Computers & Geosciences 37 (2011) 444–455

Contents lists available at ScienceDirect

Computers & Geosciences

journal homepage: www.elsevier.com/locate/cageo

Two-dimensional frequency-domain visco-elastic full waveform inversion: Parallel algorithms, optimization and performance

R. Brossier [1]

Géoazur, Université de Nice-Sophia Antipolis - CNRS, 250 rue Albert Einstein, 06560 Valbonne, France

ARTICLE INFO

Article history:
Received 21 August 2009
Received in revised form
17 September 2010
Accepted 21 September 2010
Available online 10 November 2010

Keywords:
Seismic wave modelling
Finite element discontinuous Galerkin
Multi-parameter seismic imaging
Quasi-Newton optimization
Massively parallel computing

ABSTRACT

Full waveform inversion (FWI) is an appealing seismic data-fitting procedure for the derivation of high-resolution quantitative models of the subsurface at various scales. Full modelling and inversion of visco-elastic waves from multiple seismic sources allow for the recovering of different physical parameters, although they remain computationally challenging tasks. An efficient massively parallel, frequency-domain FWI algorithm is implemented here on large-scale distributed-memory platforms for imaging two-dimensional visco-elastic media. The resolution of the elastodynamic equations, as the forward problem of the inversion, is performed in the frequency domain on unstructured triangular meshes, using a low-order finite element discontinuous Galerkin method. The linear system resulting from discretization of the forward problem is solved with a parallel direct solver. The inverse problem, which is presented as a non-linear local optimization problem, is solved in parallel with a quasi-Newton method, and this allows for reliable estimation of multiple classes of visco-elastic parameters. Two levels of parallelism are implemented in the algorithm, based on message passing interfaces and multi-threading, for optimal use of computational time and the core-memory resources available on modern distributed-memory multi-core computational platforms. The algorithm allows for imaging of realistic targets at various scales, ranging from near-surface geotechnic applications to crustal-scale exploration.

1. Introduction

Quantitative imaging of the subsurface is essential for geotechnic and civil engineering applications, for seismic hazard and natural resources exploration, and for our geodynamics knowledge. The physical properties of the subsurface are often estimated through seismic-wave analysis, with wave travel-times generally used to perform tomography of the Earth interior. In the 1980s, the full waveform inversion (FWI) method was introduced by Tarantola (1984) to exploit the complete information contained in the waveforms of seismograms, and to infer high-resolution models of the subsurface. FWI was originally developed in the time-domain (Tarantola, 1984, 1987), and it has become tractable and popular in the frequency-domain since the pioneering work of Pratt and Worthington (1990) and Pratt (1990). FWI is currently a field of active research, and particularly for active seismic surveys at various scales. Computationally efficient frequency-domain FWI was designed by limiting the inversion to a few discrete frequencies, taking advantage of the redundant wavenumber cover provided by wide-aperture surveys (Sirgue and Pratt, 2004; Brenders and Pratt, 2007a). Frequency-domain FWI potentially provides high-resolution quantitative images of physical parameters, although it suffers from two main difficulties.

The first of these difficulties relates to the computational cost of the forward problem; namely, the numerical resolution of the wave equation in heterogeneous media for multiple sources. In two dimensions, the forward problem can be solved efficiently in the frequency-domain (Marfurt, 1984), leading to the resolution of a linear system per frequency, the right-hand side of which is the source. This system is solved efficiently for multiple sources with a direct solver, which performs one LU factorization of the matrix, followed by a forward and a backward substitution per source. For the acoustic wave equation, optimal finite-difference (FD) schemes have been designed for optimal use of the CPU core-memory when used with direct solvers (Jo et al., 1996; Hustedt et al., 2004). However, these optimal schemes are difficult to design for the elastic wave equation, because the numerical scheme directly depends on the Poisson ratio of the medium, which leads to stability issues when considering free-surface boundary conditions and liquid/solid interfaces (Stekl and Pratt, 1998). Moreover, FD suffers from the usual limitation linked to Cartesian regular grids: the grid interval is defined by the smallest propagated wavelength to avoid numerical dispersion, or by complex interface geometries, which require fine discretization to avoid parasite diffractions from the corners of the Cartesian grid. These two constraints generally

E-mail address: romain.brossier@obs.ujf-grenoble.fr
[1] Now at Laboratoire de Géophysique Interne et Tectonophysique, Université Joseph Fourier, Grenoble, France.

0098-3004/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cageo.2010.09.013

lead to local oversampling of the medium, and therefore a waste of computational resources (Saenger and Bohlen, 2004; Bohlen and Saenger, 2006). To deal with non-regular grids, finite element (FE) methods can be considered with either low interpolation orders (Marfurt, 1984; Bielak et al., 2003), or high interpolation orders, using the so-called spectral element method (Komatitsch and Vilotte, 1998; Chaljub et al., 2003). These methods require, however, explicit boundary conditions for modelling the water/solid contact, and are generally developed for quadrangle grids, which leads to meshing issues. Another alternative is the finite element discontinuous Galerkin (DG) method (Käser and Dumbser, 2006; Dumbser and Käser, 2006; Brossier et al., 2008). This DG method allows for the use of triangular/tetrahedral meshes, and it is suitable for the handling of strong physical contrasts in the medium, including liquid/solid contact. The DG method is chosen in the present study to perform the FWI forward problem in the frequency-domain.

The second difficulty of FWI is related to the ill-posedness and nonlinearity of the inverse problem, which is generally formulated as a least-squares local optimization (Tarantola, 1987; Pratt and Worthington, 1990), so as to manage the numerical cost of the forward problem. The ill-posedness of FWI mainly arises from the lack of low frequencies in the source bandwidth and the incomplete illumination of the subsurface provided by conventional seismic surveys. Consequently, the problem is highly non-linear and the results strongly depend on the accuracy of the starting model in the framework of local optimization and on the presence of noise. Several hierarchical multi-scale strategies that proceed from low frequencies to higher frequencies have been proposed to mitigate the nonlinearity of the inverse problem (Pratt and Worthington, 1990; Bunks et al., 1995; Sirgue and Pratt, 2004; Brossier et al., 2009a). Subsets of data and parameter classes can also be hierarchically inverted to mitigate the nonlinearity and nonuniqueness issues in the framework of elastic inversion (Sears et al., 2008; Brossier et al., 2009b).

To tackle targets of realistic size and complexity with FWI, efficient algorithms must be implemented for modern computational platforms for both the forward and the inverse problems. An example of such algorithms is presented in Sourbier et al. (2009a), who implemented a parallel FWI algorithm for large-scale distributed-memory platforms in the framework of the two-dimensional (2D) acoustic monoparameter approximation, with a FD forward problem and steepest-descent-based optimization scheme.

The present study presents the implementation of a massively parallel frequency-domain FWI algorithm for imaging 2D viscoelastic media. The second section will focus on the forward problem resolution, as performed with a low-order DG method. In the third section, the inversion algorithm problem is presented, with special emphasis on the reconstruction of multi-parameter classes from vectorial wavefields using a quasi-Newton method. The last section will review the parallel implementation of the algorithm, with an analysis of its performance on a distributed-memory platform.

2. Forward problem

2.1. Theory and discretization

Two-dimensional, elastic, frequency-domain FWI requires the frequency-domain solutions of the 2D P-SV wave equations for heterogeneous media and multiple sources. We consider the first-order hyperbolic system where both particle velocities (V_x,V_z) and stresses $(\sigma_{xx},\sigma_{zz},\sigma_{xz})$ are unknown quantities, as described by the system:

$$
\begin{aligned}
-i\omega V_x &= \frac{1}{\rho(\mathbf{x})}\left\{\frac{\partial \sigma_{xx}}{\partial x}+\frac{\partial \sigma_{xz}}{\partial z}\right\}+f_x\\
-i\omega V_z &= \frac{1}{\rho(\mathbf{x})}\left\{\frac{\partial \sigma_{xz}}{\partial x}+\frac{\partial \sigma_{zz}}{\partial z}\right\}+f_z\\
-i\omega \sigma_{xx} &= (\lambda(\mathbf{x})+2\mu(\mathbf{x}))\frac{\partial V_x}{\partial x}+\lambda(\mathbf{x})\frac{\partial V_z}{\partial z}-i\omega\sigma_{xx_0}\\
-i\omega \sigma_{zz} &= \lambda(\mathbf{x})\frac{\partial V_x}{\partial x}+(\lambda(\mathbf{x})+2\mu(\mathbf{x}))\frac{\partial V_z}{\partial z}-i\omega\sigma_{zz_0}\\
-i\omega \sigma_{xz} &= \mu(\mathbf{x})\left\{\frac{\partial V_x}{\partial z}+\frac{\partial V_z}{\partial x}\right\}-i\omega\sigma_{xz_0},
\end{aligned}
\tag{1}
$$

where λ and μ are the Lamé coefficients, ρ is the density, and ω is the angular frequency. Source terms are either point forces (f_x, f_z) or applied stresses $(\sigma_{xx_0},\sigma_{zz_0},\sigma_{xz_0})$. The symbol i denotes the purely imaginary term defined by $i=\sqrt{-1}$. Only isotropic media are considered in this analysis. The intrinsic attenuation of the medium is easily taken into account in the frequency-domain using complex-valued wave velocities. In the present study, the Kolsky–Futterman model (Kolsky, 1956; Futterman, 1962) without the dispersion term is used (Toksöz and Johnston, 1981). Please note, however, that the dispersion term could be included without extra cost in the formulation.

A change in variable is applied to the system of Eqs. (1) to develop a pseudo-conservative formulation that is useful for the derivation of the DG formulation. We now consider the following vector with three components $(T_1,T_2,T_3)=((\sigma_{xx}+\sigma_{zz})/2,(\sigma_{xx}-\sigma_{zz})/2,\ \sigma_{xz})$. Note that, when considering this variable change, T_1 represents the hydrostatic pressure measured in liquids. In the case of liquid/solid propagation, such as in marine seismic experiments, the T_1 variable provides direct access to the pressure measured by hydrophones, while T_2 and T_3 are zero in liquid, relative to the deviatoric part of the stress tensor.

