INNER PRODUCT PRECONDITIONED TRUST-REGION METHODS FOR FREQUENCY-DOMAIN FULL WAVEFORM INVERSION*

XAVIER ADRIAENS†, LUDOVIC MÉTIVIER‡, AND CHRISTOPHE GEUZAINE§

Abstract. Full waveform inversion is a seismic imaging method which requires to solve a large-scale minimization problem, typically through local optimization techniques. Most local optimization methods can basically be built up from two choices: the update direction and the strategy to control its length. In the context of full waveform inversion, this strategy is very often a line search. We here propose to use instead a trust-region method, in combination with non-standard inner products which act as preconditioners. More specifically, a line search and several trust-region variants of the steepest descent, the limited memory BFGS algorithm and the inexact Newton method are presented and compared. A strong emphasis is given to the inner product choice. For example, its link with preconditioning the update direction and its implication in the trust-region constraint are highlighted. A first numerical test is performed on a 2D synthetic model then a second configuration, containing two close reflectors, is studied. The latter configuration is known to be challenging because of multiple reflections. Based on these two case studies, the importance of an appropriate inner product choice is highlighted and the best trust-region method is selected and compared to the line search method. In particular we were able to demonstrate that using an appropriate inner product greatly improves the convergence of all the presented methods and that inexact Newton methods should be combined with trust-region methods to increase their convergence speed.

Key words. numerical optimization, large-scale inverse problems, trust-regions methods, operator preconditioning, seismic imaging, full waveform inversion.

AMS subject classifications. 35R30, 65K10, 86-08, 49M15, 90C06

1. Introduction.

Full waveform inversion is a high-resolution seismic imaging technique formulated as a data fitting problem, whose aim is to recover some model parameters by minimizing the discrepancy between recorded data and data simulated by solving wave propagation problems [29, 35]. By nature these data are oscillatory and consequently the misfit quantifying the discrepancy features local minima [4, 21]. Global optimization techniques should ideally be used but the typically very high dimensions of the search space prohibits their use and only local optimization tools can practically be employed, with care [8]. A straightforward direction to iteratively update the model properties is of course the gradient, *i.e.* the direction of steepest decrease. However it is well-known that the inverse Hessian plays a crucial role in the reconstruction in addition to offering the possibility to account for coupling effects between parameter classes for multi-parameter inversion [3, 24, 26, 29, 38]. A theoretically simple way to incorporate these second-order derivatives is to minimize the misfit using Newton methods. In practice however the pure Newton method is too computationally intensive to implement, because it requires inverting the Hessian operator. In addition, the misfit is not necessarily quadratic, thus the exact Newton direction is not necessarily appropriate. Consequently, it is natural to turn to inexact Newton methods, where the search direction is constructed iteratively to approximate the pure Newton direction, or to quasi-Newton methods. State-of-the-art methods rely on the quasi-Newton l-BFGS algorithm, which implicitly builds an approximation of the inverse Hessian operator from l previously saved gradients and model parameters [23]. However it has been illustrated that on some specific cases involving multiple re-

*Submitted to the editors March 8, 2022.

Funding: This research was funded by the Fonds de la Recherche Scientifique de Belgique (F.R.S.-FNRS) and by the ARC "WAVES" grant 15/19-03 from the Wallonia-Brussels Federation of Belgium. Computational resources were provided by the Consortium des Équipements de Calcul Intensif (CÉCI), funded by the F.R.S.-FNRS and by the Walloon Region.

†F.R.S-FNRS, Department of Electrical Engineering and Computer Science, Université de Liège, Belgium (xavier.adriaens@uliege.be).

‡Univ. Grenoble Alpes, CNRS, LJK, F-38000 Grenoble, France (ludovic.metivier@univ-grenoble-alpes.fr).

§Department of Electrical Engineering and Computer Science, Université de Liège, Belgium (cgeuzaine@uliege.be).

flections, such quasi-Newton methods fail to converge where inexact Newton methods do succeed [20]. The latter compute the descent direction through a few iterations of a linear system involving the Hessian operator (the Newton system). One advantage over l-BFGS is the locality of the quadratic approximation: such methods do not rely on the convergence history of the algorithm, which might yield inaccurate inverse Hessian approximation for non quadratic misfit functions. The bottleneck of these methods lies in the compromise to find between a direction built in few iterations, but which hardly takes the Hessian into account and a nearly exact direction which is very expensive to compute. A complementary strategy to reduce this number of inner iteration is to apply a preconditioner to both sides of the Newton system [6, 16, 25, 36]. To implement any of the three above mentioned schemes, one can rely either on line search algorithms, or on trust-region methods. In the former case, once a direction is chosen, the outer iteration is completed by finding the optimal length of the step that should be performed along that direction. Among the non linear optimization community, it is sometimes argued however that line search is not well suited with Newton directions, especially when the Hessian is nearly singular. Indeed when the Hessian is nearly singular, the Newton direction becomes excessively long such that the local quadratic approximation implicitly made when computing it ceases to hold. Much computational effort must then be made by the line search procedure to reduce the step size [23]. Stopping the iterative solution of the Newton system earlier appears as a solution to this problem. For example, its convergence requirements could be relaxed such that they reflect the accuracy of the local quadratic approximation [9, 19]. Alternatively, a trust-region method could be used instead [18, 37, 39, 40]. The latter limits the length of the update direction depending on the accuracy of the local quadratic approximation. The length of a direction is given by its norm, itself induced by the inner product chosen for the model parameters space. The choice of this inner product is thus pivotal in the implementation of a trust-region method. Moreover changing the inner product modifies both the gradient and the Hessian and is equivalent to applying a preconditionner [7, 12, 22, 17, 41]. Consequently it also has a major impact on line search based local optimization methods.

In this paper, we tackle the three following important questions:

- Which descent direction to compute: the gradient, the l-BFGS direction or an inexact Newton direction?
- Which globalization method to select: a line search method or a trust-region method?
- Which preconditioning strategy to apply? How to enforce it?

Answering these three questions and determining the good combinations (good practices) between them is crucial for effective full waveform inversion. From our study, it appears that preconditioning is essential and that enforcing preconditioning through the inner product is elegant and, more interestingly, implies no modification to the practical implementation of the optimization algorithms. The l-BFGS method is found to be the most efficient method for the considered single-parameter inversions. It is also found to be insensitive to the globalization choice. Inexact Newton methods should not be discarded though, as considering the exact Hessian might lead to better model parameter decoupling in the case multi-parameter inversions. When using inexact Newton methods, our case studies show that using a trust region globalization consistently improves convergence.

The paper is organized as follows. In the first part, full waveform inversion is stated very generally. The optimization problem and its solution procedures using either a line search or a trust-region are introduced. The Newton system, which is pivotal in local minimization theory, is also derived. A particular emphasis is given to the inner product choice. More specifically, its link with preconditioning the Newton system is established. Local minimization methods commonly used in the context of full waveform inversion are then recalled. In the second part, the application

to acoustic imaging is detailed. The (adjoint) procedure to compute gradients and Hessian vector products is given and its computational cost is explained. The overall computational cost of each optimization method is then deduced. Finally, convergence results on the acoustic Marmousi case study are analyzed to determine the best inner product and the best parameters for a trust-region method. This best candidate is then compared to line search methods on both the Marmousi model and on a case study involving strong reflectors.

2. Local optimization methods. Full wave inversion is an imaging method based on the minimization of a misfit functional J, which exclusively depends on some model parameters m. The recovered model parameters m^* are defined as the minimizer of this misfit, *i.e.* $m^* = \arg\min J(m)$. Local optimization techniques are based on a local quadratic expansion of the misfit J around the current model estimate

$$J(m+\delta m) \approx J(m) + \{D_m J\}(\delta m) + \frac{1}{2}\{D^2_{mm} J\}(\delta m, \delta m). \tag{2.1}$$

This expansion can also be written in terms of the gradient j' and the Hessian operator H once an inner product $\langle \cdot, \cdot \rangle_M$ is chosen for the model space M

$$J(m+\delta m) \approx J(m) + \langle j', \delta m \rangle_M + \frac{1}{2} \langle H\delta m, \delta m \rangle_M . \tag{2.2}$$

The pure Newton direction p_N is then defined as the minimizer of this local quadratic expansion, which is also the solution of a linear system

$$p_N = \arg\min_{p \in M} J(m) + \langle j', p \rangle_M + \frac{1}{2} \langle Hp, p \rangle_M \quad \text{or} \quad Hp_N = -j'. \tag{2.3}$$

The large-scale nature of this linear system requires either the use of approximate Hessian operators that are straightforward to invert, or the use of Hessian-free iterative methods. Both approaches are usually referred to as quasi-Newton methods and inexact Newton methods. In the latter case, the conjugate gradient method is the ideal candidate for the iterative solver because the Hessian operator is symmetric. The conjugate gradient method is however designed for positive definite operators while the full Hessian can be indefinite, especially far from the global minimum [29, 20]. As a consequence, either an additional safeguard is added to exit prematurely when directions of negative curvature are encountered or the exact Hessian is modified such that it becomes positive definite, *e.g.* using the Gauss-Newton approximation [25].

2.1. Globalization methods. As mentioned in the introduction, the misfit is not necessarily quadratic and thus the pure Newton direction or its approximations are not always the best directions. For that reason the length of the search direction is often tweaked using a line search or a trust-region method, which ensures convergence towards the nearest local minimum [9, 11, 10, 23].

2.1.1. Line search. When using a line search procedure, a direction p must first be identified. An appropriate length γ is then given to this direction p, ideally the global minimum along the line $m + \gamma p$. In practice however less stringent satisfactory conditions are used instead to spare expensive wave problem resolutions. Maybe the best example are strong Wolfe conditions

$$J(m+\gamma p) \leq J(m) + c_1 \gamma \{D_m J(m)\}(p) \tag{2.4}$$

$$|\{D_m J(m+\gamma p)\}(p)| \leq c_2 |\{D_m J(m)\}(p)| \tag{2.5}$$

for some constant c_1 and c_2 such that $0 < c_1 < c_2 < 1$. The first condition is called the sufficient decrease condition. It ensures that updating the model in the direction γp

produces a decrease smaller than a fraction c_1 of what is expected from a local linear approximation of the misfit. The second condition, called the curvature condition, ensures that the updated model $m+\gamma p$ is sufficiently close to a local minimum along the line, where the directional derivative $\{D_m J(m+\gamma p)\}(p)$ would be zero. When this derivative is very smaller (resp. larger) than zero, then a larger (resp. smaller) step could produce a significantly bigger decrease. We choose here a line search algorithm that satisfies strong Wolfe conditions and accepts steps easily (Algorithm 3.2 from [23] with $c_1 = 10^{-4}$, $c_2 = 0.9$). The outer loop is finally obtained by repeating these two steps iteratively until convergence.

