Section 3, our SQP augmented Lagrangian approach for constrained reflection tomography problems is described and motivated. Numerical experiments on real data sets are detailed and analyzed in Section 4. We conclude with Section 5.

## 2 THE SEISMIC REFLECTION TOMOGRAPHY PROBLEM

### 2.1 The unconstrained problem

Let us first recall the problem of interest and introduce the notation. The choice of the model representation is crucial for the efficiency of the methods used to solve the forward and inverse problems. Lailly & Sinoquet (1996) have discussed the interest of different types of velocity models. We have chosen here a *blocky* model, where the velocity distribution is described by slowly varying layer velocities (or velocity blocks) delimited by interfaces (see Figure 1). With this representation, we introduce explicitly a strong a priori information: the number of layers. The number of parameters describing the velocity variations is limited thanks to the explicit introduction of velocity discontinuities (the velocity within a layer varies smoothly). The model is thus composed of two kinds of parameters: those describing the velocity variations within the layers and those describing the geometry of the interfaces delimiting the layers. Parameters describing the velocity anisotropy can also be included (see Jurado et al. (1998) and Stopin (2001) for more details).

The  $i^{th}$  interface is represented by a cubic B-spline func-



**Figure 1.** An example of a *blocky* subsurface velocity model.

tion (de Boor (1978) and Inoue (1986))  $\hat{z}_i(x, y)$ , whose coefficients define a vector  $z_i$  ( $x$  and  $y$  are the horizontal coordinates). Similarly, the  $i^{th}$  velocity field is represented by a cubic B-spline function  $\hat{v}_i(x, y, z)$  or  $\hat{v}_i(x, y) + kz$  with known scalar  $k$  ( $z$  is the vertical coordinate); the vector  $v_i$  contains the velocity coefficients. For  $n_v$  layer velocities and  $n_z$  interfaces, we collect the coefficients  $v_1, \dots, v_{n_v}$  in one vector  $v$  and the coefficients  $z_1, \dots, z_{n_z}$  in one vector  $z$ . The *model* vector  $m \in \mathbb{R}^n$  is defined here as  $m = (v, z)$ . The  $C^2$  smoothness of the functions describing the model allows the exact computation of derivatives of the traveltimes with respect to the model, quantities useful for ray tracing and tomography.

Given a model  $m$  and an acquisition survey (locations of the sources and receivers) a vector of traveltimes  $T(m)$  of seismic reflected waves can be computed by ray tracing (see Červený (1987) and Jurado et al. (1998)). The mapping  $T : \mathbb{R}^n \rightarrow \mathbb{R} : m \mapsto T(m)$  is nonlinear. We assume that it is differentiable. In practice, this assumption may not be satisfied, in particular, the forward operator may even not be defined when rays escape from the region of interest or when a layer thickness vanishes (non-physical interface intersections as mentioned in the introduction).

Reflection traveltime tomography is the corresponding inverse problem: its purpose is to adjust  $m$  so that  $T(m)$  best matches a vector of traveltimes  $T^{obs} \in \mathbb{R}^n$  (the observed traveltimes) picked on seismic data. Since Gauss (1809), it is both classical and natural to formulate this problem as a least-squares one:

$$\min_{m \in \mathbb{R}^n} \frac{1}{2} \left\| T(m) - T^{obs} \right\|^2, \quad (1)$$

where  $\|\cdot\|$  denotes the Euclidean norm.

The fact that problem (1) may be ill-posed has been pointed out by many authors (see for instance Delprat-Jannaud & Lailly (1993), Bube & Meadows (1999)). To ensure well-posedness, a curvature regularization is often introduced (Tikhonov & Arsenin (1977)). We use the sum of the squared  $L^2$ -norms of all the second order partial derivatives of every velocity  $\hat{v}_i$  and reflector  $\hat{z}_i$  (see for instance Delprat-Jannaud & Lailly (1993)). Such a regularization term can be written as  $m^\top R m$ , where  $R$  is a symmetric positive semidef-

inite matrix that only depends on the B-spline basis functions (it is independent of  $m$ ). Thus, instead of problem (1), we consider the regularized least-squares problem

$$\min_{m \in \mathbb{R}^n} \left( f(m) := \frac{1}{2} \|T(m) - T^{obs}\|^2 + \frac{\sigma}{2} m^\top R m \right), \quad (2)$$

where the regularization weight  $\sigma$  is positive, and  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is called the *cost function* (or *objective function*). The choice of the parameter  $\sigma$  is a difficult task. In practice, we use the L-curve method (see Hansen (1992)), also called the continuation method (Bube & Langan (1994)): starting from a large regularization weight, we decrease it regularly to retrieve more and more varying models until the data are fitted with the expected accuracy. The solution model is thus the smoothest model that fits the data up to a certain precision. This methodology allows us to do stable inversions. In the sequel, when we consider the objective function of problem (2), we assume that its regularization weight is fixed.

The unconstrained minimization problem of seismic reflection tomography, defined by (2), has the following features:

- the size of the data space and of the model space can be quite large (up to  $10^6$  traveltimes and  $10^5$  unknowns),
- the problem is ill-conditioned (Chauvier et al. (2000) have observed that the condition number of the approximated Hessian  $H_k$  given by equation (5) below can go up to  $10^9$  for a matrix of order 500),
- a forward simulation, i.e., a computation of  $T(m)$ , is CPU time consuming because of the large number of source-receiver pairs,
- the traveltime operator  $T$  is nonlinear, revealing the complexity of wave propagation in the subsurface.

To minimize efficiently a function like  $f$  in (2), it is highly desirable to have its gradient available. In the present context, thanks to Fermat's principle (Bishop et al. (1985)), it is inexpensive to compute row by row the Jacobian matrix  $J(m)$  of  $T$  at  $m$ . Recall that its  $(i, j)$ th element is the partial derivative

$$J(m)_{ij} = \frac{\partial T_i}{\partial m_j}. \quad (3)$$

The gradient of the objective is then obtained by the formula  $\nabla f(m) = J(m)^\top (T(m) - T^{obs}) + \sigma R m$ . It is also natural to ask whether one can compute the second derivatives of  $f$ . The answer is however negative. Therefore, using a pure Newton method to solve (2) is hopeless.

There are at least two classes of methods that can take advantage of the sole first order derivatives:

- quasi-Newton (QN) methods,
- Gauss-Newton (GN) methods.

Standard QN methods do not use the structure of the least-squares problems, but have a larger scope of application. They are used for solving least-squares problems when the computation of the Jacobian matrix  $J$  is much more time consuming than the computation of the *residual* vector  $T(m) - T^{obs}$  (see Courtier & Talagrand (1987) and Courtier et al. (1994) for a typical example in meteorology). We have mentioned above that this is not our case. On the other hand, the GN methods fully take benefit of the Jacobian matrix  $J$  by taking  $J^\top J$  as an approximation of the Hessian of the first term in  $f$ . This algorithm can exhibit slow con-

vergence when the residual is large at the solution and when  $T$  is strongly nonlinear at the solution. This does not seem to be the case in the problem we consider. Sometimes GN and QN methods are combined to improve the approximation of the Hessian of the first part of  $f$  by  $J^\top J$  (see Dennis et al. (1981), Yabe & Yamaki (1995) and the references therein).

The above discussion motivates our choice of a classical line-search GN method to solve the unconstrained optimization problem (Chauvier et al. (2000)). The  $k$ th iteration,  $k \geq 0$ , proceeds as follows. Let  $m_k$  be the approximate solution known at the beginning of the iteration. Note

$$T_k := T(m_k) \quad \text{and} \quad J_k := J(m_k). \quad (4)$$

First, an approximate solution  $d_k$  to the following tangent quadratic problem is computed

$$\min_{d \in \mathbb{R}^n} \frac{1}{2} \|J_k d + T_k - T^{obs}\|^2 + \frac{\sigma}{2} (m_k + d)^\top R (m_k + d).$$

This is the quadratic approximation of  $f$  about  $m_k$ , in which the costly computation of the second derivatives of  $T$  has been neglected. By the choice of the positive semi-definite regularization matrix  $R$ , the Hessian of this quadratic function in  $d$ , namely

$$H_k := J_k^\top J_k + \sigma R, \quad (5)$$

is usually positive definite. Therefore, it makes sense to minimize the above quadratic function by a preconditioned conjugate gradient algorithm. The next model estimation is then obtained by the formula

$$m_{k+1} = m_k + \alpha_k d_k,$$

where  $\alpha_k > 0$  is a step-size computed by a line-search technique ensuring a sufficient decrease of  $f$  at each iteration.

This method is generally able to solve the minimization problem (2). In some hard cases, however, the line-search technique fails to force convergence of the sequence  $\{m_k\}_{k \geq 0}$  to a solution. This difficulty may arise when the Hessian of  $f$  is very ill-conditioned and can often be overcome by using trust regions (see Conn et al. (2000)) instead of line-searches. The former method usually provides more stable and accurate results than the latter (Delbos et al. (2001), see also Sebudandi & Toint (1993)). In any case, we observe in practice that very few iterations are needed to get convergence, typically of the order of 10.

## 2.2 Formulation of the constrained problem

Let us now set down the formulation of the constrained seismic tomography problem. The constraints that can be introduced in the optimization problem could be nonlinear (for example we could force the impact points of some rays on a given interface to be located in a particular area) but, in this study, we limit ourselves to *linear* constraints. Even though linearity brings algorithmic simplifications, the optimization problem is difficult to solve because of the large number (up to  $10^4$ ) and the variety of the constraints. These may be of various types:

- constraints of different physical natures: on the velocities, on the interface depths, or on the derivatives of these quantities,
- equality or inequality constraints (examples: fixed value of the velocity gradient, minimal depth of an interface),

- local or global constraints (examples: local information coming from a well, interface slope in a particular region).