Moreover, we must consider a finite domain, and therefore, we apply perfectly matched layer (PML) absorbing conditions (Berenger, 1994) through the functions s_x, s_z, defined in the direction r as $s_r=1/(1+i\gamma_r/\omega)$, where γ_r is zero inside the computational domain and nonzero inside the absorbing layers. The new differential system that includes the PML functions, which is equivalent to the system of Eqs. (1), can be written as

$$
\begin{aligned}
-i\omega\rho V_x &= s_x\frac{\partial(T_1+T_2)}{\partial x}+s_z\frac{\partial T_3}{\partial z}+\rho f_x\\
-i\omega\rho V_z &= s_x\frac{\partial T_3}{\partial x}+s_z\frac{\partial(T_1-T_2)}{\partial z}+\rho f_z\\
\frac{-i\omega T_1}{\lambda+\mu} &= s_x\frac{\partial V_x}{\partial x}+s_z\frac{\partial V_z}{\partial z}-\frac{i\omega T_1^0}{\lambda+\mu}\\
\frac{-i\omega T_2}{\mu} &= s_x\frac{\partial V_x}{\partial x}-s_z\frac{\partial V_z}{\partial z}-\frac{i\omega T_2^0}{\mu}\\
\frac{-i\omega T_3}{\mu} &= s_x\frac{\partial V_z}{\partial x}+s_z\frac{\partial V_x}{\partial z}-\frac{i\omega T_3^0}{\mu}.
\end{aligned}
\tag{2}
$$

To derive the DG formulation, the model is first discretized with polygonal cells. For each cell, the system of Eqs. (2) is multiplied by a test function, corresponding to a k th-order polynomial. The test function is nonzero only in the polygonal cell that ensures the discontinuous property of the scheme. In the present study, the nodal formulation of DG method is used (Hesthaven and Warburton, 2008), based on Lagrange polynomials that can be easily defined and integrated on triangle shapes. The system of Eqs. (2) is then integrated over each cell, which leads to the so-called weak formulation of the system. We assume that the physical properties of the medium are constant inside each cell, and we use centred numerical fluxes to exchange energy between cells (Remaki, 2000). The reader is referred to Käser and Dumbser (2006), Dumbser and Käser (2006), Brossier et al. (2008),

Etienne et al. (2010) for a detailed description of the DG formulation applied to the elastodynamic equations.

Discretization of Eqs. (2) with the DG method leads to a linear system for each frequency considered:

$$\mathbf{Au} = \mathbf{s}, \tag{3}$$

where **A** is the sparse impedance matrix, which is also known as the forward problem operator, and which depends on the frequency, the mesh geometry, the DG interpolation order of each cell, and the physical properties. The right-hand side **s** represents the external excitation. The unknown vector **u** contains the particle velocity and the stress wavefields in the entire computational domain, and is represented by a local polynomial inside each cell. In the present study, the system of Eqs. (3) is solved with the massively parallel direct solver MUMPS (Amestoy et al., 2006). More details on this algorithm and its performance will be discussed in Section 4.1.

Note that visco-acoustic modelling and inversion can be performed using the elastic algorithm: the fourth and fifth lines of system 2 are not considered in the forward problem; the T_2 and T_3 wavefields are forced to zero; and V_S is set to zero.

2.2. Which interpolation?

The theoretical resolution of the FWI is $\lambda/2$ if normal-incidence data are recorded, where λ is the local propagated wavelength (Sirgue and Pratt, 2004). A suitable discretization for the inversion mesh should therefore be close to $\lambda/4$ to ensure a correct sampling of the medium. Since the physical properties are assumed to be constant inside each cell, a suitable discretization of the forward problem based on the same mesh as that of the inversion should also be close to $\lambda/4$ (Sourbier et al., 2009a).

DG methods generally take advantage of high orders of interpolation to coarsely discretize the medium and to obtain a high order of accuracy (Käser and Dumbser, 2006; Dumbser and Käser, 2006). In contrast, I will focus here on low interpolation orders for FWI applications because these low orders of interpolation will provide the best compromise between accuracy and computational cost of the forward problem for the discretization imposed by the FWI resolution.

The lowest interpolation order, P_0 (Brossier et al., 2008), is a possible choice that represents the wavefield by a series of piecewise constant functions inside each cell, and it gives sufficiently accurate solutions for 10–15 cells per wavelength in structured equilateral triangular meshes. P_0 interpolation cannot, however, be used on unstructured meshes, where it does not converge to the correct solution, whatever the size of the cell. Moreover, the liquid/solid interface causes some numerical instabilities for structured meshes, and the P_0 interpolation cannot, therefore, be used in this case. Finally, the P_0 scheme can be related to the finite volume P_0 scheme, and is very close to $O(\Delta x^2)$ FD schemes.

The higher orders of interpolation, P_1 and P_2, allow for the use of unstructured meshes, which is an appealing feature for dealing with complex topographies and heterogeneous media, which should require a local adaptation of the mesh size to the local properties (h-adaptive meshes). Sufficiently accurate solutions can be obtained in unstructured meshes with discretization of 10–15 cells per wavelength for P_1, and three to three and a half cells per wavelength for P_2. Note that for inversion, P_2 should be oversampled with four cells per wavelength to correctly sample the medium and mitigate errors in the gradient building (see Section 3.3).

Seismograms computed with the DG P_1 and DG P_2 in a homogeneous halfspace are compared with the analytical solution of the Garvin (1956) problem (Figs. 1 and 2). The source is an explosion at depth and the receivers are on the free surface. The comparison is performed in the time-domain, from DG solutions computed in the frequency-domain. To assess the accuracy of the surface wave and of the direct wave, I have here used a discretization of 12 cells and three cells per minimum wavelength in unstructured triangular meshes for P_1 and P_2 interpolations, respectively. The comparison between the DG and analytical seismograms shows that sufficiently accurate solutions for both wave types are achieved for these discretizations. For the Garvin test computed in the frequency-domain at 5 Hz, Fig. 3 reviews the accuracy of the three interpolation orders as functions of the mesh size.

The P_2 interpolation provides the best consistency between the discretizations required by the forward problem to compute sufficiently accurate solutions (Figs. 2 and 3) and the spatial resolution of FWI (i.e., the mesh interval corresponding to a quarter of a wavelength), considering my formulation based on constant properties per cell. Higher orders of interpolation would allow one to use coarser meshes for the same level of accuracy, but these coarse meshes would lead to an undersampling of the medium for FWI. Therefore, high orders of interpolation should be applied on unnecessarily fine meshes, at the expense of the computational efficiency, and should, therefore, be rejected for FWI applications. As I will illustrate later, lower orders of interpolation (i.e., P_0 and P_1) can, however, be useful as an alternative to P_2, depending on the medium properties. Please note that, if heterogeneous properties were considered in cells, higher orders of interpolation could lead to more computationally-efficient modelling engines for FWI. This approach has not been investigated so far.

For computational efficiency, an interesting feature of the DG method is the possibility of mixing the interpolation orders (p-adaptivity), because the degrees of freedom are not shared by

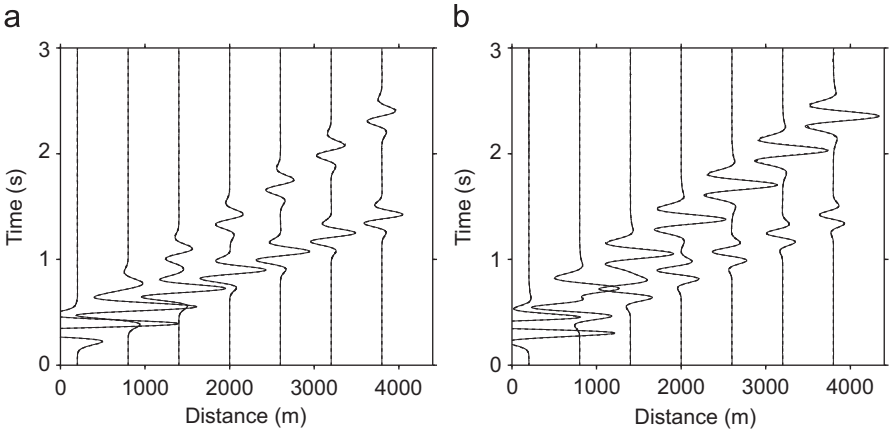

Fig. 1. Time-domain comparison between DG P_1 (solid lines) and analytical (dashed lines) solutions for the Garvin test. The seismograms represent: (a) the horizontal and (b) the vertical components of the particle velocity at the free surface.

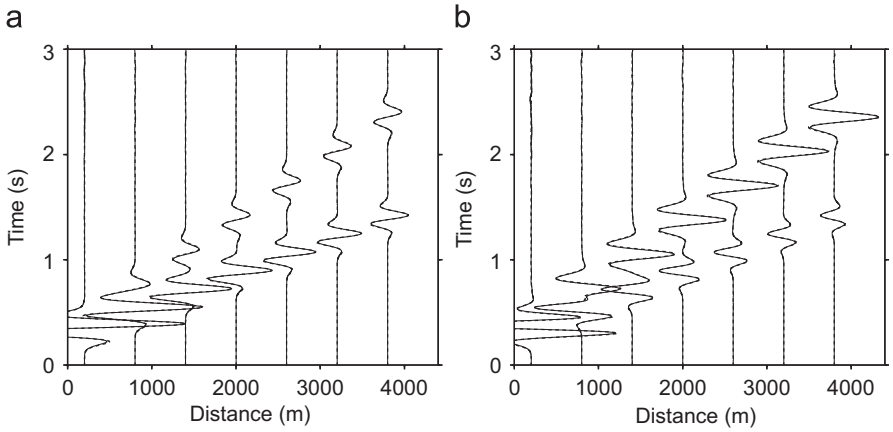

Fig. 2. Time-domain comparison between DG P_2 (solid lines) and analytical (dashed lines) solutions for the Garvin test. The seismograms represent: (a) the horizontal and (b) the vertical components of the particle velocity at the free surface.