2.1.2. Trust region. At the opposite when using a trust-region method, first a maximum length Δ is chosen. Then the best approximate solution, meaning the direction that minimizes a local prediction of the misfit but smaller than this length, is used

$$p = \underset{p\in M, \|p\|_M \leq \Delta}{\arg\min} \left[J^{\text{pred}}(m;p) := J(m) + \langle j'(m), p\rangle_M + 0.5 \left\langle \tilde{H}(m)p, p \right\rangle_M \right]. \tag{2.6}$$

This local misfit prediction J^{pred} is typically constructed based on the local quadratic approximation (2.2) through a particular choice of some approximate Hessian operator $\tilde{H}$. Of course the approximate Newton direction $\tilde{H}p = -j'$ is the solution of this problem if it lies inside the trust region. There are several possibilities to choose this length Δ and our particular choice is detailed later. More importantly, as we pointed out in the introduction, the length constraint is formulated in terms of the norm induced by the inner product $\|p\|_M^2 = \langle p, p\rangle_M \leq \Delta^2$. Modifying this inner product therefore changes the shape of the trust region and it is then desirable to choose it carefully. The size of the trust region is actually controlled by the outer iterations. The decision of modifying the trust region is based on the accuracy of the local prediction of the misfit. When the prediction is accurate but the updates are limited by the length constraint, then the trust region radius is increased. At the opposite, when the updates are out of the range of validity of the prediction, then the trust region radius is decreased. The decrease (resp. increase) rate of the radius is controlled by some parameter $c_0 < 1$ (resp. $c_1 > 1$). The quality of the prediction is quantified by the ratio between the actual decrease $\delta J_{\text{a}} := J(m_n) - J(m_{n+1})$ and the decrease predicted by the local prediction of the misfit. There are two ways to compute this predicted decrease [10]. On the one hand the expansion can be written in terms of the gradient and the Hessian operator at the previous model estimate

$$J(m_{n+1}) = J(m_n + p_n) \tag{2.7}$$

$$\approx J(m_n) + \langle j'(m_n), p_n\rangle_M + 0.5 \left\langle \tilde{H}(m_n)p_n, p_n \right\rangle_M = J^{\text{pred}}(m_n; p_n) \tag{2.8}$$

which defines the prospective predicted decrease

$$\delta J_{\text{p,p}} := J(m_n) - J^{\text{pred}}(m_n; p_n) \tag{2.9}$$

$$= -\langle j'(m_n), p_n\rangle_M - 0.5 \left\langle \tilde{H}(m_n)p_n, p_n \right\rangle_M. \tag{2.10}$$

On the other hand, it can also be written in terms of the gradient and the Hessian operator at the next model estimate

$$J(m_n) = J(m_{n+1} - p_n)$$

$$\approx J(m_{n+1}) - \langle j'(m_{n+1}), p_n\rangle_M + 0.5 \left\langle \tilde{H}(m_{n+1})p_n, p_n \right\rangle_M = J^{\text{pred}}(m_{n+1}; -p_n)$$

which defines the retrospective predicted decrease

$$\delta J_{\mathrm{p,r}} := J^{\mathrm{pred}}(m_{n+1}; -p_n) - J(m_{n+1}) \tag{2.11}$$

$$= -\left\langle j'(m_{n+1}), p_n \right\rangle_M + 0.5 \left\langle \tilde{H}(m_{n+1}) p_n, p_n \right\rangle_M . \tag{2.12}$$

These ratios between the actual decrease and one of both the predicted decreases $\rho_{\mathrm{p}} := \delta J_{\mathrm{a}}/\delta J_{\mathrm{p,p}}$ and $\rho_{\mathrm{r}} := \delta J_{\mathrm{a}}/\delta J_{\mathrm{p,r}}$ are actually both equal to one when the approximate Hessian in the update direction and the second order expansion (2.2) are exact. When the misfit is not quadratic or the Hessian approximation is not accurate, then these ratios can go away from one. Using anything else than the full Newton method can degrade these ratios, even if the misfit is quadratic. In particular for a pure quadratic misfit, neglecting the negative definite part of the Hessian makes the prospective ratio bigger than one ($\delta J_{\mathrm{p,p}}$ is underestimated) and the retrospective ratio smaller than one ($\delta J_{\mathrm{p,r}}$ is overestimated).

Standard trust-region methods directly control the radius Δ. However it is an absolute quantity, in the sense that it is compared to $\|p\|_M$, which depends on the inner product. Thus, it seems more natural to control this radius relatively to the gradient norm ($\Delta = \mu \|j'\|_M$), which provides a length reference for the (approximate) Newton system. In this way, even when the (approximate) Newton system changes scale from one iteration to another, the trust region remains relevant. This particular variant (Algorithm 2.1) has been first introduced in [11].

Algorithm 2.1 Fan trust-region algorithm

Require: retrospective or prospective, $0 \le \rho_0 < \rho_1 < 1$ and $0 < c_0 < 1 < c_1$
  $\mu_0 = 1$
  loop
    $\Delta_n = \mu_n \|j'(m_n)\|_M$
    $p_n = \begin{cases} -\mu_n j'_n \\ \text{(2.28) with } \Delta = \Delta_n \\ \text{Algorithm 2.5 with } \Delta = \Delta_n \end{cases}$
    $\delta J_{\mathrm{a}} = J(m_n) - J(m_n + p_n)$ and $\delta J_{\mathrm{p,p}} = J(m_n) - J^{\mathrm{pred}}(m_n; p_n)$
    $\rho_{\mathrm{p}} = \delta J_{\mathrm{a}}/\delta J_{\mathrm{p,p}}$
    if $\rho_{\mathrm{p}} \ge \rho_0$ **then** $m_{n+1} = m_n + p_n$ **else** $m_{n+1} = m_n$
    if prospective **or** $\rho_{\mathrm{p}} < \rho_0$ **then**
      $\rho = \rho_{\mathrm{p}}$
    else if retrospective **then**
      $\delta J_{\mathrm{p,r}} = J^{\mathrm{pred}}(m_{n+1}; -p_n) - J(m_{n+1})$
      $\rho = \rho_{\mathrm{r}} = \delta J_{\mathrm{a}}/\delta J_{\mathrm{p,r}}$
    end if
    if $\rho < \rho_1$ **then** $\mu_{n+1} = c_0 \mu_n$
    else if $\rho \ge \rho_1$ **and** $\|p_n\|_M > 0.5\Delta_n$ **then** $\mu_{n+1} = c_1 \mu_n$
    else **then** $\mu_{n+1} = \mu_n$
  end loop

According to this algorithm, a direction p_n is rejected when the prospective misfit prediction J_n^{pred} used to compute it is not accurate, in the sense that the prospective ratio is smaller than some threshold ρ_0. If not rejected, then the trust region size is updated according to either the prospective or the retrospective ratio, based on a comparison with a second threshold ρ_1. Because the updated radius Δ_{n+1} constrains the direction search around the next model estimate m_{n+1}, it makes sense to use the retrospective ratio which also involves the next model estimate m_{n+1} and not the prospective ratio which involves the current model estimate m_n. Using the retrospective ratio is however slightly more expensive because the next (approximate) Hessian operator in the current direction must be computed in addition. Moreover

the accuracy of the retrospective prediction might be good in the direction $-p_n$ while still being bad in the direction p_{n+1} and inversely. There is also no safeguards for large value of the ratios, which means that when the model is not accurate but the predicted decrease underestimates the true decrease, the radius can still be increased. Three sets of values for the threshold ρ_1 and the rates c_0/c_1 have been tested. The acceptance threshold ρ_0 is always tiny such that steps are often accepted, similarly to the line search algorithm.

(A) $\rho_0 = 10^{-4}$, $\rho_1 = 0.25$ and $c_0 = 0.20$, $c_1 = 5$.
(B) $\rho_0 = 10^{-4}$, $\rho_1 = 0.75$ and $c_0 = 0.25$, $c_1 = 2$.
(C) $\rho_0 = 10^{-4}$, $\rho_1 = 0.90$ and $c_0 = 0.50$, $c_1 = 2$.

The first one (A) is very similar to what was originally proposed in [10]. The other two (B,C) are more cautious because they modify the radius more rarely and when they do, it increases by a smaller factor. Note that the second one (B) is also close to what is proposed in [23].

2.2. Inner product. The choice of the inner product plays a central role in the inversion because it defines through the norm how directions length are measured but also because it defines both gradients and Hessian operators. Indeed the equivalence between both expansions (2.1) and (2.2) is granted by the defining property of the gradient and the Hessian operator in terms of directional derivatives

$$\langle j', \delta m_1 \rangle_M := \{D_m J\}(\delta m_1) \qquad \forall \delta m_1, \tag{2.13}$$

$$\langle H\delta m_2, \delta m_1 \rangle_M := \left\{D^2_{mm} J\right\}(\delta m_1, \delta m_2) \quad \forall \delta m_1,\, \delta m_2. \tag{2.14}$$

This link between directional derivatives and kernels is actually a straightforward application of the Fréchet-Riesz representation theorem [15].

The model parameter space is a function space defined on some region Ω and conventionally, the inner product is chosen as the $L_2(\Omega)$ inner product

$$\langle m_2, m_1 \rangle_M = \langle m_2, m_1 \rangle := \int_\Omega m_1(\boldsymbol{x}) m_2(\boldsymbol{x})\, d\Omega. \tag{2.15}$$

This straightforward choice leads to the conventional gradient j'_{L_2} and the conventional Hessian operator H_{L_2}, that can both be computed efficiently using the adjoint state method [1, 13, 28]. As an illustration, a conventional gradient is represented in Fig. 1b. It is actually the first gradient computed during the acoustic imaging of the Marmousi model. As can be seen, shallow contributions have much greater amplitudes than deeper parts. This actually reflects the bad scaling properties of this inner product and motivates the use of a spatially weighted inner product

$$\langle m_2, m_1 \rangle_M := \left\langle m_2 \sqrt{w}, \sqrt{w}\, m_1 \right\rangle, \tag{2.16}$$

with an appropriate spatially dependant weight w. Insights on how to design w can be gained by relating the conventional and the weighted gradients. Indeed, both are defined by (2.13) then by transitivity of the equality

$$\left\langle j'_{L_2}, \delta m_1 \right\rangle = \left\langle j' \sqrt{w}, \sqrt{w}\, \delta m_1 \right\rangle \quad \forall \delta m_1 \quad \text{such that} \quad j' = w^{-1} j'_{L_2}. \tag{2.17}$$

The same reasoning can be applied to both Hessian operators ($H = w^{-1} H_{L_2}$). Choosing this weight close to the Hessian operator then makes the gradient closer to the pure Newton direction and the Hessian operator closer to the identity. In other words, the Newton system (2.3) is better conditioned and iterative solvers are therefore expected to converge faster. We choose here to take this weight as the diagonal part of the Gauss-Newton Hessian ($w = \operatorname{diag}(H_{\mathrm{GN}})$) because it can be computed semi-analytically for a given model at no extra computational cost under certain circumstances [25]. A weight that has the same units than the Hessian also has the advantage

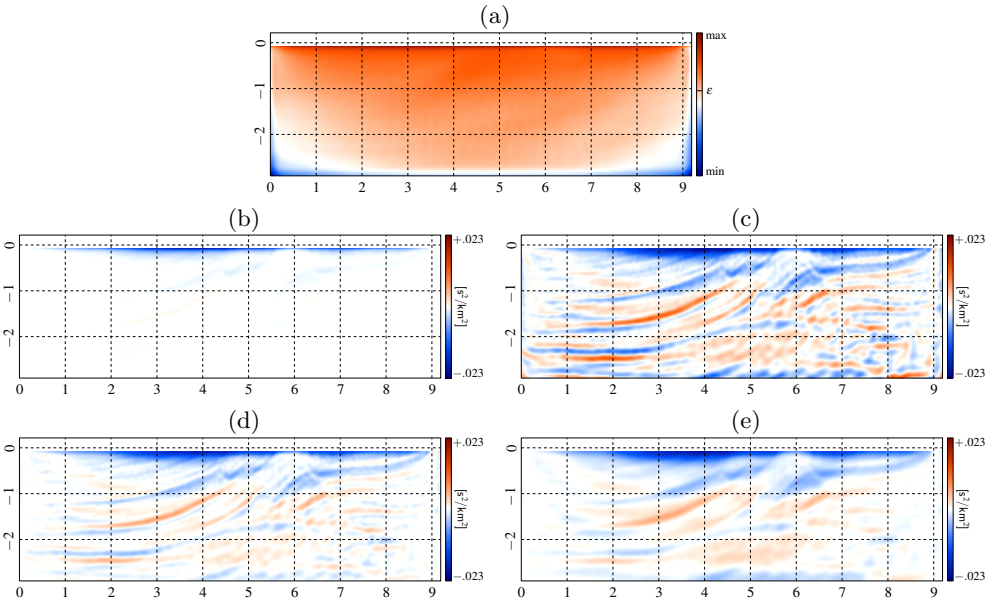

Fig. 1: Diagonal part of the Gauss-Newton Hessian (a). Conventional gradient (b). Weighted gradient (c). Weighted and thresholded gradient (d). Weighted and smoothed gradient (e). The stabilization parameter ϵ is given graphically in the top figure and a smoothing length $2\pi l_c = 0.250$ [km]) is chosen.

