Geophysical Journal International

Geophys. J. Int. (2016) **206,** 1864–1889
Advance Access publication 2016 June 17
GJI Seismology

doi: 10.1093/gji/ggw202

Seismic waveform inversion best practices: regional, global and exploration test cases

Ryan Modrak[1] and Jeroen Tromp[1,2]

[1]*Department of Geosciences, Princeton University, Princeton, NJ* 08544*, USA. E-mail: rmodrak@princeton.edu*
[2]*Program in Applied & Computational Mathematics, Princeton University, Princeton, NJ* 08544-1000*, USA*

Accepted 2016 May 25. Received 2016 May 8; in original form 2015 December 3

SUMMARY

Reaching the global minimum of a waveform misfit function requires careful choices about the nonlinear optimization, preconditioning and regularization methods underlying an inversion. Because waveform inversion problems are susceptible to erratic convergence associated with strong nonlinearity, one or two test cases are not enough to reliably inform such decisions. We identify best practices, instead, using four seismic near-surface problems, one regional problem and two global problems. To make meaningful quantitative comparisons between methods, we carry out hundreds of inversions, varying one aspect of the implementation at a time. Comparing nonlinear optimization algorithms, we find that limited-memory BFGS provides computational savings over nonlinear conjugate gradient methods in a wide range of test cases. Comparing preconditioners, we show that a new diagonal scaling derived from the adjoint of the forward operator provides better performance than two conventional preconditioning schemes. Comparing regularization strategies, we find that projection, convolution, Tikhonov regularization and total variation regularization are effective in different contexts. Besides questions of one strategy or another, reliability and efficiency in waveform inversion depend on close numerical attention and care. Implementation details involving the line search and restart conditions have a strong effect on computational cost, regardless of the chosen nonlinear optimization algorithm.

Key words: Inverse theory; Numerical approximations and analysis; Computational seismology.

1 INTRODUCTION

Waveform inversion practitioners must choose from a variety of objective functions, nonlinear optimization algorithms, preconditioning strategies, regularization methods and multiscale schemes. Though an extensive applied mathematics literature exists on these topics, much of it is based on numerical benchmarks that are less challenging and computationally expensive than commonly encountered in geophysics. In the waveform inversion literature itself, methodological comparisons sometimes lack implementation details, involve only one or two test cases, or use starting models quite close to the global minimum of the objective function.

To address such issues, we provide systematic comparisons between inversion strategies through six acoustic inversion test cases. While a comparison of objective functions and multiscale procedures would fit naturally into this framework, we choose to focus instead on numerical aspects of inversion—nonlinear optimization, preconditioning and regularization—that have received somewhat less attention in the geophysical literature.

Robustness and efficiency in waveform inversion depend in large part on numerical decisions. Because of their importance, such choices ought to be informed by both practical waveform inversion experience and numerical theory and results. Benchmark comparisons by Nash & Nocedal (1991) and Zou *et al.* (1993) and review papers by Nocedal (1992), Gould *et al.* (2005) and Burstedde & Ghattas (2009) supply a useful foothold into the numerical literature. The emerging field of PDE-constrained optimization (Biegler *et al.* 2003)—involving parameter estimation, optimal design and optimal control systems governed by partial differential equations—offers additional relevant experience. Through wide-ranging references, we seek to ground the waveform inversion results below in this literature.

This paper is organized as follows. Sections 2 and 3 provide a description of test cases and testing procedures. Sections 4 and 5 present the results of a comparison of nonlinear optimization algorithms, focusing first on the search direction and then on the line search. Sections 6 and 7 discuss preconditioning and regularization. A number of other issues that do not fit well into any of the previous

© The Authors 2016. Published by Oxford University Press on behalf of The Royal Astronomical Society.

Downloaded from https://academic.oup.com/gji/article/206/3/1864/2583505 by Princeton University user on 11 September 2020

categories are covered in Section 8. Finally, in Section 9, we conclude with a review of seismic waveform inversion 'best practices'.

2 TEST CASES

We present convergence results from the following test cases: (a) Marmousi, (b) overthrust, (c) salt, (d) anticline, (e) global, (f) regional and (g) deep Earth. Through target models representing horizontal or vertical cross-sections, each of these 2-D problems provides a window into an associated 3-D problem.

Problems a–d correspond to widely used exploration geophysics test cases. True models, shown in Fig. 1, include various thrust fault, normal fault and salt structures. Smooth starting models were obtained by convolving true models with Gaussian kernels. Inversions were carried out in the acoustic approximation, that is, using acoustic models and data to approximate the elastic subsurface. For the overthrust and Marmousi test cases, we considered both onshore and offshore variants to give a sense for how performance differs between these two cases.