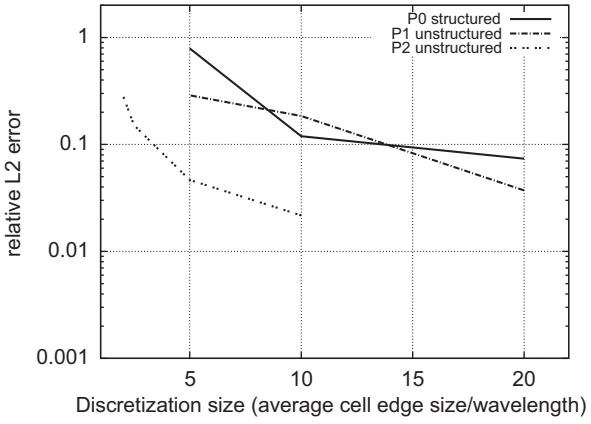

Fig. 3. Relative L_2 error ($\|u_{ref}-u\|^2/\|u_{ref}\|^2$) at the free surface between the analytical solution (u_{ref}) and the numerical solutions (u) computed with the DG P_0, P_1 and P_2 interpolation orders as a function of the mesh size. The relative L_2 error is computed for the 5 Hz frequency, and for the horizontal and vertical geophone components.

neighbouring cells in the DG formalism. An efficient mix is seen in P_0–P_1, because both interpolations share the same discretization rule (i.e., 10 cells per wavelength), and therefore, they facilitate the meshing at the transition zones between the P_0 and P_1 interpolations.

Table 1 summarizes the computational time and memory requirements for the DG method for two simple models, where the P_0, P_1, P_0–P_1 and P_2 orders of interpolation are used. These have, respectively, a discretization of: 10 cells per minimum wavelength in a structured equilateral mesh; 10 cells per local wavelength in an unstructured *h*-adaptive mesh; 10 cells per local wavelength with a structured/unstructured *hp*-adaptive mesh; and three cells per local wavelength in an unstructured *h*-adaptive mesh. It is worth remembering that these four combinations of interpolation order and discretization give similar accuracies (Fig. 3). Simulations are performed in parallel using 16 processors. The first model is a homogeneous half space, and the second one is a two-layer half space. The first simulation clearly shows that the P_0 interpolation on the regular mesh provides the best time and memory performances when the model contains a limited range of velocity. In the second model, the thickness of the upper layer is one tenth of the thickness of the half space, and the velocity in the upper layer is one tenth of the velocity of the lower layer. The upper layer mimics a complex weathering zone in the near surface, which requires a very fine mesh and, therefore, a dramatic increase in the number of cells when regular structured meshes are used (as it should also be with FD). The P_0–P_1 mesh uses *h*-adaptive unstructured cells (P_1) in the upper layer and structured equilateral cells (P_0) in the lower layer. In this case, the P_2 interpolation on the *h*-adaptive unstructured mesh provides the best time and memory performances, due to the local refinement of the mesh (Table 1). These two extreme canonical cases illustrate how the heterogeneity and the velocity range of the subsurface drive the choice of the mesh type and interpolation order. In the case of a broad velocity bandwidth in the model, P_2 interpolation used with *h*-adaptive unstructured meshes should give the best parameterization. In the case of a narrow velocity bandwidth, the P_0 or the P_0–P_1 interpolations should be more relevant. If a flat free surface is present without liquid, the P_0 interpolation can be used, while the P_0–P_1 interpolation should be used when the model contains a liquid/solid interface in a shallow water environment or a complex topography above a shallow weathered layer. For example, a layer of unstructured cells computed with P_1 allows discretization of a complex topography or water/solid interface of arbitrary shape, above a P_0 regular equilateral mesh, discretizing a more homogeneous medium. The P_0–P_1 approach is close in spirit to the hybrid FE/FD method proposed by Moczo et al. (1997). We can, however, note that the P_1 interpolation used alone never appears to be the optimal choice.

Figs. 4–6 further illustrate the mesh design for three realistic offshore and onshore models. The first represents a velocity model of the Valhall gas field in a shallow-water environment in the North Sea (Munns, 1985). The sea bed is at 70-m depth. The narrow velocity range in most parts of the model requires the use of a regular mesh as much as possible for computational efficiency. To correctly discretize the shallow-water layer, a *p*-adaptivity implemented with a mixed P_0–P_1 interpolation is chosen: a refined unstructured P_1 layer of cells is used for the first 130 m of the subsurface for accurate modelling of the interface waves at the liquid/solid interface, and for accurate positioning of the sources, which are located 5 m below the surface, and of the receivers, which are on the sea bed. The second example represents an onshore foothill model that contains a complex topography, several dipping structures, and a thin weathered layer at the surface (V_P=700 m/s, V_S=350 m/s): a dramatic configuration for regular grid-based methods. An *h*-adaptive unstructured mesh that is suitable for the P_2 interpolation order was designed for accurate modelling of the free-surface effects of the topography, and for the local adaptation of the cell size to the local wave speeds. The third example corresponds to a crustal-scale model, which represents the Nankai subduction zone, offshore of Japan (Operto et al., 2006). We use a P_2 *h*-adaptive unstructured mesh to deal with the broad velocity bandwidth spanned by the crustal model. For FWI applications in such kind of environments, the *hp*-adaptive

Table 1
Computational resources required for the forward problem solved with DGs P_0, P_1, P_0–P_1 and P_2, in two simple cases, on 16 processors.

Test	Resource	P_0	P_1	P_0–P_1	P_2
Homogeneous	Cell numbers	113 097	136 724	116 363	12 222
	Degrees of freedom	565 485	2 050 860	695 795	366 660
	Time for factorization (s)	2.1	59.2	4.1	9.4
	Memory for factorization (Gb)	2.39	19.80	3.09	4.57
	Time to solve 116 RHS (s)	6.2	28.3	8.1	6.1
Two-layers	Cell numbers	2 804 850	291 577	247 303	32 664
	Degrees of freedom	14 024 250	4 373 655	2 360 405	979 920
	Time for factorization (s)	333.2	109.5	31.8	22.1
	Memory for factorization (Gb)	80.50	33.26	14.37	10.84
	Time to solve 116 RHS (s)	190.4	72.4	30.2	16.2

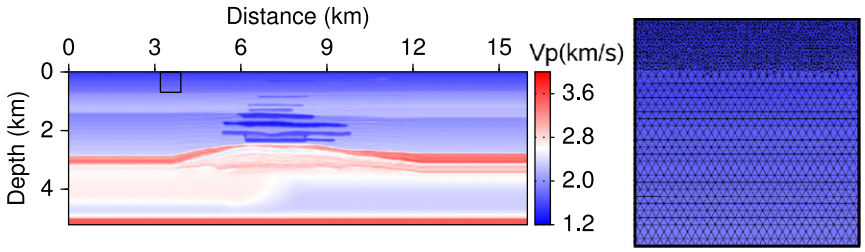

Fig. 4. The synthetic Valhall P-wave velocity model representative of oil and gas shallow water targets, and a close-up of the mesh used for mixed DG P_0–P_1 simulations.

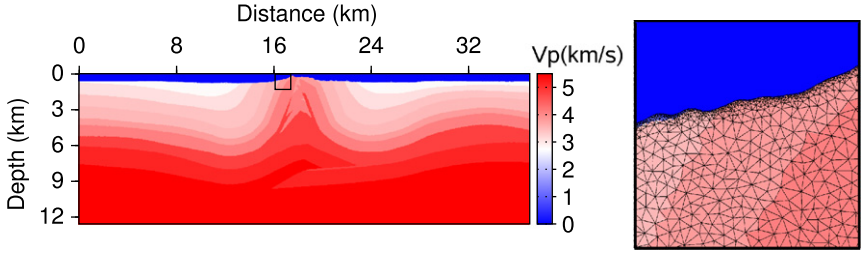

Fig. 5. The synthetic IFP/TOTAL Foothill P-wave velocity model representative of onshore targets with complex topographies and weathering zones, and a close-up of the mesh for DG P_2 simulations.

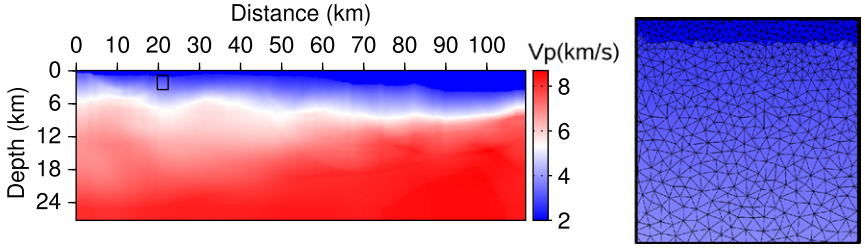

Fig. 6. The crustal P-wave velocity model representative of the Japanese Nankai trough, and a close-up of the *h*-adaptive unstructured mesh for DG P_2 simulations.

strategies just mentioned can be used during the forward modelling. The unstructured mesh is designed before inversion according to the velocity distribution in the starting model. As the velocity model is updated over FWI iterations, a quality control of the mesh can be necessary to check that the discretization criterion of the forward problem is still satisfied. If not, a remeshing can be necessary to locally refine the mesh according to the velocity update. This quality control and the remeshing are not automated so far and require human intervention. The meshing and remeshing steps are performed with Triangle software (Shewchuk, 1996) that allows a local control of cells size.