The constrained reflection tomography problem we consider is formulated as the regularized least-squares problem (2) subject to linear constraints:

$$\begin{cases} \min_{m \in \mathbb{R}^n} f(m) \\ C_E m = e \\ l \leq C_I m \leq u. \end{cases} \quad (6)$$

In this problem,  $C_E$  (resp.  $C_I$ ) is an  $n_E \times n$  (resp.  $n_I \times n$ ) matrix,  $e \in \mathbb{R}^{n_E}$ , and  $l, u \in \mathbb{R}^{n_I}$ . We note

$$n_C := n_E + n_I \quad \text{and} \quad n'_C := n_E + 2n_I.$$

It is said that an inequality constraint is *active* at  $m$  if it is satisfied with equality for this  $m$ . Determining which of the inequality constraints of (6) are active at a solution among the  $3^{n_I}$  possibilities turns out to be a major difficulty for the algorithms.

### 2.3 First order optimality conditions

Let  $\hat{m}$  be a local solution to (6). Since  $f$  is assumed to be differentiable and the constraints are linear (thus qualified), there exist  $\hat{\mu}_E \in \mathbb{R}^{n_E}$ ,  $\hat{\mu}_l \in \mathbb{R}^{n_I}$ , and  $\hat{\mu}_u \in \mathbb{R}^{n_I}$  such that the following Karush, Kuhn, and Tucker conditions (KKT) hold

$$\begin{cases} (a) \nabla f(\hat{m}) + C_E^\top \hat{\mu}_E + C_I^\top (\hat{\mu}_u - \hat{\mu}_l) = 0 \\ (b) C_E \hat{m} = e, \quad l \leq C_I \hat{m} \leq u \\ (c) (\hat{\mu}_l, \hat{\mu}_u) \geq 0 \\ (d) \hat{\mu}_l^\top (C_I \hat{m} - l) = 0, \quad \hat{\mu}_u^\top (C_I \hat{m} - u) = 0. \end{cases} \quad (7)$$

See for example Fletcher (1987) or Bertsekas (1995) for a presentation of the optimality theory of smooth problems. The vectors  $\hat{\mu}_E$ ,  $\hat{\mu}_l$ , and  $\hat{\mu}_u$  are called the Lagrange or KKT multipliers, and are associated with the equality and inequality constraints of (6). Condition (a) can also be written

$$\nabla_m \ell(\hat{m}, \hat{\mu}) = 0,$$

where  $\hat{\mu} := (\hat{\mu}_E, \hat{\mu}_l, \hat{\mu}_u)$  and the function  $\ell : \mathbb{R}^n \times \mathbb{R}^{n'_C} \mapsto \mathbb{R}$ , called the *Lagrangian* of problem (6), is defined by

$$\begin{aligned} \ell(m, \mu) = & f(m) + \mu_E^\top (C_E m - e) \\ & - \mu_l^\top (C_I m - l) + \mu_u^\top (C_I m - u). \end{aligned} \quad (8)$$

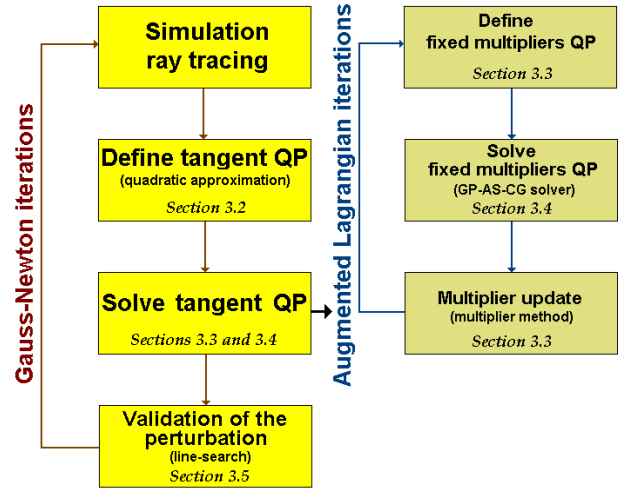
We note  $\hat{\mu}_I := \hat{\mu}_l - \hat{\mu}_u$ .

Equations (d) in (7) are known as the *complementarity conditions*. They express the fact that a multiplier  $\hat{\mu}_i$  associated with an inactive inequality constraint vanishes. For some problems, the converse is also true: active inequality constraints have positive multipliers. It is then said that *strict complementarity* holds at the solution:

$$\begin{aligned} l_i < C_i \hat{m} & \iff (\hat{\mu}_l)_i = 0 \\ C_i \hat{m} < u_i & \iff (\hat{\mu}_u)_i = 0. \end{aligned}$$

## 3 SOLVING THE CONSTRAINED SEISMIC REFLECTION TOMOGRAPHY PROBLEM

In this section we motivate and describe the optimization method used to solve (6) or its optimality conditions (7).



**Figure 2.** Operating diagram of the constrained optimization method.

A more detailed description is given by Delbos (2004). The operating diagram of the overall algorithm is presented in Figure 2 and can help the reader to follow the different levels of the approach.

### 3.1 Motivation for the chosen algorithmic approach

Presently, numerical methods to solve a nonlinear optimization problem like (6) can be gathered into two classes:

- the class of penalty methods, which includes the augmented Lagrangian approaches and the interior point (IP) approaches,
- the class of direct Newtonian methods, which is mainly formed of the sequential quadratic programming (SQP) approach.

Often, actual algorithms combine elements of the two classes, but their main features make them belonging to one of them.

In penalty methods, one minimizes a sequence of nonlinear functions, obtained by adding to the cost function in (6) terms penalizing more or less strongly the equality and/or inequality constraints as the iterations progress. For example, in the IP approaches, the inequality constraints are penalized in order to get rid of their combinatorial aspect, while the equality constraints are maintained, since they are easier to handle (see for instance Byrd et al. (2000) and the references therein). The iterations minimizing approximately each penalty function are usually based on the Newton iteration, which requires finding the solution to a linear system. Therefore the overall work of the optimization routine can be viewed as the one of solving a “sequence of sequences” of linear systems. This simplified presentation is mainly valid far from a solution, since close to a regular solution, a single Newton step is often enough to minimize sufficiently the penalty function (see Gould et al. (2000) for example). Now, each time a step is computed as a solution to a linear system, the nonlinear functions defining the optimization problem have to be evaluated in order to estimate the quality of the step. It is sensible to define an iteration of

the penalty approach as formed of a step computation and an evaluation of the nonlinear functions.

On the other hand, an SQP-like algorithm is a Newtonian method applied to the optimality conditions, equations (7) in our case (see part III in Bonnans et al. (2003) for example). Therefore, there is no sequence of nonlinear optimization problems to solve approximately, like in penalty methods. As a result, it is likely that such an approach will need less iterations to achieve convergence. Since, here also, the nonlinear functions defining the optimization problem need to be computed at each iteration to validate the step, it is likely that less nonlinear function evaluations are required with an SQP approach, in comparison with a penalty approach. Nevertheless, each SQP iteration is more complex, since it requires solving a *quadratic program* (QP), which is an optimization problem, with a quadratic cost function and linear equality and inequality constraints.

The discussion above shows that the choice of the class of algorithms strongly depends on the features of the optimization problem to solve. The key issue is to balance the time spent in the simulator (to evaluate the functions defining the nonlinear optimization problem) and in the optimization procedure (to solve the linear systems or the quadratic programs). In the unconstrained seismic reflection tomography problem, we have said in Section 2.1 that most of the CPU time is spent in the evaluation of the traveltimes (simulation) and that the Gauss-Newton algorithm converges in very few iterations (around 10). When choosing the algorithmic approach for the constrained version of the problem, we anticipated that the number of iterations will also be small with a Newton-like algorithm and that, in the current state of their development, IP algorithms will be unlikely to converge in so few iterations. This is our main motivation for developing an SQP-like algorithm: to keep as small as possible the number of nonlinear function evaluations. This strategy could be questioned with a ray tracing using massive parallelization; we leave this topic for future investigations.

### 3.2 A Gauss-Newton sequential quadratic programming approach

We have already mentioned that the sequential quadratic programming (SQP) algorithm is a Newton-like method applied to the optimality conditions of the nonlinear optimization problem under consideration, equations (7) in our case. In its standard form, it then benefits from a local quadratic convergence. Let us specify this algorithm for the constrained seismic reflection tomography problem.

The main work at the  $k$ th iteration of an SQP algorithm consists in solving the following *tangent quadratic program* (tangent QP) in  $d$  (see Chapter 13 and equation (13.4) in Bonnans et al. (2003)), in order to find the perturbation  $d_k$  to be given to  $m_k$ :

$$(QP_k) \quad \begin{cases} \min_{d \in \mathbb{R}^n} \left( F_k(d) := g_k^\top d + \frac{1}{2} d^\top H_k d \right) \\ C_E d = \tilde{e}_k \\ \tilde{l}_k \leq C_I d \leq \tilde{u}_k. \end{cases} \quad (9)$$

The cost function of this problem has a hybrid nature. Its linear part is obtained by using the gradient of the cost function of (6) at  $m_k$ , which is

$$g_k = J_k^\top (T_k - T^{obs}) + \sigma R m_k,$$

where we used the notation in (4). Its quadratic part should use, ideally, the Hessian of the Lagrangian  $\ell$  defined in (8), in order to get (quadratic) convergence. Since the constraints in (6) are linear, only the Hessian of  $f$  plays a role. Like for the unconstrained problem, we select the Gauss-Newton approximation  $H_k$  (see (5)) of this Hessian in order to avoid the computation of the expensive second derivatives of  $T$ . The constraints of (9) are obtained by the linearization of the constraints of (6) at  $m_k$ . Writing  $C_E(m_k + d) = e$  and  $l \leq C_I(m_k + d) \leq u$  leads to the constraints of (9) with

$$\tilde{e}_k := e - C_E m_k, \quad \tilde{l}_k := l - C_I m_k, \quad \text{and} \quad \tilde{u}_k := u - C_I m_k.$$

Let  $d_k$  be a solution to  $(QP_k)$ . Near a solution, the SQP algorithm updates the primal variables  $m_k$  by

$$m_{k+1} := m_k + d_k. \quad (10)$$

The SQP algorithm is actually a primal-dual method, since it also generates a sequence of multipliers  $\{\mu_k\} \in \mathbb{R}^{n'c}$ , which aims at approximating the optimal multiplier associated with the constraints of (6). These multipliers are updated by

$$\mu_{k+1} = \mu_k^{QP}, \quad (11)$$

where  $\mu_k^{QP}$  is the triple formed of the multipliers  $(\mu_k^{QP})_E$ ,  $(\mu_k^{QP})_l$ , and  $(\mu_k^{QP})_u$ , associated with the equality and inequality constraints of  $(QP_k)$ . Because of the linearity of the constraints, these multipliers do not intervene in the tangent QP, but they are useful for testing optimality and for the globalization of the algorithm (Section 3.5).