Problems e–g investigate seismic inversion at much larger scales. Wavefield simulations are once again based on the acoustic wave equation, with an analogy to horizontal surface wave propagation for the regional and global test cases and vertical compressional wave propagation for the deep Earth test case. For the regional and global problems, starting models were homogeneous, and for the deep Earth problem the starting model was a radial reference model, AK135. The true model for the global test case was based on 40 s Rayleigh wave phase speeds from Trampert & Woodhouse (2003). The true model for the regional test case was based on 10 s Rayleigh wave phase speeds from Ekström *et al.* (2009). Finally, the true model for the deep Earth test case was obtained by superimposing Gaussian random variations on AK135.

For the global test problem, periodic boundary conditions were used at the sides of a rectangular mesh to roughly approximate the spherical Earth. For the deep Earth test problems, the inner and outer core were included in wavefield simulations, but excluded from model updates.

For the near-surface problems a–d, data from 32 shots were simulated at 500 hydrophones. Shots and hydrophones were placed at 10 m depth in a 500 m water layer. Multiple reflections were excluded from both data and synthetics.

For the regional and global problems e–g, sources and receivers were chosen to mimic the actual distribution of earthquakes and seismic stations on Earth's surface. For the global test case, sources were constrained to plate boundaries and receivers to dry land. For the deep Earth test case, stations were constrained to Earth's surface and earthquakes to <300 km depth.

3 TESTING PROCEDURES

Inversions were performed in the time domain using a waveform difference objective function,

$$\chi(m) = \frac{1}{2} \sum \int |s(m,t) - d(t)|^2 \mathrm{d}t, \tag{1}$$

where m is the model, s are synthetics, d are observations, and the sum is over all sources and receivers. No muting or windowing of traces was performed in any of the inversions.

In describing algorithms, we sometimes write the model, gradient and Hessian as scalar functions of spatial position. Generalization to the multiparameter case, we note, is usually just a matter of introducing a sum over material properties. Because forward modelling dominates computational expense, cost comparisons are made in terms of wavefield simulations. In displaying convergence results, we plot $L2$ model error $\int |m - m_{\mathrm{true}}|^2 \mathrm{d}V$, where the integral is over spatial position, versus the cumulative number of wavefield simulations. We recall that each model update requires at least two sets of wavefields simulations, one for the gradient and one for the line search. The gradient evaluation itself, strictly speaking, requires two sets of wavefield simulations, but the second contributes no new cost to the inversion since it is carried over from the previous line search.

For forward and adjoint simulations, we used the spectral element solver `SPECFEM2D` (Komatitsch & Vilotte 1998), which employs an explicit time stepping scheme and an 'optimize-then-discretize' approach to the adjoint operator (Gunzburger 2000). For nonlinear optimization, data pre-processing, gradient postprocessing and workflow integration tasks, we used the `SeisFlows` framework. Both are open source packages available through GitHub (http://github.com/geodynamics/specfem2d, http://github.com/PrincetonUniversity/seisflows).

4 NONLINEAR OPTIMIZATION ALGORITHMS

The rate of convergence in waveform inversion depends on the nonlinear optimization algorithm used to iteratively update the model. The work of a model update, conventionally, is divided into two steps. First, a search direction is computed based on the gradient of the objective function. Second, a step length is determined along the search direction through a line search procedure. In this section, we compare two widely used search direction algorithms, leaving detailed discussion of the line search until Section 5.

4.1 Limited-memory BFGS algorithm

The L-BFGS algorithm (Liu & Nocedal 1989) is a quasi-Newton method, which means that search directions are based on a low-dimensional quadratic model of the objective function. After several decades of experience with such methods, L-BFGS is generally regarded as the most effective quasi-Newton method (Nocedal 1992; Kolda *et al.* 1998).

Over the course of an inversion, the quadratic model of the objective function formed by L-BFGS varies through an updating process, with each new gradient evaluation providing more information. L-BFGS is limited-memory in the sense that results from only the most recent gradient evaluations need to be stored. L-BFGS search directions are well-scaled in the sense that they terminate at the vertex of the paraboloid used to locally represent the objective function. For reference, a concise statement of the L-BFGS algorithm is given in Appendix A.

To specify the number of gradient evaluations kept track of by L-BFGS, users must choose a memory value. For waveform inversion problems, we find that values between three and seven work well. Generally, it seems that problems with high nonlinearity benefit from lower memory values, and problems with low nonlinearity benefit from higher memory values, though differences are often quite minor. Supporting results are provided in the online supplement.