3. Inverse problem

3.1. Theory review

We start with a brief review of FWI. A more extensive overview is presented in Virieux and Operto (2009). FWI is an optimization problem that is generally recast as a local non-linear least-squares problem that attempts to minimize the misfit between the recorded and the modelled wavefields (Tarantola, 1987). The inverse problem can be formulated in the frequency-domain (Pratt and Worthington, 1990), and the associated objective function to be minimized is

defined by

$$\mathcal{C}(\mathbf{m}) = \sum_{if=1}^{nf} \sum_{is=1}^{ns} \frac{1}{2} \Delta\mathbf{d}(\mathbf{m})^t \mathbf{W}_d \overline{\Delta\mathbf{d}}(\mathbf{m}), \tag{4}$$

where $\Delta\mathbf{d}(\mathbf{m}) = \mathbf{d}_{obs} - \mathbf{d}_{calc}(\mathbf{m})$ is the data misfit vector, the difference between the observed data $\mathbf{d}_{obs}$ and the modelled data $\mathbf{d}_{calc}(\mathbf{m})$ computed in the model $\mathbf{m}$ updated at the iteration n of the inversion. $\mathbf{d}_{calc}(\mathbf{m})$ is obtained by applying a sampling operator $\mathbf{S}$ to the incident wavefield $\mathbf{u}$. Superscripts ^{t} and $^-$ indicate the transposed and the conjugate, respectively, and $\mathbf{W}_d$ is a diagonal weighting matrix that is applied to the misfit vector to scale the relative contributions of each of its components. The summations in Eq. (4) are performed over the ns sources and a group of nf simultaneously inverted frequencies.

With the objective function assumed to be locally parabolic, its minimization with the Newton method provides the following expression of the perturbation model $\delta\mathbf{m}^{(n)}$:

$$\mathbf{B}^{(n)} \delta\mathbf{m}^{(n)} = -\mathcal{G}^{(n)}, \tag{5}$$

where $\mathbf{B}^{(n)}$ and $\mathcal{G}^{(n)}$ denote the Hessian matrix and the gradient of the objective function at iteration n, respectively.

The gradient $\mathcal{G}$ of the objective function with respect to the model parameters $\mathbf{m} = \{m_i\}_{i=1,N}$, where N denotes the number of unknowns, can be derived from the adjoint-state formalism using the back-propagation technique (Plessix, 2006). The adjoint wavefield λ can therefore be computed as

$$\mathbf{A}\overline{\lambda} = \mathbf{S}^t \mathbf{W}_d \overline{\Delta\mathbf{d}}, \tag{6}$$

considering that the matrix $\mathbf{A}$ is symmetric in the domain of interest (outside the PML) thanks to the reciprocity of Green functions. This gives for the ith component of the gradient $\mathcal{G}$:

$$\mathcal{G}_{m_i} = -\sum_{if=1}^{nf} \sum_{is=1}^{ns} \Re\left\{ \mathbf{u}^t \frac{\partial \mathbf{A}^t}{\partial m_i} \overline{\lambda} \right\}, \tag{7}$$

where $\Re$ denotes the real part of a complex number. The gradient is computed by a zero-lag correlation in time, between the incident wavefield $\mathbf{u}$ from the source, and the adjoint back-propagated wavefield $\overline{\lambda}$, using residuals at receiver positions as a composite source. Therefore, only two forward problems per shot are required for gradient building. The radiation pattern of the diffraction by the model parameter m_i is denoted by the sparse matrix $\partial\mathbf{A}/\partial m_i$.

Due to the computational cost of building the Hessian operator and solving Eq. (5), Newton and Gauss–Newton methods are generally not considered for realistically sized problems (Pratt et al., 1998). Steepest-descent or conjugate-gradient methods that are preconditioned by the diagonal terms of an approximate Hessian are more conventionally used to compute $\delta\mathbf{m}^{(n)}$ (Pratt et al., 1998; Operto et al., 2006; Shin et al., 2001).

The model is then updated with an optimal step-length $\alpha^{(n)}$, computed by a linear inversion (Tarantola, 1987) or by parabola fitting (Sourbier et al., 2009a)

$$\mathbf{m}^{(n+1)} = \mathbf{m}^{(n)} + \alpha^{(n)} \delta\mathbf{m}^{(n)}. \tag{8}$$

3.2. Inversion algorithm

CPU-efficient frequency-domain FWI is generally carried out by successive inversions of single frequencies, by proceeding from the low frequencies to the higher ones (Pratt and Worthington, 1990; Sirgue and Pratt, 2004). This defines a multiresolution framework that mitigates the nonlinearity of the inverse problem associated with high frequency cycle-skipping artefacts. CPU-efficient algorithms can be designed by selecting a few coarsely sampled frequencies, such that the wavenumber redundancy, which results from dense sampling of frequencies and aperture angles, is decimated. This strategy has proven to be effective for several applications of acoustic FWI (Ravaut et al., 2004; Operto et al., 2006; Brenders and Pratt, 2007a). However, it might lack robustness when complex wave phenomena are present, such as P-to-S conversions, multiples and surface waves.

A two-level hierarchical procedure (Algorithm 1) to mitigate the FWI nonlinearities in the case of elastic onshore problems was proposed by Brossier et al. (2009a). This algorithm embeds three main loops:

1. The outer loop is over the frequency groups; namely, a set of frequencies that are simultaneously inverted. In the case of complex wave phenomena, the use of simultaneous frequencies in the inversion procedure constrains better the optimization for convergence towards the global minimum, taking into account the more redundant information contained in the simultaneous frequencies.
2. The second loop is over the time-damping factors that control the amount of information over time that is preserved in the seismograms for inversion. Time-damping is performed in frequency-domain modelling through the use of complex-valued frequencies, which is equivalent to damp seismograms in time by an exponential decay (Shin et al., 2002; Brenders and Pratt, 2007b). Moreover, this data preconditioning provides a significant improvement in the signal-to-noise ratio with real data (Brenders et al., 2009).
3. The third loop is over iterations of the non-linear problem for one frequency group inversion.

Algorithm 1. Two-level hierarchical frequency-domain FWI algorithm

```
for ig = 1 to ng do
  for id=1 to nd do
    while (NOT convergence AND n < n_max)
      for if=1 to nf do
        Compute incident wavefields u from sources
        Compute residual vectors Δd and cost function C
        Compute back-propagated wavefields λ̄
        Build gradient vector G^(n) (with Algorithm 2)
      end for
      Compute perturbation vector δm^(n) (with Algorithm 3)
      Define optimal step length α^(n) by parabola fitting
      Update model m^(n+1) = m^(n) + α^(n) δm^(n)
    end while
  end for
end for
```

This algorithm was applied successfully for imaging the elastic parameters of onshore and offshore synthetic data in Brossier et al. (2009a, b).

3.3. Gradient computation

The gradient computation with the adjoint-state method is described formally by Eq. (7). For computational time and memory use efficiency, some adaptations and approximations can be applied to compute the gradient vector.

Let me first introduce several notations. The matrix **A1** denotes the impedance matrix resulting from the discretization of the first-order velocity/stress formulation of the elastodynamic equations (Eq. (2)). The matrix **A2** denotes the impedance matrix resulting from the discretization of the second-order velocity formulation of the elastodynamic equations. This matrix can be derived from the first-order velocity/stress system through a *parsimonious* formulation (Luo and Schuster, 1990), as developed in Brossier et al. (2008)

for DG P_0. Note, however, that building the **A2** matrix through the *parsimonious* formulation is more complicated for higher order of interpolation, particularly when p-adaptivity is used. This complexity drove me to perform the forward problem with the first-order matrix **A1**. Subscript p_k denotes the k order of interpolation used during forward problem modelling. The projection matrix $\mathbf{P}_{P_0}$ interpolates a p_k wavefields vector at the barycentre of each cell, giving a P_0 representation of fields. Finally the restriction operator $\mathbf{P}_V$ allows one to extract only the velocity components from a velocity/stress wavefield vector.

Considering these notations, Eq. (7) can be rewritten as

$$\mathcal{G}_{m_i} = -\sum_{if=1}^{nf}\sum_{is=1}^{ns} \Re\left\{ \mathbf{u}_{p_k}^t \frac{\partial \mathbf{A1}_{p_k}^t}{\partial m_i} \overline{\boldsymbol{\lambda}}_{p_k} \right\}. \tag{9}$$

Because the first-order velocity/stress and the second-order velocity systems are theoretically equivalent for the computing of the velocity wavefields, Eq. (9) can be rewritten, without approximation, as

$$\mathcal{G}_{m_i} = -\sum_{if=1}^{nf}\sum_{is=1}^{ns} \Re\left\{ \mathbf{u}_{p_k}^t \mathbf{P}_V^t \frac{\partial \mathbf{A2}_{p_k}^t}{\partial m_i} \mathbf{P}_V \overline{\boldsymbol{\lambda}}_{p_k} \right\} = -\sum_{if=1}^{nf}\sum_{is=1}^{ns} \Re\left\{ \mathbf{u}_V^t \frac{\partial \mathbf{A2}_{p_k}^t}{\partial m_i} \overline{\boldsymbol{\lambda}}_V \right\}. \tag{10}$$

This expression allows one to compute the gradient $\mathcal{G}$ considering only the velocity wavefields ($\mathbf{u}_V$, $\overline{\boldsymbol{\lambda}}_V$), the stress wavefields are therefore not required. Since the incident and adjoint wavefields are stored in core-memory for all the sources in the algorithm, a reduction of the memory storage of the wavefields by a factor two and a half is obtained keeping only velocity wavefields.

Since the matrix **A2** is difficult to build for P_1 and P_2 interpolation orders, Eq. (10) is not used in practice. Instead, we perform the modelling with the first-order velocity–stress matrix $\mathbf{A1}_{p_k}$ as mentioned above and extract the modelled data $\mathbf{d}_{calc}$ at receiver positions (either, the pressure or the particle velocities depending whether the hydrophone or the geophone components are inverted) for each source before deleting the stress wavefields for gradient computation. Second, we use a P_0 radiation pattern matrix $\partial \mathbf{A2}_{p_0}/\partial m_i$ instead of a P_k one, because the second-order impedance matrix $\mathbf{A2}_{p_0}$ is easier to build. The P_0 representation of the radiation pattern requires to project the P_k incident and the P_k adjoint velocity wavefields onto the P_0 mesh. This projection step leads to an additional saving of memory and computational time compared to that achieved with Eq. (10).