The following property of the SQP algorithm deserves being quoted for the discussion in Section 3.3. When strict complementarity holds at a “regular” primal-dual solution  $(\hat{m}, \hat{\mu})$  to (6) (see Section 2.3) and when the current iterate  $(m_k, \mu_k)$  is close enough to  $(\hat{m}, \hat{\mu})$ , the active constraints of the tangent QP are those that are active at the solution  $\hat{m}$  (see Theorem 13.2 in Bonnans et al. (2003)). Therefore, the difficult task of determining which constraints are active at the solution to  $(QP_k)$  disappears once  $m_k$  is close to  $\hat{m}$ , since the constraint activity is unchanged from one tangent QP to the next one.

The technique used to solve the tangent QP is essential for the efficiency of the SQP approach. In particular, because of the property mentioned in the previous paragraph, it should take benefit of an a priori knowledge of the active constraints. In the next two sections, we concentrate on this topic, which is represented by the right hand side blocks in Figure 2. We will come back to the globalization of SQP (the part of the algorithm that is depicted by the bottom block in the left hand side of Figure 2) in Section 3.5.

### 3.3 Solving the tangential quadratic problem by an augmented Lagrangian method

Because  $H_k$  is positive semidefinite (and usually positive definite), the tangent QP problem (9) is convex. Such a problem has been the subject of many algorithmic studies; we mention the following techniques:

- active set (AS),
- augmented Lagrangian (AL),
- interior points (IP).

Let us now motivate our choice of developing an AL algorithm to solve the QP in (9). The AS approach is often used to solve the QP's in the SQP algorithm. It has the advantage of being well defined, even when the problem is non-convex, and of being able to take advantage of an a priori knowledge of the active constraints at the solution. However, since this algorithm updates the active set one constraint at a time, it suffers from being rather slow when the active set is not correctly initialized and when there are many inequality constraints. For large problems, this can be a serious drawback and we have discarded this method for that reason. The IP algorithms are very efficient to solve convex QP's but, presently, they have some difficulty in taking benefit of a good knowledge of the active constraints as this is often the case after a few iterations of the SQP algorithm. On the other hand, the inherent ill-conditioning of the linear systems they generate and the necessity to use here iterative methods to solve them have appeared to us as deterrent factors.

The AL approach that we have implemented goes back to Hestenes (1969) and Powell (1969). This is a well established methodology, designed to solve nonlinear optimization problems, although its properties for minimizing a convex QP does not seem to have been fully explored (see Delbos & Gilbert (2005)). It is adapted to large problems, since it can be implemented in such a way that it does not need any matrix factorization (Fortin & Glowinski (1983) and Glowinski & Le Tallec (1989)). In the context of seismic tomography problems, a version of the AL algorithm has been proposed by Glowinski & Tran (1993) to solve the tangent QP of the SQP method. The present contribution takes inspiration from that paper and goes further by improving the efficiency of its augmented Lagrangian QP solver. Let us detail the approach.

The QP (9) is first written in an equivalent form, using an auxiliary variable  $y \in \mathbb{R}^{n_I}$  (we drop the index  $k$  for simplicity):

$$(QP') \quad \begin{cases} \min_{(d,y) \in \mathbb{R}^n \times \mathbb{R}^{n_I}} F(d) \\ C_E d = \tilde{e}, \quad C_I d = y \\ \tilde{l} \leq y \leq \tilde{u}. \end{cases}$$

Next, the AL associated with the equality constraints of that problem is considered. This is the function  $\mathcal{L}_r : \mathbb{R}^n \times \mathbb{R}^{n_I} \times \mathbb{R}^{n_C} \mapsto \mathbb{R}$  defined at  $d \in \mathbb{R}^n$ ,  $y \in \mathbb{R}^{n_I}$ , and  $\lambda = (\lambda_E, \lambda_I) \in \mathbb{R}^{n_C}$  by

$$\begin{aligned} \mathcal{L}_r(d, y, \lambda) &= F(d) + \lambda_E^\top (C_E d - \tilde{e}) + \frac{r}{2} \|C_E d - \tilde{e}\|^2 \\ &\quad + \lambda_I^\top (C_I d - y) + \frac{r}{2} \|C_I d - y\|^2. \end{aligned}$$

The scalar  $r > 0$  is known as the *augmentation parameter*.

The precise statement of our version of the AL algorithm to solve (9) or (QP') can now be given: see Algorithm 1. This method, in particular the update of the multipliers by (13), has a nice interpretation in terms of the proximal point algorithm in the dual space (see Rockafellar (1973)). It is not essential to give this interpretation here, but this one is very useful for proving the properties of the method, including its convergence. In this theory, the augmentation parameter  $r$  in the definition of  $\mathcal{L}_r$  is viewed as a step-size, damping the modification of the multipliers in (13). The factor of  $r = r^j$  in this formula is indeed a negative subgradient of the

---

**Algorithm 1.** An AL algorithm to solve (9).

---

**data:**  $\lambda^0 \in \mathbb{R}^{n_C}$  and  $r^0 > 0$ ;

**begin**

**for**  $j = 0, 1, 2, \dots$  **do**

**if**  $(C_E d^j \simeq \tilde{e})$  &  $(C_I d^j \simeq y^j)$  **then return;**  
     by Algorithm 3, find a solution  $(d^{j+1}, y^{j+1})$  to

$$\begin{cases} \min_{(d,y)} \mathcal{L}_{r^j}(d, y, \lambda^j) \\ \tilde{l} \leq y \leq \tilde{u}; \end{cases} \quad (12)$$

**if** (12) is solved **then**

$$\begin{cases} \lambda_E^{j+1} := \lambda_E^j + r^j (C_E d^{j+1} - \tilde{e}) \\ \lambda_I^{j+1} := \lambda_I^j + r^j (C_I d^{j+1} - y^{j+1}) \end{cases} \quad (13)$$

**else**

$$\lambda^{j+1} := \lambda^j;$$

**end**

    choose a new augmentation parameter  $r^{j+1} > 0$  by Algorithm 2;

**end**

**end**

---

dual function at  $\lambda^{j+1}$ . Note that these step-sizes  $r^j$  can now change at each iteration of Algorithm 1, while the penalty approach behind the definition of  $\mathcal{L}_r$  makes the possibility of such a change less natural. On the other hand, it can be shown that the algorithm converges if (9) has a solution and if the sequence  $\{r^j\}_{j \geq 0}$  is chosen bounded away from zero. If, in addition,  $r^j$  is chosen larger than some positive Lipschitz constant  $L$  (usually unknown, unfortunately), the norm of the equality constraints converges globally linearly to zero: this is inequality (14) below, on which we will come back. Actually, the larger are the augmentation parameters  $r^j$ , the faster is the convergence. The only limitation on a large value for  $r^j$  comes from the ill-conditioning that such a value induces in the AL and the resulting difficulty or impossibility to solve (12). This is why a test for updating the multipliers by (13) has been introduced. For ensuring convergence, the test prevents the multipliers from being updated when (12) is not correctly solved (a heuristic less restrictive than this test is used by Delbos (2004)).

Algorithm 1 is a dual method, since it essentially monitors the dual sequence  $\{\lambda^j\}_{j \geq 0}$ ; the augmentation parameters  $r^j$  and the primal variables  $(d^{j+1}, y^{j+1})$  are viewed as auxiliary quantities. The choice of the initial multiplier  $\lambda^0$  depends on the outer SQP iteration index. When  $k = 0$ , Algorithm 1 simply takes  $\lambda^0 = 0$ , unless an estimate of the optimal multiplier is provided. When  $k > 0$ , Algorithm 1 takes for  $\lambda^0$  the dual solution to the previous QP. A similar strategy is used for setting  $r^0$ : when  $k = 0$ ,  $r^0$  is set to an arbitrary value (the effect of taking  $r^0 = 1$  or  $r^0 = 10^4$  is tested for the 3D real data set in Section 4.2) and, when  $k > 0$ ,  $r^0$  is set to the value of  $r^j$  at the end of the previous QP.

Algorithm 1 can be viewed as transforming (9) into a sequence of bound constrained convex quadratic subproblems of the form (12). These subproblems have a solution, as soon as (9) has a solution (see Proposition 3.3 in Delbos & Gilbert (2005) for a weaker condition). Clearly, the major part of the CPU time required by Algorithm 1 is spent in solving the bound constrained subproblems (12). We describe an algorithm for doing this efficiently in Section 3.4: Algorithm 3.

Two facts contribute to the observed success of this method. First, a bound constrained QP is much easier to solve than (9), which has general linear constraints (see Moré & Toraldo (1991) and the references therein). Second, because of its dual and constraint convergence, the AL algorithm usually identifies the active constraints of (9) in a finite number of iterations. Since often these active constraints are stable when  $m_k$  is in some neighborhood of a solution, the combinatorial aspect of the bound constrained QP's rapidly decreases in intensity as the convergence progresses (and usually disappears after very few AL iterations).