4.2 Nonlinear conjugate gradient method

The nonlinear conjugate gradient (NLCG) method returns a search direction that is a linear combination of the gradient and the

Downloaded from https://academic.oup.com/gji/article/206/3/1864/2583505 by Princeton University user on 11 September 2020

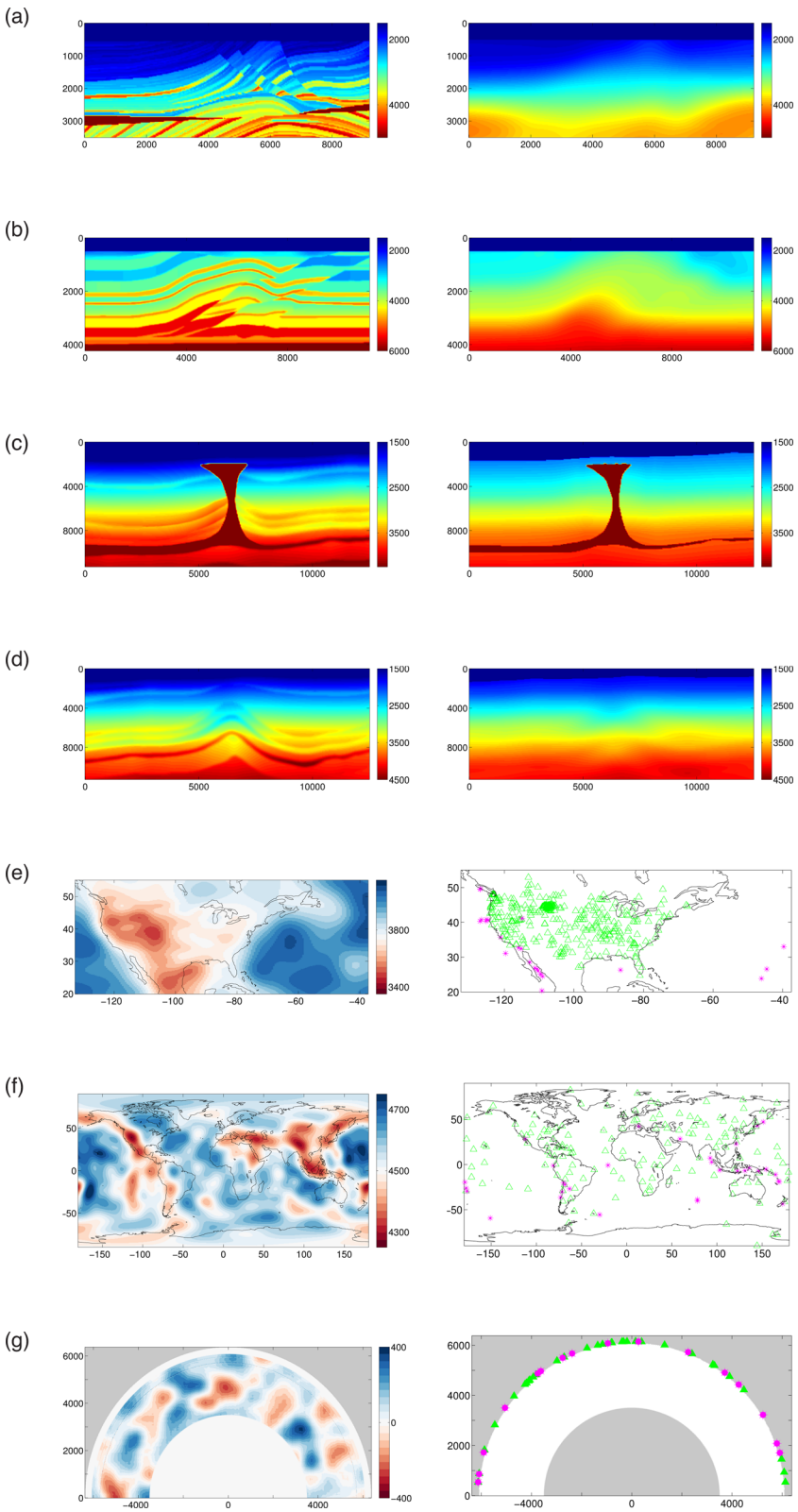

Figure 1. Waveform inversion test cases (a) Marmousi, (b) overthrust, (c) salt, (d) anticline, (e) regional, (f) global and (g) deep Earth. For the exploration test cases (a–d), target models are shown on the left and starting models on the right. For the regional and global test cases (e–g), target models are shown on the left and sources (magenta) and receivers (green) on the right. Because they are homogeneous and/or radially symmetric, starting models for the regional and global test cases are not shown.

Downloaded from https://academic.oup.com/gji/article/206/3/1864/2583505 by Princeton University user on 11 September 2020

previous search direction. Among several NLCG variants, those due to Polak & Ribière (1969) and Gilbert & Nocedal (1992) have proven particularly effective. Because NLCG does not involve a quadratic model of the objective function, the length of the search direction is not especially meaningful and, hence, more effort must be expended on the line search. For reference, a concise statement of the NLCG algorithm is given in Appendix B.