This leads to the following approximate expression of the gradient:

$$\begin{aligned}\mathcal{G}_{m_i} &= -\sum_{if=1}^{nf}\sum_{is=1}^{ns} \Re\left\{ \mathbf{u}_{p_k}^t \mathbf{P}_V^t \mathbf{P}_{P_0}^t \frac{\partial \mathbf{A2}_{p_0}^t}{\partial m_i} \mathbf{P}_{P_0} \mathbf{P}_V \overline{\boldsymbol{\lambda}}_{p_k} \right\} \\ &= -\sum_{if=1}^{nf}\sum_{is=1}^{ns} \Re\left\{ \mathbf{u}_{V/P_0}^t \frac{\partial \mathbf{A2}_{p_0}^t}{\partial m_i} \overline{\boldsymbol{\lambda}}_{V/P_0} \right\}. \end{aligned} \tag{11}$$

The approximation of the gradient, resulting from the P_0 projection of the P_k incident and adjoint wavefields and from the P_0 radiation pattern matrix on coarse mesh, can be justified by the fact that the gradient computation does not need to be discretized with the same order of accuracy as the impedance matrix used in the forward problem (Sourbier et al., 2009b). This approximation allows one to save a factor of three and six, both in computational time to build the gradient and in core-memory to store the velocity wavefields, when the forward problem is solved with DG P_1 and P_2, respectively. This approximation introduces, however, some errors in the gradient. In the unfavourable case of P_2 interpolation and coarse meshes of four cells per wavelength, these errors have been estimated to be less that nine percents on individual sensitivity kernels, which affects marginally the inversion results in the framework of iterative FWI. Please note, however, that only the incident and adjoint wavefields are P_0 projected for gradient building. The data residual $\Delta\mathbf{d}$ is computed without approximation using the k interpolation order before the P_k modelling of the adjoint wavefield.

Finally, the gradient vector is computed through Eq. (11) with Algorithm 2.

Algorithm 2. Algorithm of gradient computation

1: **for** $m_i = 1$ to m_N **do**
2: Compute $\partial \mathbf{A2}_{P_0}^t/\partial m_i$
3: **for** $is = 1$ to ns **do**
4: Compute $\boldsymbol{\beta} \leftarrow (\partial \mathbf{A2}_{P_0}/\partial m_i)\mathbf{u}_{V/P_0}$ through full vector/sparse matrix product routine
5: Compute $\mathcal{G}_{m_i} \leftarrow \mathcal{G}_{m_i} - \Re(\boldsymbol{\beta}^t \overline{\boldsymbol{\lambda}}_{V/P_0})$ with sparse vector/full vector dot product routine
6: **end for**
7: **end for**

3.4. L-BFGS and multi-parameter inversion

FWI is generally solved with the steepest-descent or conjugate-gradient methods, preconditioned by a diagonal approximation of the Hessian (Pratt et al., 1998; Operto et al., 2006; Shin et al., 2001). Limited-memory, quasi-Newton methods, such as the limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) method (Nocedal, 1980), can, however, provide an efficient alternative to classic steepest-descent or conjugate-gradient methods for FWI problems (Epanomeritakis et al., 2008; Symes, 2008; Brossier et al., 2009a).

The Hessian matrix allows for: correction of the gradient from the geometrical amplitude spreading of the incident and adjoint wavefields; improvement of the imaging resolution by correctly deconvolving the gradient from limited-bandwidth effects, taking into account the off-diagonal terms of the Hessian; suitable scaling of the model perturbation vector for different parameter classes; and accounting for double scattering effects (Pratt et al., 1998; Nocedal and Wright, 1999). L-BFGS provides an approximation of the inverse $\mathbf{H}$ of the Hessian matrix from few gradient and model difference vectors, stored from previous iterations. Moreover, the double-loop recursive algorithm designed by Nocedal (1980) does not explicitly build and store $\mathbf{H}^{(n)}$, but directly computes the perturbation vector $\delta\mathbf{m}^{(n)} = -\mathbf{H}^{(n)}\mathcal{G}^{(n)}$ with additions, differences and inner products of vectors. A diagonal approximation of the inverse Hessian computed from the diagonal terms of the approximate Hessian (Pratt et al., 1998; Operto et al., 2006) or pseudo-Hessian (Shin et al., 2001), can be provided to the L-BFGS algorithm for a better and faster estimation of $\mathbf{H}^{(n)}\mathcal{G}^{(n)}$.

In the framework of the inversion of multiple classes of parameter with different units and amplitudes (for example, reconstruction of wave speeds, density, attenuation factors), the L-BFGS algorithm requires some normalizations for consistent dimensionless computation of the model perturbation vector. Algorithms 3 and 4 review the computation of the perturbation model with appropriate normalizations that account for the different classes of parameters in the L-BFGS algorithm (Nocedal, 1980, modified following Gratton (personal communication)). The normalization of the gradient, the diagonal Hessian, and the perturbation model are performed according to a characteristic value m_{i0} that is defined for each parameter class, taken as the mean of the initial model of each parameter class. This normalization is

equivalent to the computation of the gradient and the diagonal Hessian for normalized parameters $\mathbf{m}_i/m_{i0}$, where i denotes the parameter class.

Algorithm 3. Perturbation computation: outer part

if $n=1$ **then**
  Compute diagonal of pseudo-Hessian $\mathbf{B_0}$ (Shin et al., 2001)
  for $i=1$ to $n_{parameter\_class}$ **do**
    Normalize diagonal of Hessian $\tilde{\mathbf{B}}_{\mathbf{0}\,\mathbf{m_i}} = \mathbf{B}_{\mathbf{0}\,\mathbf{m_i}} m_{i0}^2$
  end for
  Apply damping to the diagonal of Hessian (Levenberg–Marquardt method) $\tilde{\mathbf{B}}_{\mathbf{0}} = \tilde{\mathbf{B}}_{\mathbf{0}} + \varepsilon \mathbf{I}$
end if
for $i=1$ to $n_{parameter\_class}$ **do**
  Normalize gradient $\tilde{\mathcal{G}}^{(n)}_{\mathbf{m_i}} = \mathcal{G}^{(n)}_{\mathbf{m_i}} m_{i0}$
end for
Compute $\tilde{\mathcal{D}}^{(n)} = \tilde{\mathbf{B}}_{\mathbf{0}}^{\ -1} \tilde{\mathcal{G}}^{(n)}$
if $n > k$ **then**
  Discard the vector pair $(\mathbf{y}_{n-k-1}, \mathbf{s}_{n-k-1})$
end if
if $n > 1$ **then**
  Compute and store $\mathbf{s}_{n-1} \leftarrow \mathbf{m}^{(n)} - \mathbf{m}^{(n-1)}$
  Compute and store $\mathbf{y}_{n-1} \leftarrow \tilde{\mathcal{D}}^{(n)} - \tilde{\mathcal{D}}^{(n-1)}$
end if
Compute $\delta\tilde{\mathbf{m}}^{(n)}$ with Algorithm 4
for $i=1$ to $n_{parameter\_class}$ **do**
  Denormalize perturbation vector $\delta\mathbf{m}^{(n)} = \delta\tilde{\mathbf{m}}^{(n)} m_{i0}$
end for

Algorithm 4. Perturbation computation: L-BFGS algorithm (Nocedal and Wright, 1999, p. 178), modified following Gratton (personal communication)

$\mathbf{q} \leftarrow \tilde{\mathcal{D}}^{(n)}$
for $i=n-1$ to $n-k$ **do**
  $\rho_i \leftarrow \dfrac{1}{\mathbf{y}_i^t \mathbf{s}_i}$
  $\alpha_i \leftarrow \rho_i \mathbf{s}_i^t \mathbf{q}$
  $\mathbf{q} \leftarrow \mathbf{q} - \alpha_i \mathbf{y}_i$
end for
$\gamma_n \leftarrow \frac{\mathbf{s}_{n-1}^t \mathbf{y}_{n-1}}{\mathbf{y}_{n-1}^t \mathbf{y}_{n-1}}$
$\mathbf{r} \leftarrow \gamma_n \mathbf{q}$
for $i=n-k$ to $n-1$ **do**
  $\beta \leftarrow \rho_i \mathbf{y}_i^t \mathbf{r}$
  $\mathbf{r} \leftarrow \mathbf{r} + \mathbf{s}_i(\alpha_i - \beta)$
end for
$\delta\tilde{\mathbf{m}}^{(n)} = -\mathbf{r}$

Fig. 7 shows the FWI results in a canonical configuration: 113 horizontal point force sources are located all around the target each 50 m, and each source wavefield is recorded by 112 two-component geophones all around the target each 50 m. Absorbing boundary conditions are used on the four sides of the domain to simulate an infinite medium. The true models contain one circular heterogeneity of 200 m diameter at different locations for each of the five parameters: P-wave and S-wave velocities (V_P and V_S), density (ρ), and P-wave and S-wave quality factors (Q_p and Q_s) (Fig. 7). Table 2 summarizes the background and heterogeneity values for each parameter. The inversion is performed for 10 frequencies between 4 and 13 Hz each 1 Hz, starting from homogeneous background models. The inverted parameters are λ, μ (the Lamé parameters), ρ, Q_p and Q_s, to deliberately invert parameters with contrasted amplitudes (Q_p and $Q_s \backsim 10^1 \sim 10^2$ while λ and $\mu \backsim 10^8 \sim 10^9$). Despite these differences, the normalization procedure allows for a correct estimation of the Hessian through L-BFGS, and allows for a convergence towards acceptable models without any need for heuristic weighting of each parameter class. Note that the weighting provided by the inverse Hessian estimation does not require the performing of extra forward problems, whereas the subspace method of Sambridge et al. (1991), which is generally used in multi-parameter inversions, requires the performing of one extra forward problem per parameter class to estimate the steplength. Note, however, that in Fig. 7 there are some coupling effects between some parameters that result from the physics of the diffraction pattern of each parameter class, and not from the choice of the optimization method.

The algorithm allows one to reconstruct one quality factor for each wave type (Q_p and Q_s). A non-dispersive attenuation model is currently used for both modelling and inversion. A dispersive model, as the Kolsky–Futterman model (Kolsky, 1956; Futterman, 1962), could, however, be taken into account without extra computational cost. Note, however, that reliable reconstruction of attenuation factors from seismic data is a difficult task that strongly depends on acquisition geometry (Mulder and Hak, 2009). A detailed analysis of the ability of my algorithm to reconstruct attenuation is beyond the scope of this paper.