that the corresponding weighted gradient has the same units than the model parameters. Model parameters, weighted gradients and weighted Hessian vector products therefore all have the units of model parameters and the coefficients between them, for example the length γ and μ involved respectively in line search and trust region techniques, are then always dimensionless and thus easier to interpret. The weights and the corresponding weighted gradient are given in Fig. 1a and 1c respectively. As expected, the weighted inner product compensates for the geometrical spreading and restores balance between shallow and deep contributions. It is however dangerous to use this weight alone because it can be very close to zero in poorly illuminated zones as for example in the corners of the model. In these regions, the weighted inner product is insensitive and consequently the preconditioner is unstable. The simplest stabilization strategy consists in the introduction of a threshold ϵ in the weights

$$\langle m_2, m_1 \rangle_M := \langle m_2 \sqrt{w}, \sqrt{w}\, m_1 \rangle + \epsilon \langle m_2, m_1 \rangle . \tag{2.18}$$

The corresponding preconditioning effect is to keep silent regions where the weight is much smaller than the threshold. Another strategy is to use an inner product with the following stabilization term

$$\langle m_2, m_1 \rangle_M := \langle m_2 \sqrt{w}, \sqrt{w}\, m_1 \rangle + \epsilon\, l_c^2 \langle \boldsymbol{\nabla} m_2, \boldsymbol{\nabla} m_1 \rangle \tag{2.19}$$

where l_c is a characteristic length. This second term, related to spatial derivation, increases the norm of directions that are rapidly varying and also prevents the inner product from being insensitive in regions where the diagonal Hessian is close to zero. In regions where the diagonal Hessian is close to the threshold, then directions with details smaller than the characteristic length l_c are penalized with respect to smoother directions. This inner product is actually very similar to the one introduced in [41], except that the Gauss-Newton diagonal Hessian weight is used in addition. As far as preconditioning is concerned, this inner product can be reformulated through an

integration by parts as

$$\langle m_2, m_1 \rangle_M := \langle w\ m_2, m_1 \rangle + \epsilon\, l_c^2\, \langle \Delta m_2, m_1 \rangle\,. \tag{2.20}$$

Then as previously, conventional and preconditioned gradients are linked

$$\langle j', \delta m_1 \rangle_M = \left\langle j'_{L_2}, \delta m_1 \right\rangle \quad \forall \delta m_1 \tag{2.21}$$

$$\left\langle \left(w + \epsilon\, l_c^2 \Delta\right) j', \delta m_1 \right\rangle = \left\langle j'_{L_2}, \delta m_1 \right\rangle \quad \forall \delta m_1 \qquad \Leftrightarrow \quad j' = \left(w + \epsilon\, l_c^2 \Delta\right)^{-1} j'_{L_2}. \tag{2.22}$$

From the point of view of preconditioning, this inner product generates a rescaling thanks to the Gauss-Newton diagonal Hessian weight and a Laplacian filtering, whose smoothing length equals $2\pi l_c$ where the diagonal Hessian equals the threshold. The effect of these inner products is illustrated in Fig. 1d and 1e. In addition of stabilizing the weights, [41] have shown that a filtering inner product can help the convergence of full waveform inversion by mitigating its non linearity.

In general, any inner product that can be related to the conventional inner product (2.15) through some preconditioner P yields a preconditioned gradient and a preconditioned Hessian operator

$$\langle m_2, m_1 \rangle_M = \langle P m_2, m_1 \rangle \quad \Rightarrow \quad j' = P^{-1} j'_{L_2} \text{ and } H = P^{-1} H_{L_2}. \tag{2.23}$$

Changing the inner product is formally equivalent to preconditioning both the gradient and the Hessian operator. We choose to introduce preconditioning through a change in the inner product rather than through the application of an operator because it appears more elegant and rigorous to us. Moreover, this approach has the pedagogical advantage to include preconditioning inside the inner product choice and thus it does not need to appear explicitly in the description of the optimization algorithms. In terms of practical implementation, it implies that the optimization routines must not be rewritten, only the subroutine which computes the inner product have to be modified, hence providing a lot of flexibility. Basically, a different choice for the inner product does not modify the pure Newton direction because the same preconditioner is applied to both sides of the Newton system (2.3), but does modify the subspace constructed by the conjugate gradient method and does modify norms which are involved in any stopping criterion. A good choice can thus lead to better approximate directions and better truncation rules.

2.3. Steepest descent. The steepest descent is actually the simplest local optimization algorithm. It consists in taking the search direction as the opposite gradient. This is the best direction at first order ($\tilde{H} = 0$) but it can also be seen as a quasi-Newton step where the approximate Hessian operator is the identity operator ($\tilde{H} = I$). In practice however, this approximation is very crude because the Hessian operator is far from the identity operator, even after preconditioning. The downside of this simple method is its linear convergence rate. This slow convergence speed is one of the main motivation for the investigation of higher order algorithms.

2.3.1. Line search globalization. No length information can be captured from the approximate Hessian operator in this case, because it is simply the identity operator ($\tilde{H} = I$). The first trial step length is then chosen based on the history of the outer iterations to save as many step length trials as possible *e.g.* $\gamma = 2(J(m_n) - J(m_{n-1}))/\{D_m J\}(-j')$ [23].

2.3.2. Trust region globalization. Trust-region methods are barely used with steepest descent. Mostly because the linear misfit prediction

$$J^{\text{pred}}(m; p) := J(m) + \langle j'(m), p \rangle_M \tag{2.24}$$

is not accurate enough. Moreover the solution to the trust-region sub-problem (2.6) is trivially $p = -\mu j'$ and is always on the boundary, because of the absence of a second

order term. An upper bound on the relative size of the trust region (μ) is then added to compensate the fact that the trust-region algorithm will never keep it constant. This bound is set to $\mu_{\max} = 4$, 4, 5 for parameter sets A, B, C respectively.

2.4. Limited memory BFGS method. Quasi-Newton methods are expected to provide a huge improvement over the steepest descent and an attractive alternative to Newton methods because they do not involve any expensive Hessian vector product. In place of the exact Hessian, an approximation $\tilde{H} = B$ is used instead. This approximation is built only with the successive gradients and model parameters of each iteration. Moreover, since expensive Hessian vector product are avoided, quasi-Newton methods are sometimes more efficient than Newton methods. The Broyden-Fletcher-Goldfarb-Shanno algorithm, abbreviated BFGS, is maybe the most widely used quasi-Newton method. This method constructs a symmetric and positive definite approximation of the Hessian operator based on all the previous gradients and model parameters. This approximation B_{n+1} is chosen such that it verifies the secant equation

$$B_{n+1}\Delta m_n = \Delta j'_n \quad \text{with } \Delta m_n = m_{n+1} - m_n \text{ and } \Delta j'_n = j'_{n+1} - j'_n \tag{2.25}$$

while being close to the previous approximation B_n and positive definite. Note that imposing the positive definiteness of this approximation also imposes that the update direction must satisfy the (BFGS) curvature condition $\langle \Delta m_n, \Delta j'_n \rangle_M > 0$. One of the biggest advantage of the BFGS algorithm is that it is possible to directly build the approximate inverse Hessian operator B_n^{-1} from the memorized gradients and model parameters. However building explicitly this inverse operator in the context of large-scale optimization is still prohibitively expensive, as well as storing in memory all the previous gradients and model parameters. For these reasons, a limited memory version of the algorithm has been derived. Instead of memorizing all the previous iterates, it only requires the l last iterates and above all, it comes with a two-loop recursive procedure to compute the application of the inverse operator on any direction. The approximate Newton direction associated with the l-BFGS operator is therefore straightforward to compute. This two-loop recursive l-BFGS algorithm is given in Algorithm 2.2 [23].

Algorithm 2.2
Inverse l-BFGS operator application

Require: q, Δm_k, $\Delta j'_k$, $\quad \forall k \in [n-l, n-1]$
  for $k = n-1$ **down to** $k = n-l$ **do**
    $\alpha_k = \langle \Delta m_k, q \rangle_M \,/\, \langle \Delta j'_k, \Delta m_k \rangle_M$
    $q = q - \alpha_k \Delta j'_k$
  end for
  $\xi = \langle \Delta m_{n-1}, \Delta j'_{n-1} \rangle_M \,/\, \langle \Delta j'_{n-1}, \Delta j'_{n-1} \rangle_M$
  $r = \xi\, q$
  for $k = n-l$ **up to** $k = n-1$ **do**
    $\beta_k = \langle \Delta j'_k, r \rangle_M \,/\, \langle \Delta j'_k, \Delta m_k \rangle_M$
    $r = r + (\alpha_k - \beta_k)\Delta m_k$
  end for
  return r $\left(= B_n^{-1} q\right)$

Algorithm 2.3
Direct l-BFGS operator application

Require: q, Δm_k, $\Delta j'_k$, $\quad \forall k \in [n-l, n-1]$
  for $k = n-l$ **up to** $k = n-1$ **do**
    $b_k = \Delta j'_k / \sqrt{\langle \Delta j'_k, \Delta m_k \rangle_M}$
    $a_k = B_n^0 \Delta m_k$
    for $i = n-l$ **up to** $i = k-1$ **do**
      $a_k = a_k + \langle b_i, \Delta m_k \rangle\, b_i - \langle a_i, \Delta m_k \rangle\, a_i$
    end for
    $a_k = a_k / \sqrt{\langle \Delta m_k, a_k \rangle_M}$
  end for
  $r = B_n^0\, q$
  for $k = n-l$ **up to** $k = n-1$ **do**
    $r = r + b_k \langle b_k, q \rangle_M - a_k \langle a_k, q \rangle_M$
  end for
  return r $(= B_n\, q)$

It is important to highlight here that this method also benefits from the modification of the inner product. Indeed the building blocks of this approximate Hessian operator are the successive gradients, which are preconditioned through the inner product. By measuring gradient variations, this method constructs a representation of the misfit which is good enough to produce super-linear convergence, a great improvement over

the steepest descent, at no extra cost. This approximation is however positive definite while the exact Hessian might be indefinite, especially during the early iteration of the inversion. In such cases, this quasi-Newton method may fail to converge while Newton methods may not [20].

2.4.1. Line search globalization. The unit step length $\gamma = 1$ is always tried first because the length information should be captured by the inverse approximate Hessian. Importantly, it can be showed that the (BFGS) curvature condition is always satisfied if the strong Wolfe conditions (2.4) and (2.5) are enforced [23]. Therefore the l-BFGS algorithm combined with a line search will always construct a positive definite approximate Hessian operator B.

2.4.2. Trust region globalization. Finding the exact solution to the trust-region sub-problem (2.6) with the l-BFGS predicted misfit

$$J^{\text{pred}}(m;p) := J(m) + \langle j'(m), p\rangle_M + 0.5\,\langle Bp, p\rangle_M \tag{2.26}$$

is difficult for a general trust region radius. However when this radius is large enough, in particular larger than the unconstrained solution $p^{\text{u}} := -B^{-1}j'$, then it is actually also the exact constrained solution. On the other hand, when the radius is small enough, the quadratic term in the misfit prediction is negligible and the sub-problem is equivalent to the steepest descent, which indicates to follow the gradient up to the boundary. Based on these solutions for extreme value of the radius, the exact solution to the sub-problem (2.6) might be substituted by an interpolation between these two solutions. Namely, the gradient is followed each time the minimum of the misfit prediction along the gradient, *i.e.* the Cauchy point $p^{\text{c}} = -\alpha j'$ (with $\alpha = \langle j', j'\rangle_M / \langle Bj', j'\rangle_M$), is outside the radius. Then for intermediate radii, which contains this Cauchy point but not the unconstrained solution, an interpolation between both is done

$$p(\Delta) = p^{\text{c}} + \tau^* \left(p^{\text{u}} - p^{\text{c}}\right) \text{ with } 0 < \tau^* < 1 \text{ such that } \|p\|_M = \Delta. \tag{2.27}$$

Finally for large radii, the unconstrained solution is accepted. In summary

$$p(\Delta) = \begin{cases} p^{\text{u}} & \text{when } \|p^{\text{u}}\|_M \leq \Delta, \\ -\mu j' & \text{when } \|p^{\text{c}}\|_M \geq \Delta, \\ p^{\text{c}} + \tau^*(p^{\text{u}} - p^{\text{c}}) & \text{when } \|p^{\text{c}}\|_M \leq \Delta \leq \|p^{\text{u}}\|_M . \end{cases} \tag{2.28}$$

The approximate solution (2.28) to the trust-region sub-problem (2.6) is called the dogleg method [23].