We have already made it clear that the choice of the augmentation parameters  $r^j$  is crucial for the efficiency of Algorithm 1. Two pitfalls have to be avoided: a too small value slows down the convergence, a too large value makes it difficult to find a solution to (12). It is usually not easy to determine an a priori appropriate value for the augmentation parameter, so that updating  $r^j$  in the course of the AL iterations by observing the behavior of the algorithm looks better. In our implementation of Algorithm 1, the update of  $r^j$  at iteration  $j \geq 1$  is done by a heuristic close to the one given in Algorithm 2. This one deserves some explanations.

---

**Algorithm 2.** A heuristics for updating  $r^j$  in Algorithm 1.

---

**data:**  $r^{j-1}$ ,  $r^j$ ,  $\rho^j$ ,  $\rho_{\text{des}}$ ,  $\kappa^{j-1}$ , and  $\kappa^j$ ;  
**begin**  
   **if** (12) is solved **then**  
      $r^{j+1} := r^j$ ;  
 1   **if**  $\rho^j > \rho_{\text{des}}$  **then**  $r^{j+1} := r^j \rho^j / \rho_{\text{des}}$ ;  
   **else**  
 2    $r^{j+1} := r^j / 10$ ;  
     **if** ( $r^j < r^{j-1}$ ) & ( $\kappa^j > \kappa^{j-1}$ ) **then**  
 3     **stop** [failure of Algorithm 1]  
   **end**  
**end**

---

A too small value of  $r^j$  can be deduced from the observation of  $\rho^j := \nu^{j+1} / \nu^j$ , where  $\nu^j := \|(C_E d^j - \tilde{e}, C_I d^j - y^j)\|$  is the Euclidean norm of the equality constraints of (QP'). It is indeed known from Theorem 4.5 of Delbos & Gilbert (2005) that there is a constant  $L > 0$  such that, for all  $j \geq 1$ , there holds

$$\rho^j \leq \min \left( 1, \frac{L}{r^j} \right). \quad (14)$$

This estimate is valid provided (12) is solved exactly. If such is the case and if  $\rho_{\text{des}} \in (0, 1)$  is a given desired convergence rate for the constraint norm, a value  $\rho^j > \rho_{\text{des}}$  is then an incitement to increase the augmentation parameter. This update of  $r^j$  is done in statement 1 of Algorithm 2.

On the other hand, a difficulty to solve (12) can be detected by the impossibility to satisfy the optimality conditions of that problem at a given precision in a given number of preconditioned conjugate gradient (CG) iterations (the algorithm to solve (12), based on CG iterations, is explained in Section 3.4). The heuristics has also to decide whether a failure to solve (12) is due to a too large value of  $r^j$ , in which case decreasing  $r^j$  is appropriate (statement 2), or to an intrinsic excessive ill-conditioning of the problem, in

which case a definitive failure is declared (statement 3). For this, it uses the sensitivity to  $r^j$  of an estimate  $\kappa^j$  of the condition number of the preconditioned Hessian

$$M_{r^j} := P_{r^j}^{-1/2} Q_{r^j} P_{r^j}^{-1/2},$$

where  $Q_{r^j} := H + r^j (C_E^\top C_E + C_I^\top C_I)$  is the Hessian with respect to  $d$  of the criterion  $\mathcal{L}_{r^j}$  of (12) and  $P_{r^j} (\simeq Q_{r^j})$  is a preconditioner for  $Q_{r^j}$ . The estimate  $\kappa^j$  makes use of the Rayleigh quotients of  $M_{r^j}$  computed during the preconditioned CG iterations. According to Proposition 2.3 of Fortin & Glowinski (1983), the condition number of  $Q_{r^j}$  grows asymptotically linearly with  $r^j$ . The same law does not hold for  $M_{r^j}$ , since hopefully  $r^j$  intervenes in  $P_{r^j}$ . Nevertheless, it is important to have an estimate of the condition number of  $M_{r^j}$ , not of  $Q_{r^j}$ , since it is  $M_{r^j}$  that governs the performance of the CG algorithm. In view of this comments, it seems reasonable to say that, if a preceding decrease of the augmentation parameter,  $r^j < r^{j-1}$ , has resulted in an increase of the condition number estimate,  $\kappa^j > \kappa^{j-1}$ , it is likely that a new decrease of  $r^j$  will not improve the conditioning of problem (12). In that case and if it is not possible to solve (12) with  $r^j$ , a decision to stop is taken in statement 3 of Algorithm 2; the problem is declared to be too hard to solve.

The actual heuristics for updating  $r^j$  in our implementation of the AL algorithm has other safeguards, detailed by Delbos (2004), but the logic is essentially the one presented in Algorithm 2. Experiments with two different initial values  $r^0$  of the augmentation parameter are shown in Section 4.2, illustrating the behavior of the heuristics adapting  $r^j$ .

To conclude the description of Algorithm 1, we still need to say a word on its stopping criterion and to specify the value of the multiplier  $\mu_k^{\text{QP}}$  used in (11). There are many ways of showing that the stopping criterion makes sense. The shortest one here is probably to observe that a solution  $(d^{j+1}, y^{j+1})$  to (12) satisfying  $C_E d^{j+1} = \tilde{e}$  and  $C_I d^{j+1} = y^{j+1}$  is actually a solution to (QP');  $d^{j+1}$  is then a solution to (9). Finally, the optimality conditions of problems (9) and (QP') show that one can take

$$\begin{aligned} (\mu_k^{\text{QP}})_E &= \lambda_E^{j+1}, \\ (\mu_k^{\text{QP}})_I &= \max(0, \lambda_I^{j+1}), \quad \text{and} \quad (\mu_k^{\text{QP}})_u = \max(0, -\lambda_I^{j+1}), \end{aligned}$$

where  $\lambda^{j+1}$  is the value of the multiplier on return from Algorithm 1 at the  $k$ th iteration of the SQP algorithm.

### 3.4 Solving the Lagrange problem by the GP-AS-CG algorithm

Problem (12) is solved in our software by a combination of the gradient projection (GP) algorithm, the active set (AS) method, and conjugate gradient (CG) iterations. This GP-AS-CG algorithm is a classical and efficient method for minimizing a large scale bound constrained convex quadratic function, see Moré & Toraldo (1991), Friedlander & Martínez (1994), Nocedal & Wright (1999), and the references therein. We adapt it below to the special structure of problem (12), in which the variables  $d$  and  $y$  intervene differently in the objective and only  $y$  is constrained.

A brief description of the three ingredients of the GP-AS-CG algorithm is necessary for the understanding of the discussion below. The AS method solves (12) by maintaining

fixed a varying choice of variables  $y_i$  to their bounds, while minimizing the objective with respect to the other variables, which are supposed to satisfy the constraints. Each time the minimization would lead to a violation of some bounds, a displacement to the boundary of the feasible set is done and some new variables  $y_i$  are fixed to their bounds. The minimization is pursued in this way up to complete minimization with respect to the remained free variables or (in our case) up to the realization of a Rosen-like stopping criterion. The GP algorithm intervenes at this point to inactivate a bunch of erroneously fixed variables and, possibly, to activate others. This GP-AS algorithm proceeds up to finding a solution to (12). Finally, CG iterations are used to minimize the objective on the faces of the feasible set that are activated by the GP-AS algorithm.

Minimizing the objective of (12) in  $(d, y)$  jointly can be inefficient, in particular when there are many inequality constraints in the original problem (6), since then the presence of the auxiliary variable  $y$  increases significantly the number of unknowns. Our first adaptation of the GP-AS-CG algorithm consists in setting up a minimization in  $d$  only, while  $y$  is adapted to follow the change in  $d$ . Let us clarify this. Suppose that  $W \subset I$  is the *working set* at a given stage of the algorithm, that is the set of indices  $i$  of the variables  $y_i$  that are fixed at one of the bounds  $\tilde{l}_i$  or  $\tilde{u}_i$ . We note  $C := (C_E^T \ C_I^T)^T$ ,  $V := I \setminus W$ ,  $\bar{W} := E \cup W$ , and denote by  $C_V$  (resp.  $C_{\bar{W}}$ ) the matrix obtained from  $C_I$  (resp. from  $C$ ) by selecting its rows with index in  $V$  (resp. in  $\bar{W}$ ). For the current working set  $W$ , the algorithm has to solve (we drop the index  $j$  of the AL algorithm)

$$\min_{(d, y_V)} \mathcal{L}_r(d, y, \lambda).$$

with implicit bound constraints on  $y_V \in [\tilde{l}_V, \tilde{u}_V]$  and with  $y_W$  fixed. The optimality conditions of this problem can be written

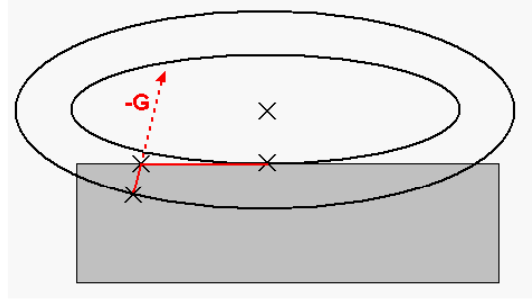
$$\begin{aligned} (H + rC^T C)d - rC_V^T y_V &= -g - C^T \lambda + rC_E^T \tilde{e} + rC_W^T y_W, \\ -rC_V d + r y_V &= \lambda_V. \end{aligned} \quad (15)$$

Substituting the value of  $y_V$  given by (15) into the first equation gives

$$(H + rC_W^T C_W)d = -g - C_W^T \lambda_W + rC_E^T \tilde{e} + rC_W^T y_W. \quad (16)$$

This is the equation that is solved by the preconditioned CG algorithm. During this process,  $y_V$  is updated so that equation (15) is continually verified. It can be shown that the directions modifying  $(d, y)$  are conjugate in  $\mathbb{R}^n \times \mathbb{R}^{n_I}$  with respect to the Hessian matrix of  $\mathcal{L}_r(\cdot, \cdot, \lambda)$ , so that this method can be viewed as a conjugate direction algorithm that maintains the iterates in the affine subspace defined by equation (15).