The reader can refer to Brossier et al. (2009a, b) for realistic applications of that strategy to onshore and offshore synthetic data from the imaging of P-wave and S-wave velocity models.

4. Parallelization and performance

To tackle imaging of realistically sized targets, the computationally demanding FWI method needs to be implemented in parallel to exploit multiple CPU facilities, also with a large amount of available core-memory, compared to sequential execution.

4.1. Forward problem

The forward problem of FWI, as the resolution of the linear system of Eq. (3) for multiple sources (right-hand-side (RHS)), is performed with the massively parallel direct solver MUMPS (Amestoy et al., 2006), which performs a LU decomposition of the matrix $\mathbf{A}$, based on a multi-frontal approach (Duff and Reid, 1983). The parallelization allows for the speeding up of the factorization by more than one order of magnitude, compared to sequential execution, and for the distribution of the LU factors in the core-memory over all of the processors (Sourbier et al., 2009b), which makes this quite efficient for solving large problems without intensive I/O resources. The MUMPS software is based on message passing interface (MPI) standards for applications with distributed memory architectures. The workflow of the MUMPS algorithm can be summarized as: (1) analysis of the $\mathbf{A}$ matrix pattern (note that unknowns are treated independently in the current version of the algorithm, without taking advantage of the block structure provided by elements), followed by reordering of the matrix to limit the fill-in during factorization. The analysis step is performed sequentially on the master processor with the current MUMPS version (but does not modify the workflow if done in parallel); (2) parallel factorization of the $\mathbf{A}$ matrix with dynamic pivoting; (3) parallel forward and backward substitutions to obtain solution vectors $\mathbf{u}$ from RHS vectors $\mathbf{b}$.

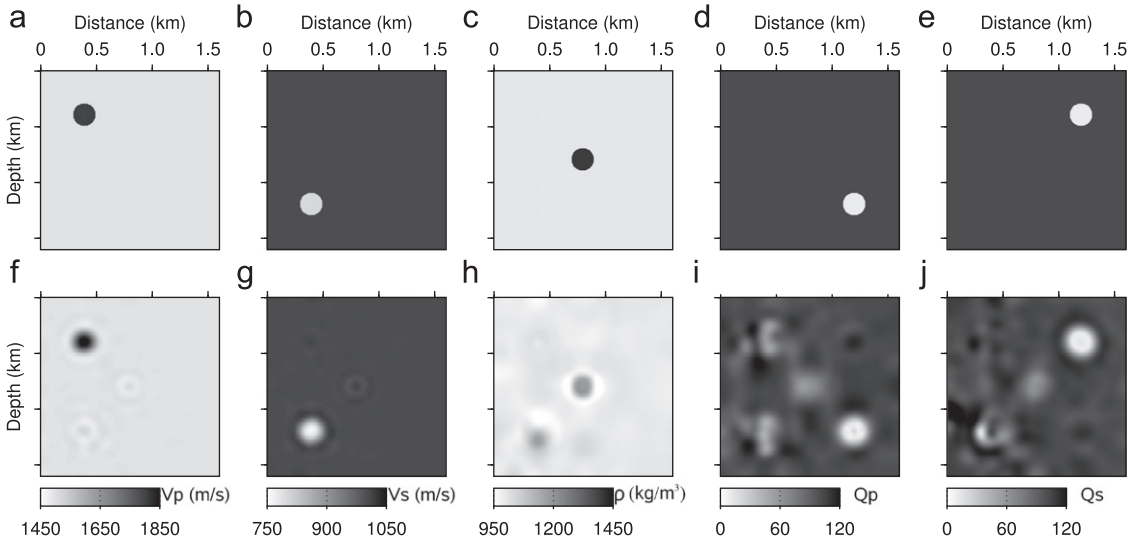

Fig. 7. Example of multi-parameter inversion. (a–e) True models: (a) V_P, (b) V_S, (c) ρ, (d) Q_P and (e) Q_S models. (f–j) Reconstructed FWI models: (f) λ, (g) μ, (h) ρ, (i) Q_P and (j) Q_S.

Table 2
Physical parameters in the background and heterogeneities for the canonical inversion test.

Parameter	Background value	Heterogeneity value
V_P (m/s)	1500	1800
V_S (m/s)	1000	800
ρ (kg/m^3)	1000	1400
Q_P	100	10
Q_S	100	10

4.2. Overview of the parallel inverse problem

In the framework of iterative FWI, the physical properties of the model change at each iteration, but the mesh geometry is not changed. Therefore, the **A** matrix keeps the same pattern, allowing a single analysis phase in the whole inversion procedure. This feature is interesting because the matrix analysis is the only time-demanding step performed sequentially on the master process. After the substitution step, the solution vectors **u** are returned by MUMPS in a distributed form over the in-core memory of the processors. However, the distribution of the solutions performed by MUMPS does not always provide a well-balanced load for the processors, because the MUMPS strategy is to optimize LU decomposition with accurate dynamic pivoting and reduced fill in. The distribution is therefore not suitable for the application of the projection operator $\mathbf{P}_{P_0}\mathbf{P}_V$ to the local in-core solutions (i.e., the solutions at the different nodes for one wavefield of a triangular cell are not systematically stored in-core on the same processor). To overcome this limitation, the solutions are re-ordered following well-balanced mesh partitioning, performed with the METIS software (Karypis and Kumar, 1999). The MUMPS-distributed solutions are mapped to the METIS decomposition with MPI point-to-point communications, before the application of operator $\mathbf{P}_{P_0}\mathbf{P}_V$. This mapping/projection step represents about 25% of the MUMPS substitution time. The forward and back-propagated wavefields are stored in the in-core memory for all sources. The gradient and the diagonal of the Hessian are therefore efficiently computed in parallel with local in-core wavefields for the local subdomain associated with each process, before being centralized on the master processor for perturbation-model building. Of note, the assemblage and storage of the sparse **A** matrix are performed by the master process, as the storage of the mesh-related tables, the wavefield solutions at the receiver positions for computing the objective function, and the composite residual sources for adjoint-wavefield computation. To avoid prohibitive memory allocation on the master processor, the master process is not involved in the factorization and substitution/projection steps, the wavefield storage, and the gradient/Hessian computation.

4.3. Two parallelism levels

The scalability of parallel direct solvers is intrinsically limited because of the large amount of communications between the MPI processes, and the memory overhead generated by the number of MPI process involved in the factorization (Sourbier et al., 2009b). Therefore, the single parallelization level with MPI cannot be adapted for large-scale applications. A second parallelism level, based on multi-threading, can, however, be efficiently implemented on multi-core architectures. The main computationally demanding parallel tasks of the FWI algorithm are the forward problem resolutions (factorization and substitutions/projection) and the building of the gradient and diagonal Hessian.

The MUMPS solver uses the Basic Linear Algebra Subroutines (BLAS) library mainly for level 2 and 3 operations (matrix/vector, and matrix/matrix operations). Several BLAS libraries are available in multi-thread distributions, which allow an efficient low-level of parallelization for multi-core architectures with shared memory.

The gradient and diagonal Hessian computations have several nested loops embedded (Algorithm 2). In addition to the MPI-based domain-decomposition parallelism, the outer loop over the local model parameters can be easily parallelized with shared-memory thread technology (as with standard OpenMP) to speed up the loop on the available computing-cores without duplicating or exchanging data.

An alternative to classical parallelization using one MPI process per computing-core is, therefore, to run one MPI process on n computing-cores that share the same memory (physically on the same node), and to launch n shared-memory threads per MPI-process during the BLAS operations, and the gradient and diagonal Hessian computations. These two levels of parallelism can mitigate the memory overhead of the factorization, the network usage for intensive communications, and the computational time for LU factors.

A third level of parallelism could also be implemented over frequencies and/or time-damping factors (Shin and Min, 2006). This third level of parallelism has, however, not been investigated in the current algorithm.

4.4. Performance

Simulations are performed on a distributed-memory architecture composed of 18 nodes, each of which includes two quad-core 2.7 GHz Opteron processors with 64 GB of shared memory. The nodes are interconnected by an Infiniband ConnectX network. This cluster is located at Geoazur Institute, and it provides 144 computing-cores and 1.125 TB of memory.

A first test corresponds to a middle-scale application that is focused on the elastic Valhall model (Fig. 4). A 1 042 720-cell mesh was designed with the Triangle software (Shewchuk, 1996). The mesh is composed of a thin unstructured layer on top for P_1 interpolation, and a large zone of equilateral structured triangles for P_0 in depth. This mesh allows frequencies of up to 7 Hz to be modelled in the elastic case. The substitution/projection phase is performed for 315 RHSs, corresponding to 315 explosive sources. Fig. 8 shows the evolution of the computational times and the memory usage for the main inversion steps as a function of the number of computing-cores, using one- and two-level parallelism.

We clearly see a decrease in the computational time when the computing-cores number increases, although the efficiency is limited by the intrinsic scalability of the MUMPS direct solver. The best trade-off between performance and computational resources is obtained with 24 computing-cores, with one MPI-process per computing-core used. When two levels of parallelism are used with shared-memory threads, the in-core memory required by the factorization decreases, an appealing feature if the memory available is limited. However, the memory saving is at the expense of the computational time, particularly for the substitution/projection step. Here, I postulate that for this middle-scale application, the speed-up provided by the multi-thread computation is not significant with respect to the time overheads due to thread launching, leading to a limited time performance for this application.

A second test focuses on a large-scale elastic crustal model (Fig. 6). An h-adaptive unstructured mesh was designed with Triangle for DG P_2 modelling up to 6.5 Hz, leading to 528 080 cells. To mimic the SFJ-OBS experiment (Operto et al., 2006), 93 sources were considered, corresponding to 93 ocean-bottom seismometers, for the substitution/projection phase. Fig. 9 summarises the computational times and memory usage for this test. Note that the time for gradient building is negligible compared to that of the MUMPS steps, due to the small number of RHSs and the cell number in the mesh. For this large-scale application that requires a large amount of distributed memory to store the 200 GB of LU factors, shared-memory-threads technology is the alternative choice to the one-level parallelism, to minimize the number of MPI processes, and therefore the memory overheads during LU factorization. The data obtained with multiple threads show that both the computational time for the factorization and the substitution/projection steps can be decreased, while saving a significant amount of core memory.