A huge difference with the line search implementation of the l-BFGS algorithm is that now the direct application of the approximate Hessian operator B on some directions must be computed. Unfortunately there is no equivalent to Algorithm 2.2 for the direct l-BFGS operator and its application must then be computed from its recursive definition

$$B_n q = B_n^0 q + \sum_{k=n-l}^{n-1} b_k \langle b_k, q\rangle_M - a_k \langle a_k, q\rangle_M \tag{2.29}$$

$$\text{with} \quad a_k = \frac{B_k \Delta m_k}{\sqrt{\langle B_k \Delta m_k, \Delta m_k\rangle_M}} \quad \text{and} \quad b_k = \frac{\Delta j'_k}{\sqrt{\langle \Delta j'_k, \Delta s_k\rangle_M}}. \tag{2.30}$$

It is important to highlight that the sequence of directions a_k could not be memorized because at each iterations the oldest information is discarded, which modifies the whole a_k sequence. A complete procedure to compute the application of the direct

l-BFGS operator is given in Algorithm 2.3. Faster but more sophisticated procedure do exist [23]. However manipulations in the model parameter space are computationally negligible with respect to wave propagation problems hence the speedup would also be negligible. Thanks to this procedure the prospective and retrospective predicted decrease (2.10) and (2.12) can be evaluated. Interestingly, the prospective decrease is evaluated with the current Hessian approximation B_n while the retrospective decrease is evaluated with the next Hessian approximation B_{n+1}. The retrospective ratio is therefore expected to be more often close to one because this next Hessian approximation B_{n+1} is specifically constructed from the update direction $p_n = \Delta m_n = m_{n+1} - m_n$.

2.5. Newton methods. In contrast with quasi-Newton methods, Newton methods use the Hessian operator explicitly, as they try to solve the Newton system (2.3). The interest of these method lies in their independence on the convergence history and in their quadratic convergence rate in the vicinity of the minimum. Far from this minimum, the Hessian operator might however be indefinite, which complicates the solution procedure for the Newton system. For that reason, it is frequent to make the Gauss-Newton approximation ($\tilde{H} = H_{\mathrm{GN}}$), which consist in keeping only the positive definite part of the Hessian operator. The downside of this approximation is then that the second order representation (2.2) of the misfit is less accurate, especially if the negative definite part of the Hessian is not negligible, which might prevent the method from converging. In this section, we present inexact Newton methods based on a line search procedure or a trust region method. Both are valid for the full Hessian and for its Gauss-Newton approximation.

Algorithm 2.4 Conventional conjugate gradient algorithm

$p_0 = 0$, $r_0 = j'$, $q_0 = -j'$
if $\langle Hj', j' \rangle_M \leq 0$ **then return** $-j'$
loop
  if $\langle Hq_k, q_k \rangle_M \leq 0$ **then return** p_k
  $\alpha_k = \langle r_k, r_k \rangle_M \,/\, \langle Hq_k, q_k \rangle_M$
  $p_{k+1} = p_k + \alpha q_k$ and $r_{k+1} = r_k + \alpha H q_k$
  if $\|r_{k+1}\|_M < \eta \|j'\|_M$ **then return** p_{k+1}
  $\beta_{k+1} = \|r_{k+1}\|_M^2 \,/\, \|r_k\|_M^2$
  $q_{k+1} = -r_{k+1} + \beta q_k$
end loop

Algorithm 2.5 Steihaug conjugate gradient algorithm

$p_0 = 0$, $r_0 = j'$, $q_0 = -j'$
loop
  if $\langle Hq_k, q_k \rangle_M \leq 0$ **then**
    $\tau^* = \tau > 0 \mid \|p_k + \tau q_k\|_M = \Delta$
    return $p_k + \tau^* q_k$
  end if
  $\alpha_k = \langle r_k, r_k \rangle_M \,/\, \langle Hq_k, q_k \rangle_M$
  if $\|p_k + \alpha_k q_k\|_M \geq \Delta$ **then**
    $\tau^* = \tau > 0 \mid \|p_k + \tau q_k\|_M = \Delta$
    return $p_k + \tau^* q_k$
  end if
  $p_{k+1} = p_k + \alpha_k q_k$ and $r_{k+1} = r_k + \alpha_k H q_k$
  if $\|r_{k+1}\|_M < \eta \|j'\|_M$ **then return** p_{k+1}
  $\beta_{k+1} = \|r_{k+1}\|_M^2 \,/\, \|r_k\|_M^2$
  $q_{k+1} = -r_{k+1} + \beta_{k+1} q_k$
end loop

2.5.1. Line search globalization. Newton methods can be combined with a line search procedure. In this case a direction p is first found by solving the Newton system approximately with the conventional conjugate gradient method (Algorithm 2.4) [23]. This algorithm constructs iteratively the solution of a linear system without requiring the explicit expression of the Hessian matrix but only its action in particular directions. The iterative procedure is stopped when the residuals have decreased more than some threshold, called the forcing sequence η, which is typically close to zero

$$(\|r_k\|_M :=) \, \|Hp_k + j'\|_M < \eta \, \|j'\|_M \, (= \eta \, \|r_0\|_M) \, . \tag{2.31}$$

Over-solving is here avoided through this forcing term η, which is not systematically close to zero but which is instead chosen to reflect the accuracy of the second-order expansion. Three possible choices for this sequence have been described and studied by [9]. These three choices were then compared in the context of acoustic imaging in [20], who advise to use the forcing sequence

$$\eta_n = \frac{\|j'(m_n) - j'(m_{n-1}) - \gamma_{n-1} H(m_{n-1}) p_{n-1}\|_M}{\|j'(m_{n-1})\|_M}. \tag{2.32}$$

If the accuracy of the local quadratic approximation is good then this forcing term is close to zero and the Newton system is solved accurately. If not, then iterations are truncated sooner. This forcing sequence plays a similar role than the prospective ratio for trust-region method. It is however based on a (prospective) expansion of the gradient while the prospective ratio is based on an expansion of the misfit. Additional safeguards are also added to prevent this forcing term to decrease too fast or to increase above $\eta_0 = 0.9$. Interestingly, directions of negative curvatures are never investigated, except if it is the gradient. As previously an appropriate length γ is then given to this direction p through a line search. The unit step length $\gamma = 1$ is again tried first because it is the best choice if the misfit were quadratic.

2.5.2. Trust region globalization. When the Newton method is associated with a trust-region technique, the direction is found by minimizing the local quadratic expansion of the misfit

$$J^{\text{pred}}(p) := J(m) + \langle j', p\rangle_M + 0.5 \langle Hp, p\rangle_M \tag{2.33}$$

inside a sphere of radius Δ. The constraint $\|p\|_M \leq \Delta$ limits the size of the direction and aims at preventing over-solving. This trust-region sub-problem can be solved approximately with the Steihaug conjugate gradient method (Algorithm 2.5) [33]. This method actually exploits two properties of the conjugate gradient algorithm: successive approximate solutions always grow in norm ($\|p_k\|_M < \|p_{k+1}\|_M$) while the misfit prediction always decrease ($J^{\text{pred}}(p_k) > J^{\text{pred}}(p_{k+1})$). The underlying idea of the method is then to minimize the second order expansion of the misfit iteratively using the conventional conjugate gradient algorithm until either convergence is achieve, either the boundary is reached. Basically there are only two modifications compared to Algorithm 2.4. First, the inner iterations are cropped to the trust region radius Δ when the unconstrained solution increases beyond it. Second, when a direction of negative curvature is encountered, it is followed up to the boundary of the trust region and the algorithm is stopped. Interestingly these directions were never investigated in the conventional version. The convergence criterion is unchanged but here the forcing term is kept constant ($\eta = 0.5$).

3. Numerical investigations. Numerical studies are performed in the context of subsurface acoustic imaging in the frequency domain [29, 32]. In that particular case, the misfit is conventionally chosen as the least-squares distance between some acoustic pressure measurements $d_{\omega er}$ (at some receiver r, for several excitation sources e and for different frequencies ω) and the corresponding computed acoustic pressures $p_{\omega e}(\boldsymbol{x}_r)$, obtained by solving the Helmholtz equation

$$J(s^2) = 0.5 \sum_{\omega,e,r} \left| p_{\omega e}(\boldsymbol{x}_r; s^2) - d_{\omega er} \right|^2 \quad \text{with} \quad \Delta p + \omega^2 s^2 p = \delta(\boldsymbol{x} - \boldsymbol{x}_e). \tag{3.1}$$

It is here chosen that the subsurface model parameter is the slowness squared distribution s^2 [s^2/km^2] (also called the sloth), as could be guessed from the expression of the Helmholtz operator $A_\omega(s^2) := \Delta + \omega^2 s^2$. The slowness squared s^2 is actually the squared inverse of the velocity v. Several other parametrizations are also possible but

it has been shown that the slowness squared can yield a fast convergence and accurate results [2, 5, 14, 27]. Implementation of any of the above described local optimization algorithms requires an efficient procedure to compute the misfit and the gradient for a given slowness squared distribution s^2 and the action of the Hessian operator for any given slowness squared perturbation δs^2. The well-known adjoint state method has been developed for that specific purpose. It is summarized here below and detailed in [1, 13, 28]. The two terms in gray should be removed under the Gauss-Newton approximation.

1. Find the forward fields $p_{\omega e}$ such that
$$A_\omega(s^2)p_{\omega e} = \delta(\boldsymbol{x}-\boldsymbol{x}_e). \tag{3.2}$$
2. Find the adjoint fields $p^\dagger_{\omega e}$ such that
$$A_\omega(s^2)p^\dagger_{\omega e} = \sum_r (\overline{p}_{\omega e}(\boldsymbol{x}_r) - \overline{d}_{\omega e r})\delta(\boldsymbol{x}-\boldsymbol{x}_r). \tag{3.3}$$
3. Find the preconditioned gradient j' such that
$$Pj' = -\sum_\omega \omega^2 \sum_e p^\dagger_{\omega e}\overline{p}_{\omega e}. \tag{3.4}$$
4. Find the perturbed forward fields $\delta p_{\omega e}$ such that
$$A_\omega(s^2)\delta p_{\omega e} = -\omega^2 \delta s^2 p_{\omega e}. \tag{3.5}$$
5. Find the perturbed adjoint fields $\delta p^\dagger_{\omega e}$ such that
$$A_{\omega,e}(s^2)\delta p^\dagger_{\omega e} = \sum_r \delta p_{\omega e}(\boldsymbol{x}_r)\delta(\boldsymbol{x}-\boldsymbol{x}_r) - \omega^2 \delta s^2 p^\dagger_{\omega e}. \tag{3.6}$$
6. Find the preconditioned Hessian operator $H\delta s^2$ in the direction δs^2 such that
$$PH\delta s^2 = -\sum_\omega \omega^2 \sum_e (\delta p^\dagger_{\omega e}\overline{p}_{\omega e} + p^\dagger_{\omega e}\delta\overline{p}_{\omega e}). \tag{3.7}$$

Independently of any practical solver for these wave propagation problems, a misfit evaluation only requires to perform step 1 and thus only requires to solve a single wave propagation problem. A gradient evaluation requires steps 1 to 3, thus a single supplementary wave propagation problem must be solved if the misfit has already been computed. Similarly, steps 1 to 6 are necessary for the application of the (Gauss-)Newton Hessian operator in a particular direction, thus again two supplementary wave propagation problems if the gradient has already been computed for the same model parameters.