In this process, as soon as  $y_V$  hits a new bound, the working set  $W$  is enlarged. Then a new CG process is started on the updated equations (16) and (15), from the  $(d, y)$  obtained so far. Note that the updated (15) is verified at the beginning of this new process, since it is obtained by deleting equations from (15) and since (15) was verified at the end of the previous CG process. We will see that (15) is also verified at the end of a GP phase of the algorithm, so that this equation makes no difficulty to be maintained all along the GP-AS-CG algorithm, provided this one starts by a GP phase.



**Figure 3.** A 2D view of one iteration of the gradient projection (GP) algorithm: the ellipses are the level curves of the objective function, the vector  $-G$  is its negative gradient at the current iterate, and the shadow box represents the feasible set.

The goal of the gradient projection (GP) phase of the algorithm is to change drastically the working set if this one is felt to be very different from the set of the active constraints at the solution. As explained below, the GP phase is actually an adapted version of a single iteration of the GP algorithm (see Bertsekas (1995) for example); more iterations would be useless in our case, as we will see. Its property mentioned above, which is observed in practice, is then also supported by the fact that the GP algorithm identifies the active constraints at the solution in a finite number of iterations when strict complementarity holds.

An iteration of the standard GP algorithm forces the decrease of the objective of (12) along the piecewise linear path obtained by projecting the steepest descent path on the feasible set (see Figure 3). If  $P_{[\tilde{l}, \tilde{u}]}$  stands for the projection operator on  $[\tilde{l}, \tilde{u}]$  in  $\mathbb{R}^{n_I}$ , the projected steepest descent path emanating from the current iterate  $(d, y)$  is the mapping

$$p : (\alpha > 0) \mapsto \begin{pmatrix} d - \alpha g_d \\ P_{[\tilde{l}, \tilde{u}]}(y - \alpha g_y) \end{pmatrix},$$

where  $g_d$  (resp.  $g_y$ ) is the gradient of  $\mathcal{L}_r$  with respect to  $d$  (resp.  $y$ ). There holds

$$g_y = -\lambda_I - r(C_I d - y).$$

In the standard GP algorithm, a local minimum or a step-size ensuring a sufficient decrease of the complex piecewise quadratic function  $\alpha \mapsto \mathcal{L}_r(p(\alpha), \lambda)$  is usually taken. Because of the particularly simple structure of  $\mathcal{L}_r$  in  $y$ , we prefer maintaining  $d$  fixed and minimizing completely

$$\alpha \mapsto \mathcal{L}_r(d, P_{[\tilde{l}, \tilde{u}]}(y - \alpha g_y), \lambda).$$

This is our second adaptation of the standard GP-AS-CG algorithm. It has the following interesting property. Because the Hessian of  $\mathcal{L}_r$  with respect to  $y$  is a multiple of the identity matrix, the new  $y$  is the projection on  $[\tilde{l}, \tilde{u}]$  of the unconstrained minimizer of  $\mathcal{L}_r(d, \cdot, \lambda)$ :

$$y := P_{[\tilde{l}, \tilde{u}]} \left( C_I d + \frac{\lambda_I}{r} \right). \quad (17)$$

Observe that, if  $W$  is now set to  $\{i : y_i = \tilde{l}_i \text{ or } \tilde{u}_i\}$ , equation (15) is satisfied, as claimed above.

Algorithm 3 summarizes our version of the GP-AS-CG algorithm. The use of Rosen's stopping test for the CG iterations and other algorithmic details are not mentioned. See

Delbos (2004) for further information on the implementation.

---

**Algorithm 3.** The GP-AS-CG algorithm to solve (12).

---

**data:**  $r, \lambda, d := 0$ ;  
**begin**  
 GP := true;  
**while** optimality of (12) is not satisfied **do**  
   **if** GP **then**  
     compute  $y$  by (17);  
     update  $W := \{i : y_i = \tilde{l}_i \text{ or } \tilde{u}_i\}$  and  $V := I \setminus W$ ;  
     GP := false;  
   **else**  
     **while**  $\tilde{l}_V < y_V < \tilde{u}_V$  **do**  
       use CG iterations to solve (16) in  $d$ , while  
       updating  $y_V$  to satisfy (15);  
     **end**  
     update the index sets  $W$  and  $V$ ;  
     **if**  $W$  has not been enlarged **then** GP := true;  
   **end**  
**end**  
**end**

---

### 3.5 Globalization by line-search

It is now time to remember that the constrained minimization problem (6) we want to solve is nonlinear and that, just as in unconstrained optimization, the update of the model  $m_k$  by (10), where  $d_k$  is the computed solution to the quadratic problem (QP <sub>$k$</sub> ), is unlikely to yield convergence if the initial estimate  $m_0$  is not close enough to a solution. For forcing convergence from a remote starting model, we follow the standard globalization technique presented in Chapter 15 of Bonnans et al. (2003), which uses an exact penalty merit function. Using a filter method would have been an alternative, but we did not try it. We have also implemented a line-search technique, since the combination of trust regions with a QP approximately solved by an augmented Lagrangian algorithm is a technique that does not seem to have been explored. Due to its usefulness to solve the unconstrained tomography problem, we plan to investigate this possibility in a future research.

We use the exact penalty function  $\Theta_\tau : \mathbb{R}^n \rightarrow \mathbb{R}$  defined by

$$\Theta_\tau(m) = f(m) + \Psi_\tau(m),$$

where

$$\begin{aligned} \Psi_\tau(m) = & \sum_{i \in E} \tau_i |C_i m - e_i| \\ & + \sum_{i \in I} \tau_i \max(l_i - C_i m, 0, C_i m - u_i), \end{aligned}$$

in which the  $\tau_i$ 's are penalty weights. Exactness of the penalization means that a solution to (6) is also a minimizer of  $\Theta_\tau$ . To get that important property, the  $\tau_i$ 's need to be large enough, although finite. It is usual to update them at some iteration, so that they always satisfy

$$\begin{cases} \tau_i \geq |(\mu_k^{\text{QP}})_i| + \bar{\tau}, & \text{for } i \in E \\ \tau_i \geq \max(|(\mu_k^{\text{QP}})_i|, |(\mu_k^{\text{QP}})_u|) + \bar{\tau}, & \text{for } i \in I, \end{cases} \quad (18)$$

where  $\mu_k^{\text{QP}}$  is defined after (11) and  $\bar{\tau} > 0$  is a small security threshold.

In our case, the convexity of the QP (9) defining  $d_k$  and the inequalities (18) ensure that  $\Theta_\tau$  decreases along  $d_k$ . A line-search along this direction is then possible. A step-size  $\alpha_k > 0$ , typically determined by backtracking, can be found such that for some small constant  $\omega \in (0, 1/2)$ :

$$\Theta_\tau(m_k + \alpha_k d_k) \leq \Theta_\tau(m_k) + \alpha_k \omega \Delta_k, \quad (19)$$

where  $\Delta_k := \nabla f(m_k)^\top d_k - \Psi_\tau(m_k)$  can be shown to be negative when  $m_k$  is not stationary. Now the model and the multiplier are updated by

$$\begin{cases} m_{k+1} = m_k + \alpha_k d_k \\ \mu_{k+1} = \mu_k + \alpha_k (\mu_k^{\text{QP}} - \mu_k). \end{cases} \quad (20)$$

Algorithm 4 summarizes the overall algorithm to solve problem (6).

---

**Algorithm 4.** The overall algorithm to solve (6).

---

**data:**  $m_0, \mu_0$ ;  
**begin**  
   **for**  $k = 0, 1, 2, \dots$  **do**  
     **if** optimality of (6) holds at  $(m_k, \mu_k)$  **then stop**;  
     compute  $(d_k, \mu_k^{\text{QP}})$  a primal-dual solution to (12) by Algorithm 1;  
     update  $\tau$  to satisfy (18);  
     determine a step-size  $\alpha_k > 0$  by backtracking, in order to satisfy (19);  
     update  $(m_{k+1}, \mu_{k+1})$  by (20);  
   **end**  
**end**

---

## 4 APPLICATIONS OF THE CONSTRAINED REFLECTION TOMOGRAPHY ON REAL DATA SETS

### 4.1 2D PP/PS data set

In this section, we present an application of constrained reflection tomography to one 2D line of a 3D 4C OBC (Ocean Bottom Cable) survey with PP and PS data from bp. Broto et al. (2003) have already interpreted and studied this data set using an unconstrained inversion method. The velocity model is described by four velocity layers and five interfaces (cf Figure 4 left). The isotropic assumption was satisfying until the last layer (layer which contains the last two interfaces h4 and h5). By applying the anisotropic inversion methodology of Stopin (2001) on the last layer, they obtained a model that fits the traveltimes better than any of the previous isotropic models and that, in addition, has more reliable velocity variations. Two parameters  $(\eta, \delta)$  describe the velocity anisotropy:  $\eta$  can be seen as a measure of the an-ellipticity, whereas  $\delta$  controls the near vertical velocity propagation of the P-waves (Stopin (2001)).