The results of these two applications suggest that the two-level parallelism over MPI processes and shared-memory threads that are embedded in my FWI algorithm provide an efficient and flexible tool to tackle realistic size target imaging. Depending on the computing facilities available and the size of the case study, a judicious combination of the two levels of parallelism can be found for optimal performances in terms of computational time and core-memory usage.

5. Conclusion

A massively parallel algorithm was developed to perform frequency-domain full-waveform modelling and inversion for imaging 2D visco-elastic media. The forward problem of inversion, as the resolution of the elastodynamics equation in the frequency-domain, is solved using a low-order finite element discontinuous Galerkin method. This

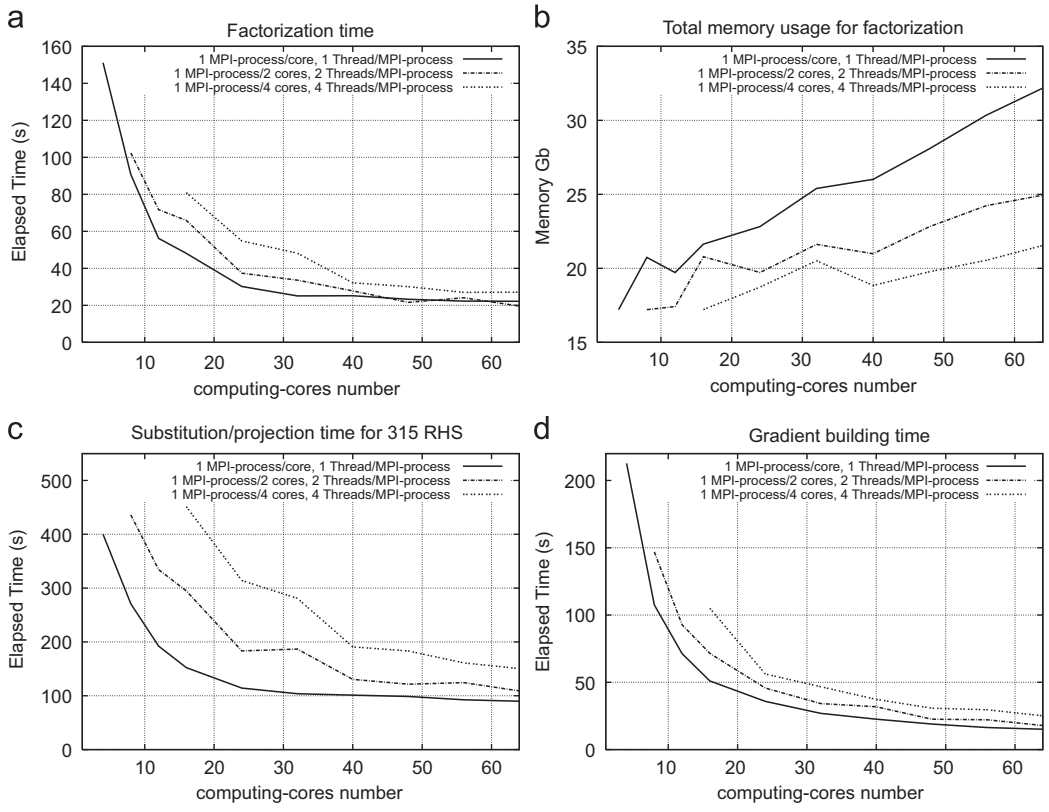

Fig. 8. Computational time and memory use for the main inversion tasks for a middle-scale application, as a function of the number of computing-cores, and the parallel strategy, as indicated.

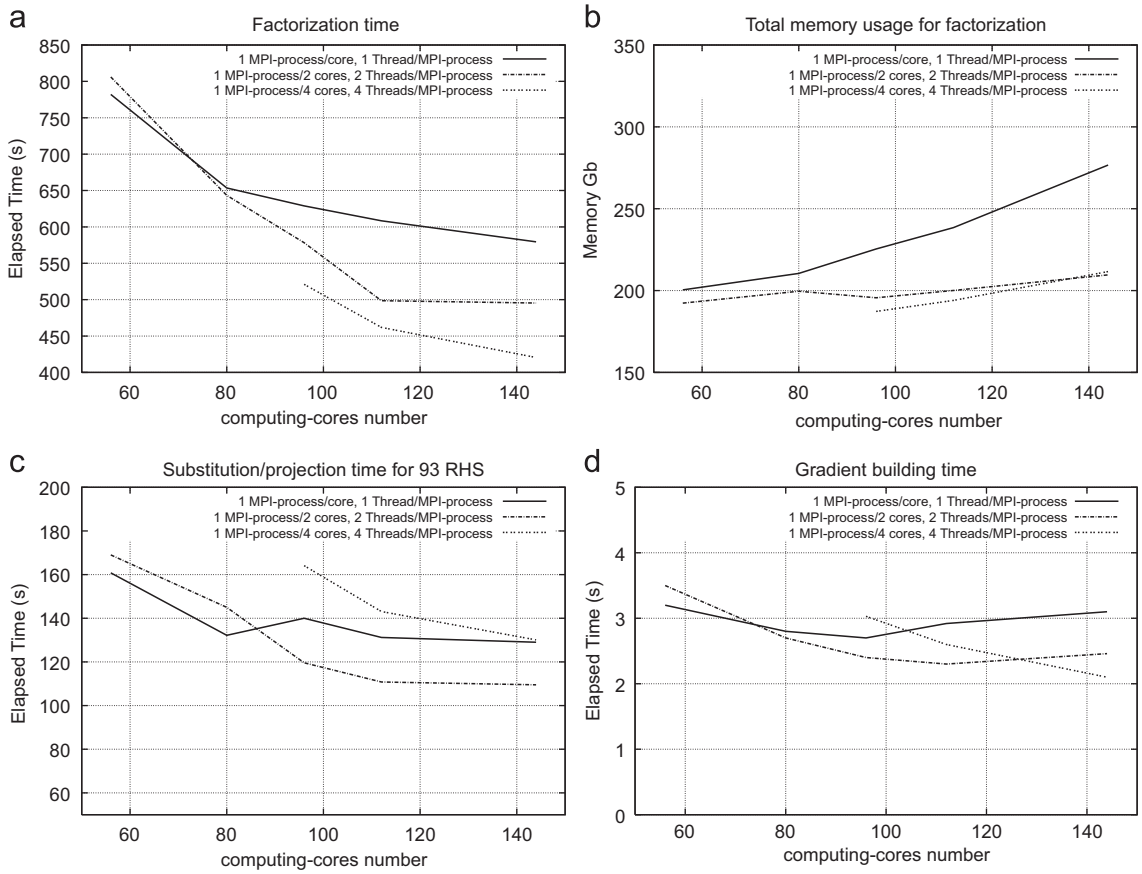

Fig. 9. Computational time and memory use for the main inversion tasks for a large-scale application, as a function of the number of computing-cores, and the parallel strategy, as indicated.

allows for complex topographies and high velocity-contrasts to be taken into account, and to use unstructured *h*-adaptive triangular meshes combined with *p*-adaptive interpolations. The linear system, resulting from this discretization is solved with the parallel MUMPS direct solver, allowing the LU factors to be stored in a distributed form over the processors. The wavefield solutions are also stored in a distributed form, following mesh partitioning for a well-balanced workload over the processors. The inverse problem of the FWI is solved with a non-linear local optimization scheme. The implementation of the gradient of the objective function has been optimized for computational time and memory saving. A quasi-Newton L-BFGS optimization has been implemented to estimate the inverse of the Hessian matrix at a negligible computational cost, improving the reconstruction of several classes of parameter (P-wave and S-wave speeds, density, and attenuation factors Q_P and Q_S). A two-parallelism level through MPI-processes and shared-memory threads has been implemented for optimal use of the computational time and core-memory resources, which will depend on the computer facilities.

This FWI algorithm will be used to tackle the imaging of realistically sized targets at various scales, from near-surface geotechnic applications to crustal-scale exploration. Ongoing studies include the extension to anisotropic (vertically transverse isotropic) media and the adaptation to teleseismic configurations for lithospheric imaging.

Acknowledgements

This study was funded by the SEISCOPE consortium (http://seiscope.oca.eu), which is sponsored by BP, CGG-VERITAS, ENI, EXXON-MOBIL, SHELL and TOTAL, and by Agence Nationale de la Recherche (ANR) under project ANR-05-NT05-2-42427. The author is grateful to S. Operto (Géoazur, CNRS) and J. Virieux (LGIT, Université Joseph Fourier) for fruitful and stimulating discussions on full waveform modelling and inversion, on parallel implementation, and for manuscript review. Many thanks also go to Serge Gratton (CERFACS and CNES) for interesting discussions on quasi-Newton optimization methods. The LU factorization of the impedance matrix was performed with MUMPS (http://graal.ens-lyon.fr/MUMPS/index.html). The mesh generation was performed with the help of Triangle (http://www.cs.cmu.edu/~quake/triangle.html). The Garvin problem analytical solutions were computed with code provided by U. Iturrarán and F. J. Sánchez-Sesma. Access to the high-performance computing facilities of the Mesocentre SIGAMM computer center, the IDRIS national center (project 082280) and the GEOAZUR computing center, provided the required computer resources. Many thanks go to J. Kommendal and L. Sirgue from BP, for providing the elastic synthetic models of Valhall. I would like to thank the two anonymous reviewers for their fruitful comments.

References

Amestoy, P.R., Guermouche, A., L'Excellent, J.Y., Pralet, S., 2006. Hybrid scheduling for the parallel solution of linear systems. Parallel Computing 32, 136–156.