Consequently the steepest descent and the l-BFGS directions require to solve two wave problems while any Newton-based direction has an initial cost of four wave propagation problems and each supplementary conjugate gradient iteration requires two more wave problems. To the price of the directions must be added the cost of the line search or the trust-region methods. Line search typically accepts a step length if it verifies sufficient conditions (2.4) and (2.5) which involves the misfit and its gradient. Thus it requires one or two additional wave problems each time a trial step length is rejected. Prospective trust-region has no additional cost because the evaluation of the trust region only depends on quantities already computed. At the opposite, retrospective (Gauss-)Newton trust-region requires the application of the Hessian operator on the preceding direction and thus needs to solve two additional wave propagation problems. The table here below summarizes this accounting.

	Base	CG	LS	TR
SD	2	-	$2N_{\text{LS}}$	-
l-BFGS	2	-	$2N_{\text{LS}}$	-
LS-NCG	2	-	$2N_{\text{LS}}$	-
TR-NCG (P)	2	$2N_{\text{CG}}$	-	0
TR-NCG (R)	2	$2N_{\text{CG}}$	-	2

It is interesting to highlight here that the first inner iteration of any conjugate gradient Newton method is simply the steepest descent but it is twice more expensive because the curvature is computed. Subsequent inner iterations must therefore provide large decrease of the misfit to compensate this high entry cost. This phenomenon is even worse with the retrospective trust region algorithm because there is a systematical additional cost to update the trust region radius.

In this work, solutions to partial differential equations (3.2) to (3.7) are obtained numerically with the finite element method. In what follows, we specify the exact numerical procedure in that context. Note however that the analysis would nearly be identical with finite differences. Finite element discretization assembles operators into matrices and source terms into vectors. Wave propagation problems (3.2), (3.3), (3.5) and (3.6) therefore transform into a linear system whose left-hand-side matrix A is always the same for a given frequency while the right-hand-side source b is different for any field type, frequency and excitation index. The solution of this system is obtained by first computing its lower-upper factorization then by performing an upward-backward substitution for each right-hand-side source

$$\mathrm{Ap} = \mathrm{b} \quad \Leftrightarrow \quad \mathrm{A} = \mathrm{LU}, \quad \mathrm{Lq} = \mathrm{b} \text{ and } \mathrm{Up} = \mathrm{q}. \tag{3.8}$$

Huge computational reduction is therefore obtained because only one matrix per frequency is assembled and factorized. The computation of any wave field then requires the assembly and the upward-backward substitution of a vector per excitation source, but no more matrix factorization. The numerical equivalence of the preceding six steps procedure is given here below.

1.	• Factorize wave propagation operators	(n_ω)
	• Substitute forward sources	$(n_\omega \times n_e)$
2.	• Substitute adjoint sources	$(n_\omega \times n_e)$
3.	• Factorize the preconditioner	(1)
	• Substitute the conventional gradient	(1)
4.	• Substitute perturbed forward sources	$(n_\omega \times n_e)$
5.	• Substitute perturbed adjoint sources	$(n_\omega \times n_e)$
6.	• Substitute the conventional Hessian	(1)

It is interesting to highlight that model problems (steps 3 and 6) are negligible with respect to wave problems. Indeed while wave problems involve a matrix per frequency and a vector per excitation source, model problems only involve a single matrix (*i.e* the preconditioner) and a single source vector (*i.e* the conventional gradient or Hessian). Moreover the model discretization is usually coarser than the wave field discretization. Consequently not considering these model problems when quantifying the computational complexity is not dramatic. It should however be highlighted that forward problems are more expensive than the corresponding adjoint problem, because the matrix factorization is reused. Moreover the perturbed forward problem and the perturbed adjoint problem are slightly heavier than the adjoint problem, because both their sources are dense, at the opposite of forward and adjoint sources, which are sparse. Nevertheless we weight equally all of these four problems when quantifying the computational complexity.

In the next two sections, two synthetic numerical case studies are investigated. The first one is based on the widely used Marmousi benchmark [34] while the second one, replicated from [20], is inspired from a near-surface imaging of close concrete structures and features important multiple scattering. Multiple scattering is responsible for the indefiniteness of the Hessian operator, which, as mentioned in the previous part, is challenging for optimization algorithms. This second example is thus chosen to emphasize which optimization methods are able to overcome such difficulties. For both case studies, the influence of the inner product choice on the convergence speed and the quality of the inverted model is studied first. Once the inner product is chosen, prospective and retrospective trust-region methods with different parameter sets

are compared and the best option is selected. Advantages and drawbacks of trust-region methods in the context of full waveform inversion are then finally discussed based on a comparison with the corresponding line search methods. In the remainder of this section, data misfit are normalized such that the misfit corresponding to the initial model is one and computational complexity is measured in numbers of forward problems solved, as explained above.

3.1. Case study 1. Numerical inversions are performed on the 2D Marmousi model (Fig. 2a) [34] in the frequency domain. Three frequencies (4, 6 and 8 [Hz]) are inverted simultaneously. The surface acquisition system is composed of 122 equally spaced (72 [m]) excitation sources and 243 equally spaced (36 [m]) receivers. Outer iterations are stopped when satisfying the convergence criterion $J(s^2)/J(s^2_{\text{init}}) < 10^{-3}$. A smoothed version of the exact Marmousi model is used as an initial guess (Fig. 2b). This initial model is computed with a Laplacian filter $s^2_{\text{init}} = (1 + (l_c/2\pi)^2\Delta)^{-1} s^2_{\text{exact}}$ with $l_c = 2$ [km]. Slowness squared fields and pressure fields at the three frequencies are discretized on a square grid (36 [m]) by hierarchical finite elements, respectively of order 1 and of order 2, 3, 4. A water layer (216 [m]) is also added at the top of the model but it is kept constant during the inversion. The model is spatially truncated by Sommerfeld boundary conditions [31]. Recorded data are generated synthetically using the same hierarchical finite elements setting than for the inversion. An inversion result, *i.e.* an estimated squared slowness, is shown in (Fig. 2c). From a relatively low resolution initial guess, full waveform inversion indeed provides a high-resolution estimation of the exact model. Images obtained with the other methods do not differ significantly.

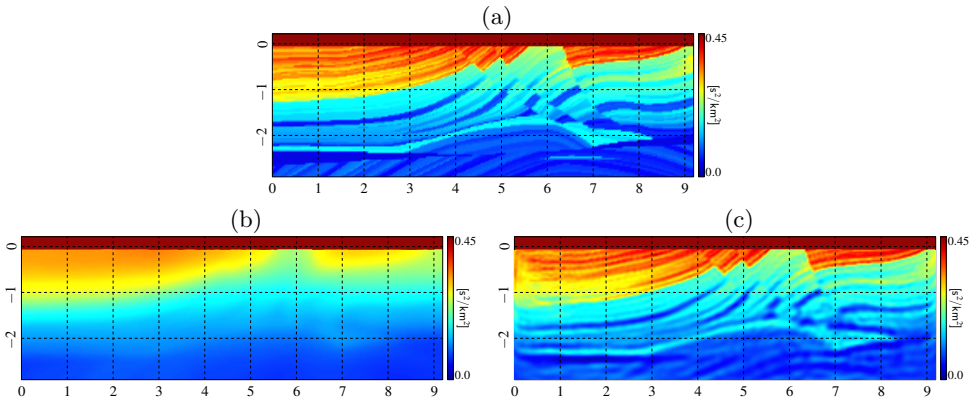

Fig. 2: Marmousi model (a), initial guess (b) and inversion results using a line search l-BFGS algorithm with a weighted and thresholded inner product (c).

3.1.1. Inner product. As explained earlier, the inner product has an influence on both the gradient and the Hessian. Its choice is therefore expected to influence the convergence speed but also the particular minimizer that is reached. To illustrate both these effects, the line search l-BFGS algorithm has been applied with the four different inner products introduced in this work, *i.e.* the conventional inner product (2.15), the weighted inner product (2.16) and its regularized variants (2.18) or (2.19). Corresponding convergence curves and error maps are given in Fig. 3 and 4 respectively. Both these figures are also summarized in Table 1. As can be seen from these figures, all weighted inner product increase the convergence speed with respect to the conventional, *i.e.* unweighted, one. However the minimizer obtained with the weighted inner product alone is further away from the exact solution, in particular in the right corner of the model. Avoiding such artifacts is precisely one

of the reasons for the introduction of regularized inner products, as they dampen the contributions in these poorly illuminated regions. Both the thresholding and the smoothing strategy perform similarly in reducing the error back to the same level than the unweighted solution but the thresholding strategy converges faster. It is thus kept for the sequel of this case study. The advantages of the smoothing inner product will be highlighted during the second case study. In the next three subsections, the behaviour of the steepest descent method, the l-BFGS method, the full Newton and the Gauss-Newton methods is analysed. Convergence curves and interesting statistics for all these methods are given in Fig. 5 and Table 2 respectively.

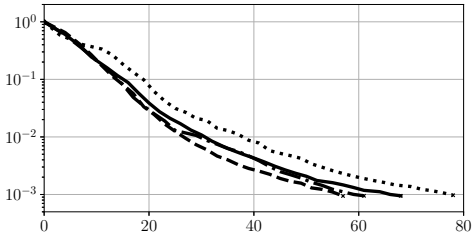

Fig. 3: Data misfit as a function of the computational complexity for the line search l-BFGS algorithm with a conventional (••), only weighted (-•), weighted and stabilized (--) or weighted and smoothed (−) inner product.

		Wave sol. (tot)	Error rms ([s^2/km^2])
Conventional		78	0.0174
Weighted	only	61	0.0202
	and stabilized	57	0.0174
	and smoothed	68	0.0173

Table 1: Computational complexity and root-mean squared error for the line search l-BFGS algorithm with different inner products.

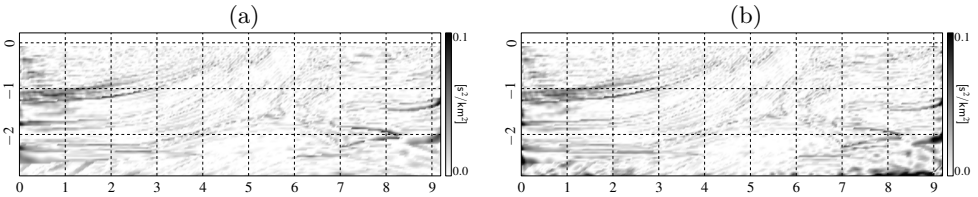

Fig. 4: Final inversion error for the line search l-BFGS algorithm with a conventional (a) or a weighted (b) inner product. Inversion errors for both regularized inner products are not shown because these are very similar to those obtained with the conventional inner product.

3.1.2. Steepest descent. There is no dramatic improvement when using one or another direction scaling method, because actually the direction itself is bad. Nevertheless, it appears that methods which reject less frequently the proposed update direction are faster, *i.e* the prospective trust-region method with the more cautious parameters sets (B and C) and the line search method. Retrospective radius update does not speed up convergence. Actually we observed that the retrospective predicted decrease (2.12) sometimes largely underestimates the actual decrease, illustrating that the retrospective misfit prediction is very not accurate, but still producing an increase of the trust region radius. Finally, among the three best methods, the slope is slightly steeper for the two trust-region methods, probably because they systematically try to increase the length given to the gradient direction.