The value of the  $\delta$  anisotropy parameter has been obtained by a trial and error approach in order to match approximately the h5 depth given by well logs. Actually, the under-determination of the inverse problem does not allow the joint inversion of the velocities, interfaces and

**Table 1.** Inversion results of the unconstrained inversion.

| Layer 4 | RMS of traveltimes misfits | Depth mismatch at well location | $\eta$ | $\delta$ |
|---------|----------------------------|---------------------------------|--------|----------|
| h4-PP   | 3.6ms                      | 190m                            | 6.2%   | 2%       |
| h5-PP   | 4.6ms                      | 150m                            |        |          |
| h5-PS   | 9.8ms                      |                                 |        |          |

**Table 2.** Inversion results of the constrained inversion.

| Layer 4 | RMS of traveltimes misfits | Depth mismatch at well location | $\eta$ | $\delta$ |
|---------|----------------------------|---------------------------------|--------|----------|
| h4-PP   | 6.3ms                      | 0m                              | 8.2%   | 15.9%    |
| h5-PP   | 3.9ms                      | 0m                              |        |          |
| h5-PS   | 8.1ms                      |                                 |        |          |

anisotropy parameters ( $\delta$  parameter is particularly undetermined, Stopin (2001)). This applied methodology is obviously not optimal. Indeed, the manual tuning of the anisotropy parameters requires a lot of time: an important numbers of anisotropic inversion with different pairs ( $\eta, \delta$ ) have to be performed before getting a satisfying result. Secondly, it is very hard to make this method accurate: we note a discrepancy of 150 meters for the reflector depth h5 compared to the depth given by the well logs data. Finally, it turned out impossible to determine the anisotropy parameter  $\delta$  so that both the reflector depths of h4 and h5 given by the well logs are reasonably matched.

The solution we propose here is to compute a model using our constrained inversion method in order to fit the reflector depths given by the well, while considering ( $\eta, \delta$ ) as additional unknowns to determine. This consists in the inversion of P and S velocity variations, of the geometries of interfaces h4 and h5 and of the two anisotropy parameters, i.e., inversion of 1024 parameters from 32468 picked traveltimes (associated with reflections of PP waves and PS waves on h4 and h5).

In Tables 1 and 2, we have respectively summed up the results of the final models obtained with the unconstrained and constrained inversions. The final model (Figure 4 right) of the constrained inversion matches the traveltimes data with the same accuracy than the result obtained by the unconstrained inversion (Figure 4 left), and it strictly verifies the reflector depths given by the well logs. This solution has been obtained after only 9 nonlinear SQP iterations.

We see that, the introduction of constraints at wells for the two reflectors h4 and h5 reduces the under-determination and allows for a joint inversion of velocities, interfaces, and anisotropy parameters. The resulting model matches the traveltimes and well data. The values of its anisotropy parameters are very different from those obtained with the unconstrained inversion (Tables 1 and 2).

## 4.2 3D PP data set

During the European KIMASI project, reflection tomography was applied on a 3D North Sea data set from bp (Ehinger et al. (2001)). A P-velocity model was obtained thanks to a top-down layer-stripping approach where lateral and vertical velocity variations within Tertiary, Paleocene,

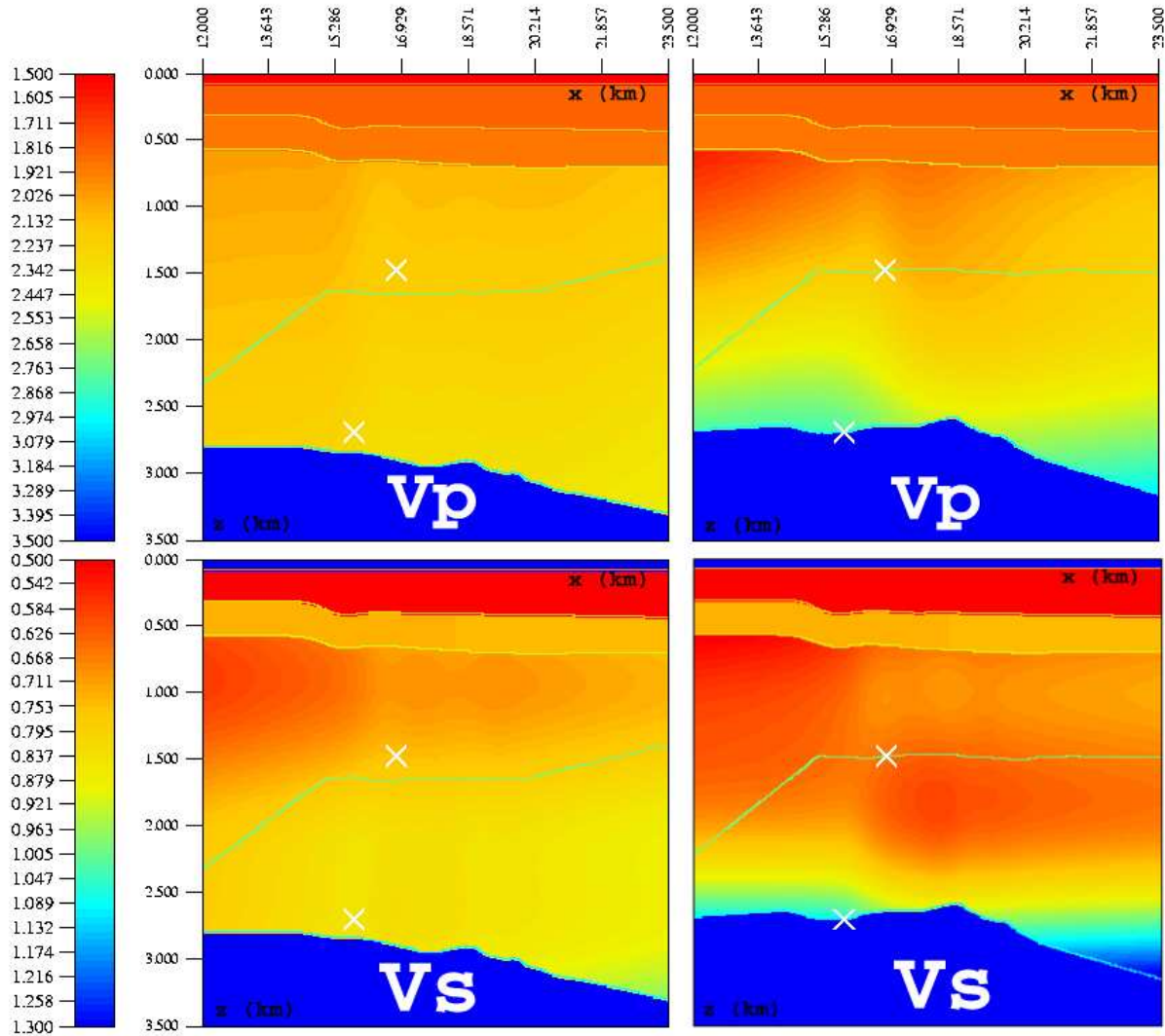
and Cretaceous units (this last layer being divided in two velocity layers) have been determined sequentially. A strong velocity under-determination in the upper Tertiary layer was detected during the inversion process due to the large layer thickness (2.5km) and to the very poor ray aperture (despite the introduction of the intermediary reflector named layer1). Several velocity parameterizations (Table 3) were inverted and led to solution models that match traveltimes data with the same accuracy. These different tests are time consuming (each test is a whole inversion) and the reflector depths given by well data are not well retrieved, this information being not explicitly introduced in the inversion process. One of the resulting model is presented in Figure 5.

To obtain a model consistent with well data, we propose to apply our developed constrained tomography inversion. The interface depths are constrained at 5 well locations and we constrain the range of variations of the vertical velocity gradient in the Tertiary layer thanks to well measurements (Table 4). A global inversion of the 4 velocity layers is preferred to the layer-stripping approach: it avoids bad data fitting for deeper layers due to errors in shallow layers. The simultaneous inversion of all the layers is guided by constraints on layer thicknesses to avoid any non-physical interface intersections: Figure 6 shows an example of non-admissible model obtained by global unconstrained inversion (it presents non-physical interface intersections that make layers vanish in the pointed region and thus a large number of rays are discarded).

The experiment then consists in a global inversion of 127 569 traveltimes for 5960 unknowns describing 4 velocity layers and 5 interfaces, subject to 2300 constraints (Table 4). The constraints on the velocity variations (resp. on the layer thicknesses) are applied on a grid of 10x10x10 (resp. 20x20 points). The results are presented in Figure 7: the obtained model matches the data with the same accuracy as the models obtained by Ehinger et al (2001) and verifies all the introduced constraints (see Tables 4 and 3). The model obtained by unconstrained inversion (Figure 5) and the model obtained by constrained inversion (Figure 7) are very different: by the geometry of the interfaces and by the velocity variations within the layers. The introduction of constraints leads to local velocity variations at well locations that may perhaps be attenuated by a stronger regularization.

The total number of conjugate gradient iterations for each Gauss-Newton step (the total number of conjugate gradient iterations takes into account all the iterations of augmented Lagrangian method) is less than  $10^4$  (less than twice the number of unknowns), which looks like a very good result for a problem with 2300 constraints. In this experiment, only 6 nonlinear SQP iterations are necessary to reach convergence.

The automatic adaptation of the augmentation parameter  $r^j$  (see Algorithm 2 in Section 3.3) is illustrated in Figure 8. The initial value  $r^0$  can be chosen in a large interval. Algorithm 2 modifies  $r^j$  in order to obtain a good convergence rate of the multipliers in the AL algorithm (Algorithm 1), without deterioration of the conditioning of the bound constrained problem (12).



**Figure 4.** The velocity models ( $V_p$  and  $V_s$ ) on the left hand side are computed with the unconstrained inversion method; those on the right hand side are computed with the constrained inversion method. The white crosses locate reflector depths measured from the deviated well logs, which are imposed as constraints. These additional constraints allows the software to determine the anisotropy parameters in the same run.