Berenger, J.-P., 1994. A perfectly matched layer for absorption of electromagnetic waves. Journal of Computational Physics 114, 185–200.

Bielak, J., Loukakis, K., Hisada, Y., Yoshimura, C., 2003. Domain reduction method for three-dimensional earthquake modeling in localized regions, part I: theory. Bulletin of the Seismological Society of America 93 (2), 817–824.

Bohlen, T., Saenger, E.H., 2006. Accuracy of heterogeneous staggered-grid finite-difference modeling of Rayleigh waves. Geophysics 71, 109–115.

Brenders, A., Pratt, R., Charles, S., 2009. Waveform tomography of 2-d seismic data in the canadian foothills—data preconditioning by exponential time-damping. In: Expanded Abstracts. EAGE, p. U041.

Brenders, A.J., Pratt, R.G., 2007a. Efficient waveform tomography for lithospheric imaging: implications for realistic 2D acquisition geometries and low frequency data. Geophysical Journal International 168, 152–170.
Brenders, A.J., Pratt, R.G., 2007b. Full waveform tomography for lithospheric imaging: results from a blind test in a realistic crustal model. Geophysical Journal International 168, 133–151.
Brossier, R., Operto, S., Virieux, J., 2009a. Seismic imaging of complex onshore structures by 2D elastic frequency-domain full-waveform inversion. Geophysics 74 (6), WCC63–WCC76.
Brossier, R., Operto, S., Virieux, J., 2009b. Two-dimensional seismic imaging of the Valhall model from synthetic OBC data by frequency-domain elastic full-waveform inversion. SEG Technical Program Expanded Abstracts, Houston 28 (1), 2293–2297.
Brossier, R., Virieux, J., Operto, S., 2008. Parsimonious finite-volume frequency-domain method for 2-D P-SV-wave modelling. Geophysical Journal International 175 (2), 541–559.
Bunks, C., Salek, F.M., Zaleski, S., Chavent, G., 1995. Multiscale seismic waveform inversion. Geophysics 60 (5), 1457–1473.
Chaljub, E., Capdeville, Y., Vilotte, J., 2003. Solving elastodynamics in a fluid-solid heterogeneous sphere: a parallel spectral element approximation on non-conforming grids. Journal of Computational Physics 187, 457–491.
Duff, I.S., Reid, J.K., 1983. The multifrontal solution of indefinite sparse symmetric linear systems. ACM Transactions on Mathematical Software 9, 302–325.
Dumbser, M., Käser, M., 2006. An arbitrary high order discontinuous Galerkin method for elastic waves on unstructured meshes II: the three-dimensional isotropic case. Geophysical Journal International 167 (1), 319–336.
Epanomeritakis, I., Akçelik, V., Ghattas, O., Bielak, J., 2008. A Newton-CG method for large-scale three-dimensional elastic full waveform seismic inversion. Inverse Problems 24, 1–26.
Etienne, V., Chaljub, E., Virieux, J., Glinsky, N., 2010. An hp-adaptive discontinuous Galerkin finite-element method for 3D elastic wave modelling. Geophysical Journal International 183 (2), 941–962.
Futterman, W., 1962. Dispersive body waves. Journal of Geophysics Research 67, 5279–5291.
Garvin, W.W., 1956. Exact transient solution of the buried line source problem. Proceedings of Royal Society of London 234, 528–541.
Hesthaven, J.S., Warburton, T., 2008. Nodal Discontinuous Galerkin Method. Algorithms, Analysis, and Application. Springer, New York.
Hustedt, B., Operto, S., Virieux, J., 2004. Mixed-grid and staggered-grid finite difference methods for frequency domain acoustic wave modelling. Geophysical Journal International 157, 1269–1296.
Jo, C.H., Shin, C., Suh, J.H., 1996. An optimal 9-point, finite-difference, frequency-space 2D scalar extrapolator. Geophysics 61, 529–537.
Karypis, G., Kumar, V., 1999. A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM Journal on Scientific Computing 20 (1), 359–392.
Käser, M., Dumbser, M., 2006. An arbitrary high order discontinuous Galerkin method for elastic waves on unstructured meshes I: the two-dimensional isotropic case with external source terms. Geophysical Journal International 166, 855–877.
Kolsky, H., 1956. The propagation of stress pulses in viscoelastic solids. Philosophical Magazine 1, 693–710.
Komatitsch, D., Vilotte, J.P., 1998. The spectral element method: an efficient tool to simulate the seismic response of 2D and 3D geological structures. Bulletin Seismology Society of America 88, 368–392.
Luo, Y., Schuster, G.T., 1990. Parsimonious staggered grid finite-differencing of the wave equation. Geophysical Research Letters 17 (2), 155–158.
Marfurt, K., 1984. Accuracy of finite-difference and finite-elements modeling of the scalar and elastic wave equation. Geophysics 49, 533–549.
Moczo, P., Bystricky, E., Carcione, J.M., Bouchon, M., 1997. Hybrid modeling of P-SV seismic motion at inhomogeneous viscoelastic topographic structures. Bulletin Seismology Society of America 87, 1305–1323.
Mulder, W.A., Hak, B., 2009. Simultaneous imaging of velocity and attenuation perturbations from seismic data is nearly impossible. In: Expanded Abstracts. p. S043.
Munns, J.W., 1985. The Valhall field: a geological overview. Marine and Petroleum Geology 2, 23–43.
Nocedal, J., 1980. Updating quasi-Newton matrices with limited storage. Mathematics of Computation 35 (151), 773–782.
Nocedal, J., Wright, S.J., 1999. Numerical Optimization. Springer, New York, US.
Operto, S., Virieux, J., Dessa, J.X., Pascal, G., 2006. Crustal imaging from multifold ocean bottom seismometers data by frequency-domain full-waveform tomography: application to the eastern Nankai trough. Journal of Geophysical Research 111 (B09306). doi:10.1029/2005JB003835.
Plessix, R.-E., 2006. A review of the adjoint-state method for computing the gradient of a functional with geophysical applications. Geophysical Journal International 167 (2), 495–503.
Pratt, R.G., 1990. Inverse theory applied to multi-source cross-hole tomography. Part II: elastic wave-equation method. Geophysical Prospecting 38, 311–330.
Pratt, R.G., Shin, C., Hicks, G.J., 1998. Gauss–Newton and full Newton methods in frequency-space seismic waveform inversion. Geophysical Journal International 133, 341–362.
Pratt, R.G., Worthington, M.H., 1990. Inverse theory applied to multi-source cross-hole tomography. Part I: acoustic wave-equation method. Geophysical Prospecting 38, 287–310.
Ravaut, C., Operto, S., Improta, L., Virieux, J., Herrero, A., dell'Aversana, P., 2004. Multi-scale imaging of complex structures from multi-fold wide-aperture seismic data by frequency-domain full-wavefield inversions: application to a thrust belt. Geophysical Journal International 159, 1032–1056.
Remaki, M., 2000. A new finite volume scheme for solving Maxwell's system. COMPEL 19 (3), 913–931.
Saenger, E.H., Bohlen, T., 2004. Finite-difference modelling of viscoelastic and anisotropic wave propagation using the rotated staggered grid. Geophysics 69, 583–591.
Sambridge, M.S., Tarantola, A., Kennett, B.L., 1991. An alternative strategy for non-linear inversion of seismic waveforms. Geophysical Prospecting 39, 723–736.
Sears, T., Singh, S., Barton, P., 2008. Elastic full waveform inversion of multi-component OBC seismic data. Geophysical Prospecting 56 (6), 843–862.
Shewchuk, J.R., May 1996. Triangle: engineering a 2D quality mesh generator and delaunay triangulator. In: Lin, M.C., Manocha, D. (Eds.), Applied Computational Geometry: Towards Geometric Engineering. Lecture Notes in Computer Science, vol. 1148. Springer-Verlag, pp. 203–222 from the First ACM Workshop on Applied Computational Geometry.
Shin, C., Jang, S., Min, D.J., 2001. Improved amplitude preservation for prestack depth migration by inverse scattering theory. Geophysical Prospecting 49, 592–606.
Shin, C., Min, D.-J., 2006. Waveform inversion using a logarithmic wavefield. Geophysics 71 (3), R31–R42.
Shin, C., Min, D.-J., Marfurt, K.J., Lim, H.Y., Yang, D., Cha, Y., Ko, S., Yoon, K., Ha, T., Hong, S., 2002. Traveltime and amplitude calculations using the damped wave solution. Geophysics 67, 1637–1647.
Sirgue, L., Pratt, R.G., 2004. Efficient waveform inversion and imaging: a strategy for selecting temporal frequencies. Geophysics 69 (1), 231–248.
Sourbier, F., Operto, S., Virieux, J., Amestoy, P., L'Excellent, J.-Y., 2009a. Fwt2d: a massively parallel program for frequency-domain full-waveform tomography of wide-aperture seismic data-part 1: algorithm. Computers & Geosciences 35 (3), 487–495.
Sourbier, F., Operto, S., Virieux, J., Amestoy, P., L'Excellent, J.-Y., 2009b. Fwt2d: a massively parallel program for frequency-domain full-waveform tomography of wide-aperture seismic data-part 2: numerical examples and scalability analysis. Computers & Geosciences 35 (3), 496–514.
Stekl, I., Pratt, R.G., 1998. Accurate viscoelastic modeling by frequency-domain finite difference using rotated operators. Geophysics 63, 1779–1794.
Symes, W.W., 2008. Migration velocity analysis and waveform inversion. Geophysical Prospecting 56, 765–790.
Tarantola, A., 1984. Inversion of seismic reflection data in the acoustic approximation. Geophysics 49 (8), 1259–1266.
Tarantola, A., 1987. Inverse Problem Theory: Methods for Data Fitting and Model Parameter Estimation. Elsevier, New York.
Toksöz, M.N., Johnston, D.H., 1981. Seismic Wave Attenuation. Geophysics reprint series, No. 2. Society of Exploration Geophysicists, Tulsa, OK.
Virieux, J., Operto, S., 2009. An overview of full waveform inversion in exploration geophysics. Geophysics 74 (6), WCC127–WCC152.