		Wave sol. (tot)	Outer it. (tot)	Inner it. (avg)	Rejected (%)	Constrained (%)	Negative curv. (%)
SD	LS	244	111	10	-	-	-
	TR-P (A)	396	198	-	40	100	-
	TR-P (B)	280	140	-	6	100	-
	TR-P (C)	264	132	-	5	100	-
	TR-R (A)	328	164	-	20	100	-
	TR-R (B)	354	177	-	20	100	-
	TR-R (C)	330	165	-	25	100	-
LB	LS	57	27	-	7	-	-
	TR-P (A)	58	29	-	3	10	-
	TR-P (B)	58	29	-	3	34	-
	TR-P (C)	64	32	-	13	50	-
	TR-R (A)	58	29	-	3	10	-
	TR-R (B)	56	28	-	0	11	-
	TR-R (C)	56	28	-	0	11	-
FN	LS	139	17	2.9	12	-	29
	TR-P (A)	178	22	3.0	32	77	0
	TR-P (B)	106	13	3.1	0	69	0
	TR-P (C)	106	16	2.3	0	75	0
	TR-R (A)	144	14	3.1	14	64	0
	TR-R (B)	128	14	2.6	0	79	0
	TR-R (C)	142	17	2.2	0	82	0
GN	LS	124	15	3.13	0	-	-
	TR-P (A)	130	11	4.9	0	10	-
	TR-P (B)	98	10	3.9	0	30	-
	TR-P (C)	98	10	3.9	0	30	-
	TR-R (A)	152	11	4.9	0	10	-
	TR-R (B)	132	14	2.7	0	79	-
	TR-R (C)	184	24	1.8	0	83	-

Table 2: Statistics related to the implementation of the steepest descent (SD), the l-BFGS (LB), the full Newton (FN) method and the Gauss-Newton (GN) methods combined with a line search (LS) or a trust region (TR) with a prospective (P) or retrospective (R) radius update with different parameter sets (A,B,C).

3.1.3. Limited memory BFGS method. There is hardly no difference between all the methods combined with the l-BFGS algorithm. We observed that the line search method only rejects the unit step length $\gamma = 1$ for the first two iterations. Similarly, we observed that the retrospective ratio is always very close to one, such that the trust region radius for retrospective methods quickly becomes large and thus the pure l-BFGS direction is always accepted after the first few iterations. An algorithm that unconditionally follows the pure l-BFGS direction would therefore already be very good and neither a line search nor a trust-region method can actually drastically improve it, as far as convergence speed is concerned. Nevertheless the more cautious prospective trust-region methods (B,C) also converge fast, which shows that, on the other hand, constraining the size of the update directions does not slow down the inversion.

3.1.4. Newton methods. As far as trust-region methods are concerned, it first clearly appears that the retrospective radius update is not worth its computational cost. Indeed it does not require less wave solutions than the best prospective ones, even if the computation cost of the retrospective predicted decrease is withdrawn (two wave solutions per outer iterations). Retrospective radius update has been introduced

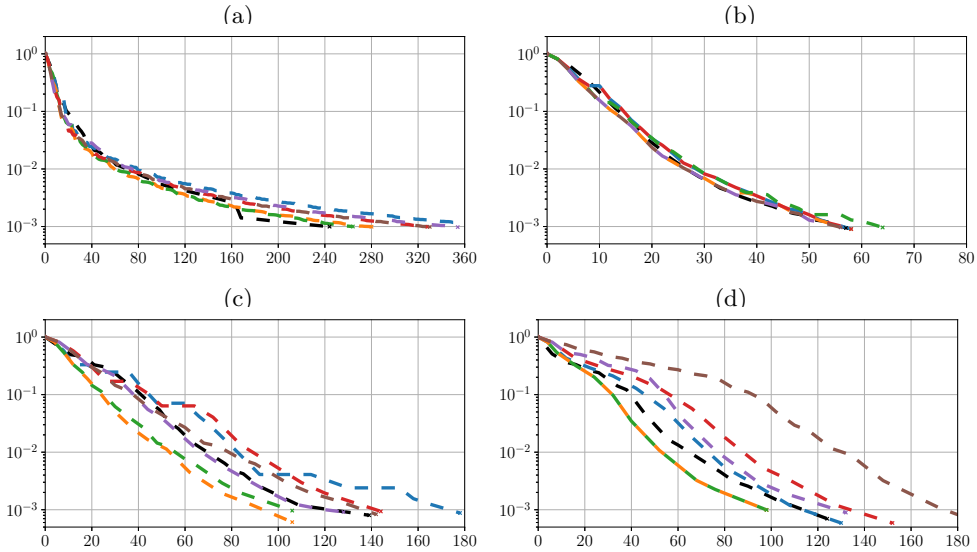

Fig. 5: Data misfit as a function of the computational complexity for the steepest descent (a), the l-BFGS (b), the full Newton (c) and the Gauss-Newton (d) methods combined with either a line search (▬) or a prospective trust region (A (▬), B (▬), C (▬)) or a retrospective trust region (A (▬), B (▬), C (▬)).

to anticipate and prevent failures. However the prospective Newton method combined with the more cautious parameters sets (B and C) does already not reject any direction and there is then no interest in computing the retrospective ratio.

Among the prospective methods, it appears that the two more cautious (B and C) yield the fastest convergence. Convergence speed decreases when using parameter set A with both the full Newton method and the Gauss-Newton method but for two different reasons. With parameter set A, the trust-region radius grows quickly and the full Newton method is thus allowed to explore large directions, beyond the validity of the exact second order expansion (2.2). Such directions produce a high rejection rate (32%) and thus a waste of computational effort. At the opposite, the Gauss-Newton method never rejects a direction and the explanation for its slower convergence can therefore not be the same. During the earliest iterations, far from the global minimum, the Gauss-Newton approximation is not valid (because data residuals are not small yet) and thus the Gauss-Newton Hessian is quiet different from the full Hessian. The misfit prediction under the Gauss-Newton approximation is thus cruder than the exact second order expansion (2.2) and the ratio ρ_{p} is even more likely to be away from one. This ratio ρ_{p} is given in Fig. 6c. As can be seen, during the first few iterations, this ratio is actually very larger than one, which indicates that the misfit prediction is indeed not accurate. Nevertheless, the trust region radius is still increased and the system is solved accurately while the Hessian and the misfit are not approximated accurately. This effect generates over-solving the system at the earliest iteration and slows down the Gauss-Newton method, as can be seen by comparing the initial slopes between variant A and B/C in Fig. 5d. This effect would be even more dominant if the convergence requirements, *i.e.* the forcing sequence η, was smaller. With the large value $\eta = 0.5$ chosen here, convergence of the conjugate gradient algorithm is attained relatively fast. Actually variant B and C perform better than variant A only because it takes more iterations for the trust region constraint to become inactive. Starting with a larger initial radius would result in the same convergence speed than variant A. Also, it is interesting to highlight that when using the retrospective radius

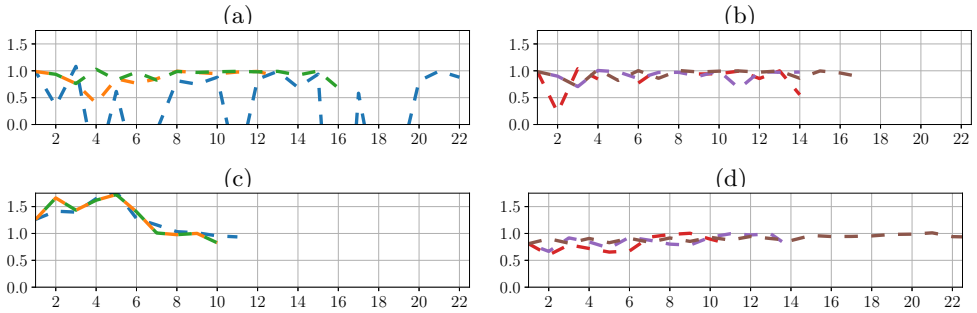

Fig. 6: Prospective ratio ρ_{p} (a,c) or retrospective ratio ρ_{r} (b,d) during the outer iterations of the full Newton method (a,b) and the Gauss-Newton method (c,d) with different parameter sets using a prospective radius update (a,c) (A (▬), B (▬), C (▬)) or a retrospective radius update (b,d) (A (▬), B (▬), C (▬)).

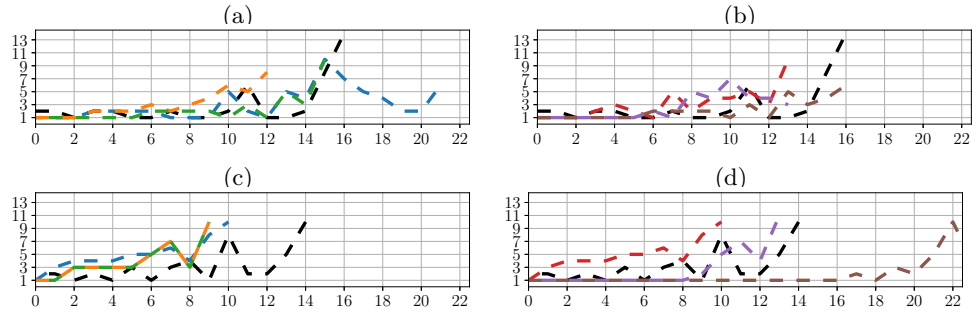

Fig. 7: Inner iterations per outer iteration for the full Newton method (a,b) and the Gauss-Newton method (c,d) with different parameter sets using a prospective radius update (a,c) (A (▬), B (▬), C (▬)) or a retrospective radius update (b,d) (A (▬), B (▬), C (▬)).

update with the Gauss-Newton approximation, the situation is reversed because the retrospective ratio is then smaller than one. Instead of over-solving, under-solving then appears. Therefore we believe that it is better to use trust-region methods with the full Newton Hessian, because it constructs the best possible misfit prediction while it does not introduce supplementary difficulties.

The full Newton method and the Gauss-Newton method are slightly slower when combined with a line search method. As far as the full Newton method is concerned, directions of very small curvature can produce large update directions, far beyond the validity of the expansion (2.2). In such cases the initial length $\gamma = 1$ is rejected and some computational cost must be involved to reduce it to satisfy Wolfe conditions. This effect has actually been observed twice using the full Newton method. Moreover during the first fifth outer iterations, the full Newton method using the line search globalization stops because a direction of negative curvature is encountered. At the opposite of its trust-region counterpart, the line search variant of the conjugate gradient algorithm discard any direction of negative curvature, thus wasting the associated computational cost. Of course within the Gauss-Newton approximation this second effect can not appear (and the first one was actually not observed). The line search globalization therefore seems more suited with the Gauss-Newton approximation. Nevertheless it is not much faster. In the context of line search globalization, the accuracy of the second order local expansion is expressed through the forcing sequence η, which is, as can be seen in Fig. 8, away from zero during the first few iterations. Consequently, the convergence of the conjugate gradient algorithm is quickly

reached and only a few inner iterations are performed per outer iterations as can be seen from Fig. 7c. Fig. 7c actually show how hard it is to stop the Gauss-Newton inner iterations at the right time: the fastest method is the prospective trust region B/C and it performs less inner iterations then the variant A but more than the line search method. The difficulty to pick up an appropriate stopping criterion for the Gauss-Newton method is another motivation to use the full Newton method instead. Using the full Newton method, the line search variant actually suffers from directions of small or negative curvature while trust-region methods do not. Based on this case study, we would therefore recommend to use the full Newton method combined with a trust-region method and a prospective radius update.

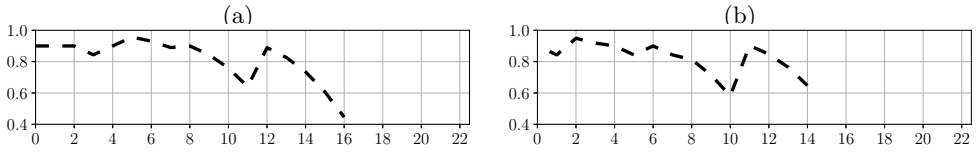

Fig. 8: Forcing sequence η for the full Newton (a) and the Gauss-Newton (b) methods combined with a line search method (▬). The forcing sequence for methods combined with a trust-region is constant ($\eta = 0.5$).