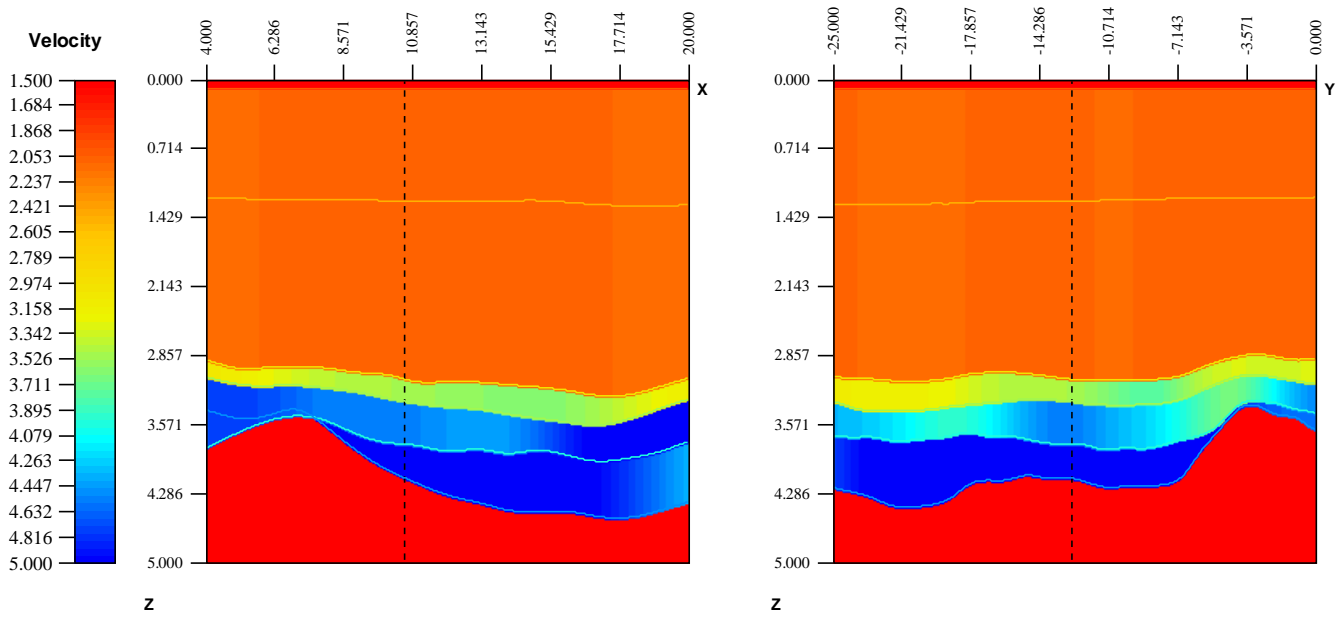
| Tertiary velocity parameterization |                            | RMS of traveltimes misfits |                  | Mean depth mismatch at the 5 well locations |
|------------------------------------|----------------------------|----------------------------|------------------|---|
| Velocity type                      | Vertical velocity gradient | Layer 1                    | Top of Paleocene |   |
| $V_0(x, y)$                        | fixed to 0/s               | 4.3ms                      | 4.2ms            | 96m   |
| $V_0(x, y) + 0.2z$                 | fixed to 0.2/s             | 3.4ms                      | 4.6ms            | 300m  |
| $V(x, y, z)$                       | without constraints        | 4.1ms                      | 4.4ms            | 100m  |
|                                    | with constraints           | 3.9ms                      | 4.4ms            | 0m  |

**Table 3.** The different velocity parameterizations tested for the inversion of the Tertiary layer (in the layer-stripping approach applied by Ehinger et al. (2001)) for the unconstrained inversion and in a global constrained approach (last line of the table).

## 5 CONCLUSION

Reflection tomography often requires the introduction of additional a priori information on the model in order to reduce the under-determination of the inverse problem. A nonlinear optimization method that can deal with linear constraints on

the model has been developed. This dedicated method has proved its efficiency on various concrete inversions, including those related to the two real data sets presented in this paper. The number of Gauss-Newton iterations has the same order of magnitude than the number of Gauss-Newton iterations necessary in the unconstrained case ( $\sim 10$  iterations).



**Figure 5.** Velocity model (slices along  $x$  (left) and along  $y$  directions at one of the 5 well locations) obtained with a layer-stripping approach using the unconstrained reflection tomography. This model corresponds to the  $v(x, y, z)$  velocity parameterization (see Table 3). The RMS value of the traveltimes misfits is 6.1ms.

**Table 4.** Description of the equality and inequality constraints introduced in the inversion. We succeed to obtain a model that matches the data and the 2300 constraints.

| Constraints                                 |                                      | Model obtained with the UNCONSTRAINED inversion | Model obtained with the CONSTRAINED inversion |
|---|--------------------------------------|---|---|
| Mean depth mismatch at the 5 well locations | tpal                                 | 96m   | 0m  |
|   | tchalk                               | 132m  | 0m  |
|   | bchalk                               | 140m  | 0m  |
| Vertical velocity gradient in Tertiary      | $0.1 < k < 0.3/s$                    | $k=0/s$   | $k \sim 0.18/s$                               |
| Velocity range                              | $2.5 < v_{pal} < 4 \text{ km/s}$     | ok  | ok  |
|   | $3.5 < v_{chalk} < 5.7 \text{ km/s}$ | ok  | ok  |
|   | $4.2 < v_{chalk} < 5.8 \text{ km/s}$ | ok  | ok  |

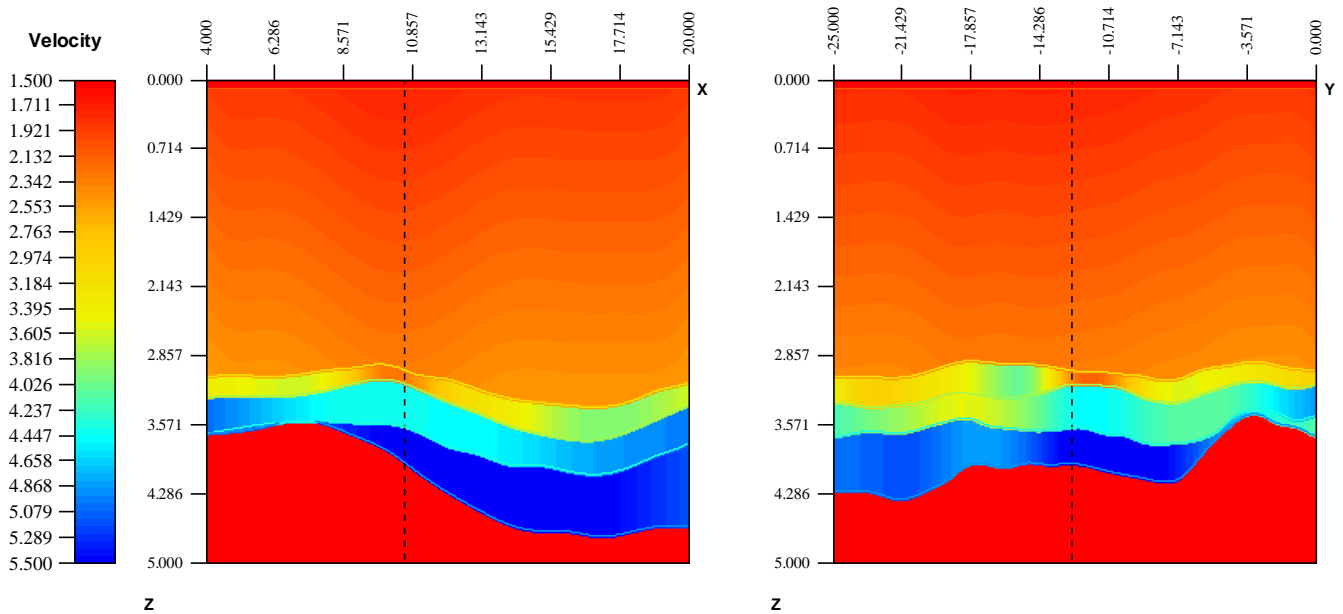
This and the fact that solving the forward problem is time consuming were preponderant factors in favor of a sequential quadratic programming approach, instead of a nonlinear interior point method. The chosen constraint activation method is efficient even for a large number of constraints. At last, the algorithm developed for updating the augmentation parameter discharges the user of the software from such a technical concern and allows an adequate choice of its value in terms of the convergence rate of the augmented Lagrangian algorithm.

## ACKNOWLEDGMENTS

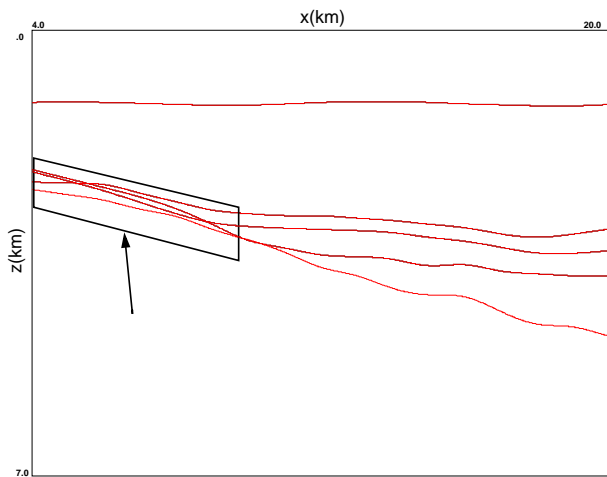
We would like to thank Guy Chavent and Roland Masson for helpful discussions and advises at various stages of the elaboration of this work. We acknowledge Karine Broto and Anne Jardin for their precious help for the applications on real datasets. The datasets studied in this paper were kindly provided by bp.