3.2. Case study 2. The configuration of this second case study is replicated from [20]. The true velocity distribution is given in Fig. 9a. It presents two T-shaped concrete structures ($v_c = 4$ [km/s]) embedded in a homogeneous background ($v_b = 0.3$ [km/s]) with a horizontal layer reflector in the bottom ($v_r = 0.5$ [km/s]). The depth of investigation is limited to 3 [m] while the width is 30 [m]. The aspect ratio and the propagation scales are thus very different from the Marmousi model. These two concrete foundations, buried at few meters deep, generate high-amplitude reflections because of the very high velocity contrast with the background. Moreover, important multiple scattering appears between the two structures, as they are relatively close to each other. The acquisition system is divided into three segments: one on the surface and the two others inside boreholes on both lateral sides. Sources and receivers are equally spaced (15 [cm]) along these three segments. Note that the surface sources and receivers that would lie inside the two concrete structures are not considered in the modelling, leading to an actual number of sources and receivers totaling 227. Nine frequencies (100, 125, 150, 175, 200, 225, 250, 275, and 300 [Hz]) are inverted simultaneously from an initial model composed of the homogeneous background and the bottom reflector only. For this second case study, a logarithmic slowness squared parametrization is used $m := \ln s^2$. This parametrization has the advantage to be unable to produce negative values of the slowness squared. Inverting the slowness squared actually drives it into negative values, because of the two concrete structures whose slowness squared is really close to zero. Outer iterations are stopped when satisfying the convergence criterion $J(\ln s^2)/J(\ln s^2_{\text{init}}) < 10^{-2}$. Slowness squared fields and pressure fields at the nine frequencies are discretized on a square grid (15 [cm]) by hierarchical finite elements, respectively of order 1 and of order 2, 3, 4. At the light of the first case study, trust-region methods with parameter sets A and C will no longer be considered, as both were systematically outperformed by parameter set B.

3.2.1. Inner product. Illumination of the medium is nearly perfect and consequently, the diagonal part of the Gauss-Newton Hessian that we previously used as a weight can reasonably be approximated by a constant h_{GN}. However the part related to the change of variable is varying spatially $\delta s^2 = \frac{ds^2}{d\ln s^2}\,\delta \ln s^2 = s^2\,\delta \ln s^2$. Hence the weight for the inner product is chosen as $w = h_{\text{GN}}\ s^4$. As previously, the line search

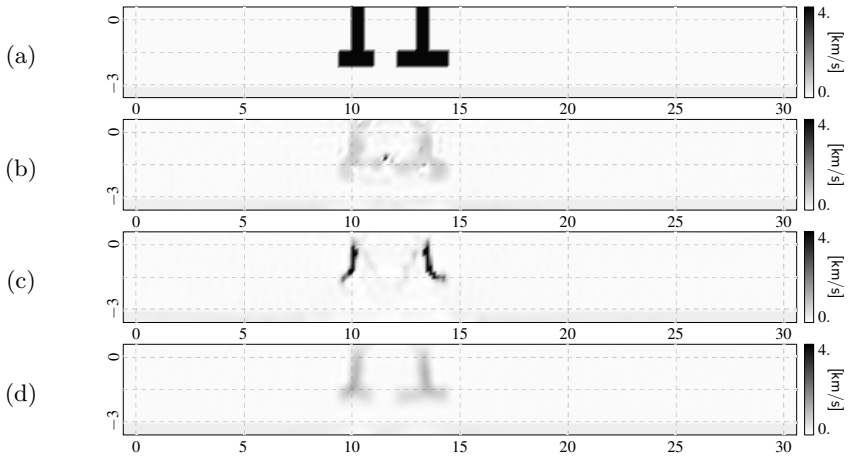

Fig. 9: Near-surface concrete structures velocity model (a) and inversion results using a line search l-BFGS algorithm with a conventional (b), a weighted only (c) or a weighted and smoothed (d) inner product.

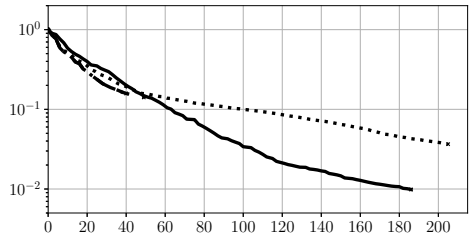

Fig. 10: Data misfit as a function of the computational complexity for the line search l-BFGS algorithm with a conventional (••), only weighted (-•) or weighted and smoothed (−) inner product.

l-BFGS algorithm has been applied with the four different inner products introduced in this work. Convergence curves are given in Fig. 10 while inversion results are given in Fig. 9. For the weighted and smoothed variant, the threshold is set as $\epsilon = h_{\text{GN}}\ s_b^4$ while the characteristic length for the smoothing inner product is set to $l_c = 3$ [m]. It is important to highlight that this length is greater than the smallest wavelength in the background medium (1 [m]) while for the first case study, this length was actually close to the smallest wavelength. The weighted and thresholded variant has been tested for several values of the threshold, from $\epsilon = h_{\text{GN}}\ s_c^4$ to $\epsilon = h_{\text{GN}}\ s_b^4$ but none of them provided inversion results significantly different from the conventional or the weighted inner products. Only the smoothing inner product is able to reconstruct the model parameter accurately. This smoothing inner product actually mitigates the non-linearity of the misfit, because spatial roughness is incorporated progressively in the model parameter [41]. During the inversion, the model parameter never explores extremely high velocity values, at the opposite of the other variants. It is thus able to converge to an accurate solution while more straightforward optimization is not. Consequently, this inner product is used for the remainder of this study. The performance of the three optimization methods is described in the next three subsections. Convergence curves, inversion results and statistics are given in Fig. 12 and 11 and in Table 3 respectively.

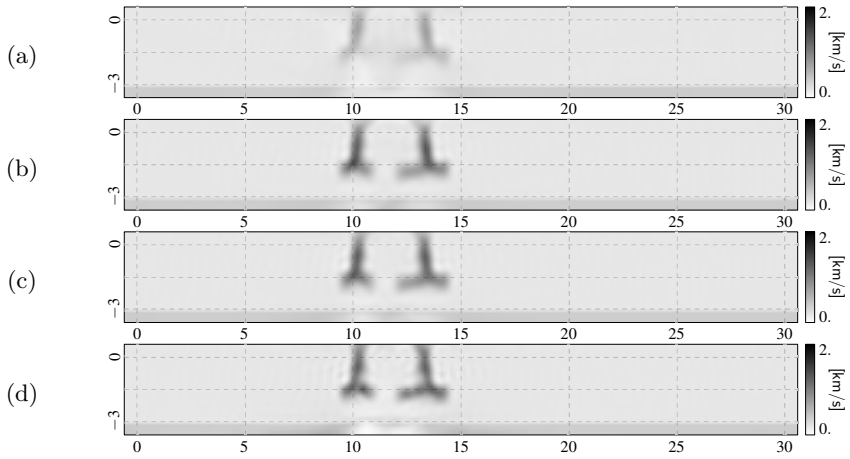

Fig. 11: Inversion results for the steepest descent (a), the l-BFGS (b), the full Newton (c) and the Gauss-Newton (d) methods combined with trust-region method using a prospective radius update (B). Note the the upper color scale limit is only 2 [km/s].

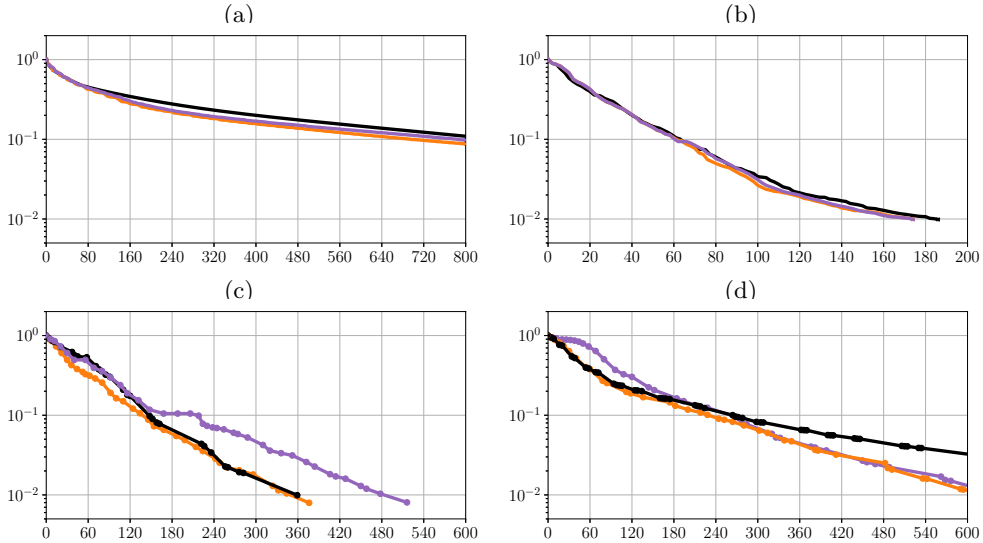

Fig. 12: Data misfit as a function of the computational complexity for the steepest descent (a), the l-BFGS (b), the full Newton (c) and the Gauss-Newton (d) methods combined with either a line search (▬) or a trust-region with a prospective (B (▬)) or a retrospective (B (▬)) radius update. Dots on (Gauss)-Newton curves indicate outer iterations.

3.2.2. Steepest descent. The steepest descent method is not able to reach convergence in a reasonable amount of computations. Progressively decreasing the smoothing length l_c during the inversion would accelerate the convergence [41], but it is not needed for more sophisticated methods and thus it is not done here neither. As for the first test case, the slope of trust-region methods is slightly steeper than the line search method. The prospective radius update rejects less often directions and hence converges faster than the retrospective radius update.

		Wave sol. (tot)	Outer it. (tot)	Inner it. (avg)	Rejected (%)	Constrained (%)	Negative curv. (%)
SD	LS*	803	400	-	1	-	-
	TR-P (B)*	800	400	-	9	100	-
	TR-R (B)*	800	400	-	18	100	-
LB	LS	186	88	-	8	-	-
	TR-P (B)	174	87	-	1	23	-
	TR-R (B)	174	87	-	2	6	-
FN	LS	359	28	5.0	29	-	25
	TR-P (B)	376	33	4.7	0	70	0
	TR-R (B)	516	38	4.8	8	68	0
GN	LS*	923	60	6.6	8	-	-
	TR-P (B)	680	38	7.9	0	42	-
	TR-R (B)	672	39	6.6	0	41	-

Table 3: Statistics related to the implementation of the steepest descent (SD), the l-BFGS (LB), the full Newton (FN) and the Gauss-newton (GN) methods combined with a line search (LS) or a trust-region (TR) with a prospective (P) or retrospective (R) radius update with parameter set B. Star marker $*$ indicates methods that have been stopped before convergence.

3.2.3. Limited memory BFGS method. Similarly to the first test case, the influence of the globalization method on the convergence speed is small. Trust-region methods actually spare a part of the line search cost, but it already represents only a tiny fraction (20 wave solutions) of the overall computational cost (186 wave solutions). Retrospective ratio is again always very close to one and the only difference between retrospective and prospective radius update is the frequency the size constraint is active, although it does not influence the convergence speed.

3.2.4. Newton methods. For this case study, the full Newton method clearly outperforms the Gauss-Newton method, independently of the globalization method used. On the one hand, the convergence speed is much higher and on the other hand the accuracy of the inversion results is superior. As demonstrated in [20], the missing negative definite part of the Hessian can prevent the Gauss-Newton method from reaching convergence. Here, thanks to the inner product preconditioning, every method is able to find the minimum but the invalidity of the Gauss-Newton approximation impacts the convergence speed and the inversion results. Interestingly, for the Gauss-Newton method, the retrospective radius update succeeds to compensate its cost (2 wave solutions per outer iteration). Indeed, during the earliest outer iterations when the Gauss-Newton and the full Hessian are different, we observed that the retrospective ratio is smaller than one while the prospective ratio is bigger than one. Consequently the retrospective method performs less inner iterations per outer iterations than the prospective method (Fig. 13b), and thus avoids early over-solving. In the end both methods still converge at the same speed, but the retrospective method has spent less time in the computation of linear system solutions (680 versus $672 - 2 \times 39 = 594$ wave solutions). At the opposite, for the full Newton method, the retrospective method spent even more time in the computation of linear system solutions than the prospective method. The prospective method is actually already efficient because the prospective misfit prediction is accurate. The line search globalization also provides fast convergence in this case, despite the fact that directions of negative curvature are often encountered (14 wasted wave solutions) and that the

unit step length is often rejected. However the flow of the method is very different from trust-region methods. Indeed line search methods have a tendency to compute a single very accurate system solution, followed by several very inaccurate system solutions as can be seen from Fig. 13a and from the dots spacing in Fig. 12c while trust-region methods perform a nearly steadily increasing number of inner iterations per outer iteration. Whether a flow is better than the other has not been emphasized by our case studies. In the case of noisy data, we however believe it could have an influence.

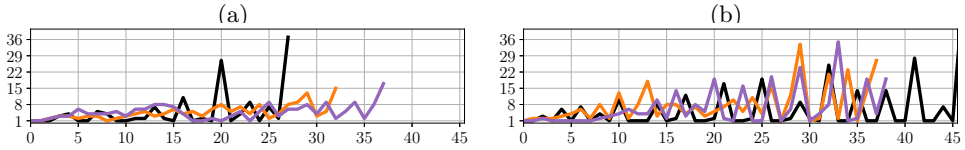

Fig. 13: Inner iterations per outer iteration for the full Newton method (a) and the Gauss-Newton method (b) combined with either a line search (▬) or a trust-region with a prospective (B (▬)) or a retrospective (B (▬)) radius update.

4. Conclusion. In this work, we investigated the use of trust-region methods in the context of full waveform inversion in the frequency domain. At the heart of any trust-region method is the trust-region constraint, which is expressed in terms of the inner product chosen for the model parameter space. Consequently we begun our analysis by investigating different inner product choices that could be implemented. We showed that changing the inner product does not only modify how lengths are measured but also acts as a preconditioner on both the gradient and the Hessian operator. Based on two numerical case studies, we showed that moving from a conventional inner product to a smoothed and/or weighted inner product can accelerate the convergence and mitigate the non-linearity of the misfit, for any optimization method independently of the globalization method (line search or trust region).

In parallel with this inner product choice, we also introduced line search and trust-region variants of the steepest descent, the l-BFGS and the (Gauss-)Newton methods. The number of wave propagation problems to be solved for each method was derived in order to compare them fairly. For each optimization method, the line search and the trust-region globalizations were then compared based on two different case studies. Thanks to the inner product preconditioning, every combination actually already yields very satisfying results. Nevertheless, we showed that trust-region methods outperform line search methods in numerous situations. In particular, we observed that the steepest descent converges slightly faster, because the trust-region methods always tried to increase the step length. As far as the l-BFGS method is concerned, very few differences were noted, but interestingly, constraining the size of the update direction did not decrease the convergence speed. The more dramatic differences appeared when using the full Newton method. Trust-region methods actually overcome the difficulties that appeared when using a line search method with the full Newton method. The Gauss-Newton approximation is not required with trust-region methods and actually degrades their performances, because this approximation also degrades the misfit prediction.

We believe that more sophisticated optimization methods, for example combining l-BFGS and Newton methods, could increase even more the convergence speed. Future works should also investigate the behaviour of inner product preconditioned trust-region methods in the presence of noise, possibly with new inner products that involve prior information on the model parameter space. We believe that the size constraint could act as a regularization method *per se*. Based on our study and these potential extensions, trust-region methods and inner product preconditioning seem to be two very useful tools for full waveform inversion.

5. Acknowledgements. The authors would like to thank Anthony Royer for his help on the finite element solver used in this work [30].

REFERENCES

[1] X. Adriaens, F. Henrotte, and C. Geuzaine, *Adjoint state method for time-harmonic scattering problems with boundary perturbations*, Journal of Computational Physics, 428 (2021), p. 109981.
[2] A. Y. Anagaw and M. D. Sacchi, *Model parametrization strategies for Newton-based acoustic full waveform inversion*, Journal of Applied Geophysics, 157 (2018), pp. 23–36.
[3] R. Brossier, S. Operto, and J. Virieux, *Seismic imaging of complex onshore structures by 2D elastic frequency-domain full-waveform inversion*, GEOPHYSICS, 74 (2009), pp. WCC105–WCC118.
[4] R. Brossier, S. Operto, and J. Virieux, *Which data residual norm for robust elastic frequency-domain full waveform inversion?*, GEOPHYSICS, 75 (2010), pp. R37–R46.
[5] M. d. S. R. Carneiro, B. Pereira-Dias, D. M. Soares Filho, and L. Landau, *On the Scaling of the Update Direction for Multi-parameter Full Waveform Inversion: Applications to 2D Acoustic and Elastic Cases*, Pure and Applied Geophysics, 175 (2018), pp. 217–241.
[6] E. Causse, R. Mittet, and B. Ursin, *Preconditioning of full-waveform inversion in viscoacoustic media*, GEOPHYSICS, 64 (1999), pp. 130–145.
[7] B. Consolvo, M. Zuberi, R. Pratt, and P. Cary, *FWI with Scaled-Sobolev Preconditioning Applied to Short-offset Vibroseis Field Data*, in 79th EAGE Conference and Exhibition 2017, no. July, 6 2017, pp. 10–15.
[8] D. Datta and M. K. Sen, *Estimating a starting model for full-waveform inversion using a global optimization method*, GEOPHYSICS, 81 (2016), pp. R211–R223.
[9] S. C. Eisenstat and H. F. Walker, *Choosing the forcing terms in an inexact Newton method*, SIAM Journal of Scientific Computing, 17 (1996), pp. 16–32.
[10] J. Fan, J. Pan, and H. Song, *A Retrospective Trust Region Algorithm with Trust Region Converging to Zero*, Journal of Computational Mathematics, 34 (2016), pp. 421–436.
[11] J.-y. Fan and Y.-x. Yuan, *A new trust region algorithm with trust region radius converging to zero*, Proceedings of the 5th International Conference on Optimization: Techniques and Applications, (2001).
[12] Y. Favennec, F. Dubot, D. Le Hardy, B. Rousseau, and D. R. Rousse, *Space-Dependent Sobolev Gradients as a Regularization for Inverse Radiative Transfer Problems*, Mathematical Problems in Engineering, 2016 (2016), pp. 1–18.
[13] A. Fichtner and J. Trampert, *Hessian kernels of seismic data functionals based upon adjoint techniques*, Geophysical Journal International, 185 (2011), pp. 775–798.
[14] A. Guitton, G. Ayeni, and E. Díaz, *Constrained full-waveform inversion by model reparameterization*, GEOPHYSICS, 77 (2012), pp. R117–R127.
[15] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich, *Optimization with PDE Constraints*, vol. 23 of Mathematical Modelling: Theory and Applications, Springer Netherlands, Dordrecht, 2009.
[16] W. Hu, A. Abubakar, T. M. Habashy, and J. Liu, *Preconditioned non-linear conjugate gradient method for frequency domain full-waveform seismic inversion*, Geophysical Prospecting, 59 (2011), pp. 477–491.
[17] P. Kazemi and R. Renka, *Minimization of the Ginzburg–Landau energy functional by a Sobolev gradient trust-region method*, Applied Mathematics and Computation, 219 (2013), pp. 5936–5942.
[18] P. Lin, S. Peng, Y. Lu, and W. Du, *The trust region method for time-domain full waveform inversion*, in Proceedings of the 7th International Conference on Environment and Engineering Geophysics & Summit Forum of Chinese Academy of Engineering on Engineering Science and Technology, Paris, France, 2016, Atlantis Press, pp. 220–223.
[19] L. Métivier, F. Bretaudeau, R. Brossier, S. Operto, and J. Virieux, *Full waveform inversion and the truncated Newton method: quantitative imaging of complex subsurface structures*, Geophysical Prospecting, 62 (2014), pp. 1353–1375.
[20] L. Métivier, R. Brossier, J. Virieux, and S. Operto, *Full Waveform Inversion and the Truncated Newton Method*, SIAM Journal on Scientific Computing, 35 (2013), pp. B401–B437.
[21] W. Mulder and R.-E. Plessix, *Exploring some issues in acoustic full waveform inversion*, Geophysical Prospecting, 56 (2008), pp. 827–841.
[22] J. Neuberger, *Sobolev Gradients and Differential Equations*, vol. 1670 of Lecture Notes in Mathematics, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
[23] J. Nocedal, S. J. Wright, and S. M. Robinson, *Numerical Optimization*, Springer Series in Operations Research and Financial Engineering, Springer New York, 2006.
[24] S. Operto, Y. Gholami, V. Prieux, A. Ribodetti, R. Brossier, L. Metivier, and J. Virieux, *A guided tour of multiparameter full-waveform inversion with multicompo-*

nent data: From theory to practice, The Leading Edge, 32 (2013), pp. 1040–1054.
[25] W. Pan, K. A. Innanen, and W. Liao, *Accelerating Hessian-free Gauss-Newton full-waveform inversion via l-BFGS preconditioned conjugate-gradient algorithm*, GEOPHYSICS, 82 (2017), pp. R49–R64.
[26] W. Pan, K. A. Innanen, G. F. Margrave, M. C. Fehler, X. Fang, and J. Li, *Estimation of elastic constants for HTI media using Gauss-Newton and full-Newton multiparameter full-waveform inversion*, GEOPHYSICS, 81 (2016), pp. R275–R291.
[27] B. Park, W. Ha, and C. Shin, *A comparison of the preconditioning effects of different parameterization methods for monoparameter full waveform inversions in the Laplace domain*, Journal of Applied Geophysics, 172 (2020), p. 103883.
[28] R.-E. Plessix, *A review of the adjoint-state method for computing the gradient of a functional with geophysical applications*, Geophysical Journal International, 167 (2006), pp. 495–503.
[29] R. G. Pratt and C. Shin, *Gauss-Newton and full Newton methods in frequency-space seismic waveform inversion*, Geophysical Journal International, 133 (1998), pp. 341–362.
[30] A. Royer, E. Béchet, and C. Geuzaine, *GMSH-FEM : an efficient finite element library based on GMSH*, in 14th World Congress on Computational Mechanics (WCCM), no. July, 2020, pp. 19–24.
[31] S. H. Schot, *Eighty years of Sommerfeld's radiation condition*, Historia Mathematica, 19 (1992), pp. 385–401.
[32] L. Sirgue and R. G. Pratt, *Efficient waveform inversion and imaging: A strategy for selecting temporal frequencies*, GEOPHYSICS, 69 (2004), pp. 231–248.
[33] T. Steihaug, *The Conjugate Gradient Method and Trust Regions in Large Scale Optimization*, SIAM Journal on Numerical Analysis, 20 (1983), pp. 626–637.
[34] R. Versteeg, *The Marmousi experience: Velocity model determination on a synthetic complex data set*, The Leading Edge, 13 (1994), pp. 927–936.
[35] J. Virieux and S. Operto, *An overview of full-waveform inversion in exploration geophysics*, GEOPHYSICS, 74 (2009), pp. WCC1–WCC26.
[36] P. Wellington*, R. Brossier, S. Garambois, and J. Virieux, *Laplacian based Preconditioning of FWI: Using prior information from seismic reflection data.*, in SEG Technical Program Expanded Abstracts 2015, Society of Exploration Geophysicists, 8 2015, pp. 1436–1440.
[37] X. Yan, Q. He, and Y. Wang, *Truncated trust region method for nonlinear inverse problems and application in full-waveform inversion*, Journal of Computational and Applied Mathematics, 404 (2022), p. 113896.
[38] P. Yang, R. Brossier, L. Métivier, J. Virieux, and W. Zhou, *A Time-Domain Preconditioned Truncated Newton Approach to Visco-acoustic Multiparameter Full Waveform Inversion*, SIAM Journal on Scientific Computing, 40 (2018), pp. B1101–B1130.
[39] H. Zhang, X. Li, H. Song, and S. Liu, *An adaptive subspace trust-region method for frequency-domain seismic full waveform inversion*, Computers & Geosciences, 78 (2015), pp. 1–14.
[40] W. Zhang and Y. Li, *Elastic wave full-waveform inversion in the time domain by the trust region method*, Journal of Applied Geophysics, 197 (2022), p. 104540.
[41] M. A. Zuberi and R. G. Pratt, *Mitigating nonlinearity in full waveform inversion using scaled-Sobolev pre-conditioning*, Geophysical Journal International, 213 (2018), pp. 706–725.