## REFERENCES

- Bertsekas, D., 1995. *Nonlinear Programming*, Athena Scientific, (second edition, 1999).
- Bishop, T. N., Bube, K. P., Cutler, R. T., Langan, R. T., Love, P. L., Resnick, J. R., Shuey, R. T., Spindler, D. A., & Wyld, H. W., 1985. Tomographic determination of velocity and depth in laterally varying media, *Geophysics*, **50**(6), 903–923.
- Bonnans, J., Gilbert, J. C., Lemaréchal, C., & Sagastizábal, C., 2003. *Numerical Optimization – Theoretical and Practical Aspects*, Springer Verlag, Berlin.
- Broto, K., Ehinger, A., & Kommedal, J. H., 2003. Anisotropic traveltimes tomography for depth consistent imaging of PP and PS data, *The Leading Edge*, pp. 114–119.
- Bube, K. P. & Langan, R. T., 1994. A continuation approach to regularization for traveltimes tomography, *64<sup>th</sup> Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts*, pp. 980–983.
- Bube, K. P. & Meadows, M. A., 1999. The null space of a generally anisotropic medium in linearized surface reflection tomography, *Geophys. J. int.*, **139**, 9–50.
- Byrd, R., Gilbert, J. C., & Nocedal, J., 2000. A trust region method based on interior point techniques for nonlinear programming, *Mathematical Programming*, **89**, 149–185.
- Červený, V., 1987. Ray tracing algorithms in three-dimensional



**Figure 7.** Velocity model (slices along  $x$  (left) and along  $y$  directions at one of the 5 well locations) obtained with a global inversion using the constrained reflection tomography (constraints described in Table 4). The RMS value of the traveltime misfits is 6.5ms.



**Figure 6.** Interfaces (slice along  $x$ ) obtained with a global inversion using the unconstrained reflection tomography. Difficulties are encountered during the Gauss-Newton iterations. Some iterates lead to non-admissible models in which the forward operator is not defined. For instance, the model found at iteration 7 presents non-physical intersections that make layers vanish in the pointed region and thus a large number of rays are discarded. The resulting discontinuity of the cost function leads to convergence troubles of the Gauss-Newton method.

laterally varying layered structures, in *Seismic Tomography*, edited by G. Nolet, pp. 99–133, D. Reidel.

Chauvier, L., Masson, R., & Sinoquet, D., 2000. Implementation of an efficient preconditioned conjugate gradient in jerry, *KIM Annual Report, Institut Français du Pétrole, Rueil-Malmaison, France*, <http://consortium.ifp.fr/KIM/>.

Conn, A., Gould, N., & Toint, P., 2000. *Trust-Region Methods*, MPS/SIAM Series on Optimization 1, SIAM and MPS.

Courtier, P. & Talagrand, O., 1987. Variational assimilation of meteorological observations with the adjoint vorticity equation.

II: Numerical results, *Quarterly Journal of the Royal Meteorological Society*, **113**, 1329–1347.

Courtier, P., Thépaut, J., & Hollingsworth, A., 1994. A strategy for operational implementation of 4D-Var, using an incremental approach, *Quarterly Journal of the Royal Meteorological Society*, **120**, 1367–1387.

de Boor, C., 1978. *A Practical Guide to Splines*, Springer.

Delbos, F., 2004. *Problème d'optimisation de grande taille avec contraintes en tomographie de réflexion 3D*, Ph.D. thesis, Université Pierre et Marie Curie, Paris VI, France.

Delbos, F. & Gilbert, J. C., 2005. Global linear convergence of an augmented Lagrangian algorithm for solving convex quadratic optimization problems, *Journal of Convex Analysis*, to appear.

Delbos, F., Sinoquet, D., Gilbert, J. C., & Masson, R., 2001. A trust-region Gauss-Newton method for reflection tomography, *KIM Annual Report, Institut Français du Pétrole, Rueil-Malmaison, France*, <http://consortium.ifp.fr/KIM/>.

Delprat-Jannaud, F. & Lailly, P., 1993. Ill-posed and well-posed formulations of the reflection travel time tomography problem, *J. Geophys. Res.*, **98**, 6589–6605.

Dennis, J., Gay, D., & Welsch, R., 1981. An adaptive nonlinear least squares algorithm, *ACM Transactions on Mathematical Software*, **7**, 348–368.

Ehinger, A., Broto, K., Jardin, A., & the KIMASI team, 2001. 3D tomographic velocity model determination for two North Sea case studies, in *Expanded Abstracts*, EAGE.

Fletcher, R., 1987. *Practical Methods of Optimization*, John Wiley & Sons, Chichester, 2nd edn.

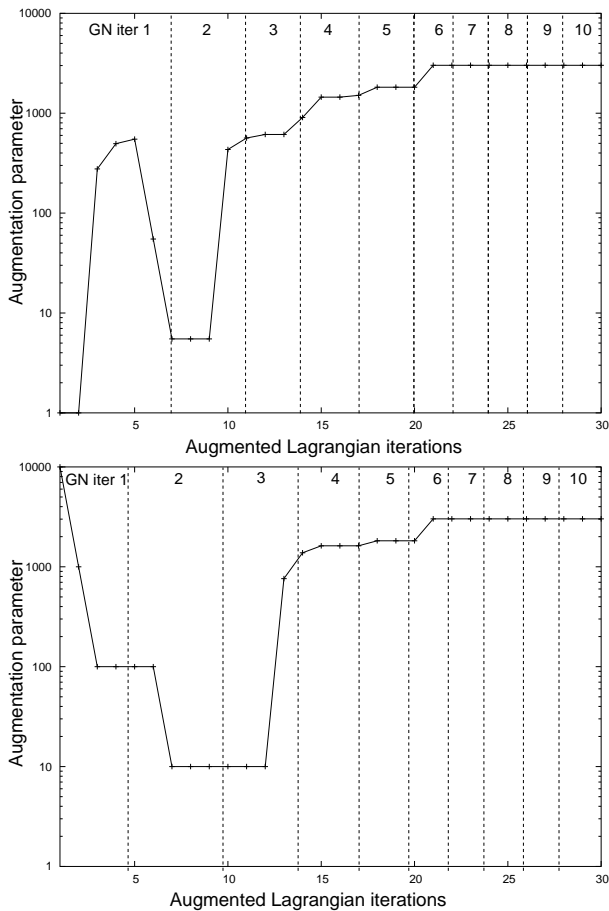
Fortin, M. & Glowinski, R., 1982. *Méthodes de Lagrangien Augmenté – Applications à la Résolution Numérique de Problèmes aux Limites*, no. 9 in Méthodes Mathématiques de l'Informatique, Dunod, Paris.

Fortin, M. & Glowinski, R., 1983. *Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems*, North-Holland, Amsterdam.

Friedlander, A. & Martínez, J. M., 1994. On the maximization of a concave quadratic function with box constraints, *SIAM Journal on Optimization*, **4**, 177–192.

Gauss, 1809. *Theoria motus corporum coelestium*.

Glowinski, R. & Le Tallec, P., 1989. *Augmented Lagrangian and*



**Figure 8.** Automatic adaptation of augmentation parameter  $r^j$ : two experiments are performed, with an initial augmentation parameter either set to  $r^0 = 1$  (top) or to  $r^0 = 10^4$  (bottom). Algorithm 2 adapts  $r^j$  during the AL iterations. It may decrease its value to improve the conditioning of problem (12) or increase  $r^j$  to speed up the convergence of the constraint norm to zero (the desired convergence rate  $\rho_{\text{des}}$  is fixed to  $10^{-3}$ ).

*Operator Splitting Methods in Nonlinear Mechanics*, SIAM, Philadelphia.

Glowinski, R. & Tran, Q., 1993. Constrained optimization in reflection tomography: the Augmented Lagrangian method, *East-West J. Numer. Math.*, **1**, 213–234.

Gould, N., Orban, D., Sartenaer, A., & Toint, P., 2000. Super-linear convergence of primal-dual interior point algorithms for nonlinear programming, *Mathematical Programming*, **87**, 215–249.

Hansen, P. C., 1992. Analysis of discrete ill-posed problems by mean of the L-curve, *SIAM Review*, **34**, 561–580.

Hestenes, M., 1969. Multiplier and gradient methods, *Journal of Optimization Theory and Applications*, **4**, 303–320.

Inoue, H., 1986. A least-squares smooth fitting for irregularly spaced data: finite element approach using the cubic B-splines basis, *Geophysics*, **51**, 2051–2066.

Jurado, F., Lailly, P., & Ehinger, A., 1998. Fast 3D two-point raytracing for traveltimes tomography, *Proceedings of SPIE, Mathematical Methods in Geophysical Imaging V*, **3453**, 70–81.

Lailly, P. & Sinoquet, D., 1996. Smooth velocity models in reflection tomography for imaging complex geological structures, *Geophys. J. Int.*, **124**, 349–362.

Moré, J. & Toraldo, G., 1991. On the solution of large quadratic

programming problems with bound constraints, *SIAM Journal on Optimization*, **1**, 93–113.

Nocedal, J. & Wright, S., 1999. *Numerical Optimization*, Springer Series in Operations Research, Springer, New York.

Powell, M., 1969. A method for nonlinear constraints in minimization problems, in *Optimization*, edited by R. Fletcher, pp. 283–298, Academic Press, London.

Rockafellar, R., 1973. A dual approach to solving nonlinear programming problems by unconstrained optimization, *Mathematical Programming*, **5**, 354–373.

Sebudandi, C. & Toint, P. L., 1993. Non linear optimization for seismic traveltimes tomography, *Geophysics Journal International*, **115**, 929–940.

Stopin, A., 2001. *Détermination de modèle de vitesses anisotropes par tomographie de réflexion des modes de compression et de cisaillement (in English)*, Ph.D. thesis, Université de Strasbourg I, France, <http://consortium.ifp.fr/KIM/>.

Tikhonov, A. & Arsenin, V., 1977. *Solution of ill-posed problems*, Winston.

Yabe, H. & Yamaki, N., 1995. Convergence of a factorized Broyden-like family for nonlinear least squares problems, *SIAM Journal on Optimization*, **5**, 770–790.