Although NLCG and L-BFGS share certain theoretical underpinnings (Nazareth 1979), the two algorithms provide quite different user experiences. While the L-BFGS search direction computation is more complicated than the NLCG search direction computation, L-BFGS may be easier to implement overall, perhaps, on account of its simpler initial step length selection, line search and restart procedures. Importantly, both algorithms can be combined with stochastic inversion strategies for additional savings (van Leeuwen *et al.* 2011; van Leeuwen & Herrmann 2013; Castellanos *et al.* 2015).

As an aside, we note that linear conjugate gradient methods and NLCG methods differ in that while the former are used to solve systems of linear equations, the latter are used to solve non-quadratic optimization problems. Here we consider only NLCG methods, noting in passing that the use of linear conjugate gradient methods to solve a series of linear subproblems forms the basis for another nonlinear optimization algorithm, the truncated Newton method (Nash 2000), which comprises an active research area (Burstedde & Ghattas 2009; Métivier *et al.* 2014).

4.3 Comparisons

L-BFGS has been shown to provide computational savings over NLCG and other competitors in a number of classic studies (Liu & Nocedal 1989; Nash & Nocedal 1991; Zou *et al.* 1993; Kolda *et al.* 1998). Many of these early comparisons involved inexpensive, low-dimensional optimization problems from a list compiled by Moré *et al.* (1981). As Gould *et al.* (2005) point out, more recent benchmarks are in short supply.

There are significant differences between early nonlinear optimization test cases and waveform inversion problems not only in terms of computational expense and model space dimensionality, but also in terms of nonlinearity and nonconvexity. A set of updated performance comparisons might therefore be useful. To provide one such benchmark, we compared the efficiency of L-BFGS and NLCG in experiments with the waveform inversion examples described above. Out of curiosity, the steepest descent algorithm was also included in the comparisons. To allow straightforward comparisons between test problems, we used the 'regularization by convolution' method described in Section 6, Polak-Ribière NLCG search directions and an L-BFGS memory value of five. The results of these experiments, shown in 2, show L-BFGS as the clear winner, with computational savings of 30 to 50 per cent over NLCG.

These findings provide the starting point for much further analysis. In Section 4, we describe how a backtracking line search procedure contributes to the efficiency of L-BFGS. In Section 5, we demonstrate additional computational savings through preconditioning. Finally, in Section 6, we show how regularization can help solve convergence problems evident in the nonlinear optimization comparisons.

Having used the waveform inversion test cases to compare optimization algorithms, we can, looking at things the other way around, use the results to infer how nonlinearity varies from one test case to another. The most obvious differences occur between near-surface problems and regional and global problems. While the convergence rate in the near-surface inversions is often slow and erratic, the regional and global inversions settle quickly into superlinear convergence. Among the near-surface problems themselves there is considerable variation, with the overthrust models, which generate strong diving waves and weak reflections, converging fastest, and the Marmousi models, which generate weak diving waves and strong reflections, converging slowest.

4.4 Numerical issues

Nonlinearity in waveform inversion can cause significant problems for optimization algorithms. Drawing on the waveform inversion test cases, we describe two numerical problems that can usually be resolved by restarting, that is, by discarding the algorithm's accumulated state and continuing as if no prior gradient evaluations were available. More details about restart conditions are given in Appendix C.

4.4.1 Lack of a descent direction

The most basic requirement of a search direction it that it provides a reduction in the objective function. If p is the search direction and g is the gradient, then p must satisfy

$$p^T g < 0 \tag{2}$$

in order to provide such a decrease. Explicitly checking this condition adds virtually no cost since it requires only a vector product.

In waveform inversion, occasional failure of the optimization algorithm to provide a descent direction is not unexpected. Out of some 1800 model updates carried out for Fig. 2, L-BFGS and NLCG required restarts about one per cent of the time. The restart rate can be much higher, we find, in applications involving multiparameter inversion, noisy data, or stochastic optimization, though we do not include any such test cases here.

4.4.2 Loss of conjugacy of NLCG search directions

On highly nonlinear problems, NLCG search directions gradually lose the property of conjugacy, or orthogonality with respect to an inner product involving the Hessian, on which good performance of the method depends. Fast convergence can usually be regained by restarting the algorithm, as described for example by Powell (1977). In our experience, such safeguards are occasionally necessary in waveform inversion to avoid stagnation.

5 LINE SEARCH ALGORITHMS

Given a model m and search direction p, the work of the line search is to find a step length α such that the updated model $m + \alpha p$ satisfies the decrease and curvature conditions described in Appendix D. In Section 4, we compared search direction algorithms using the type of line search appropriate for each one: for NLCG we used a bracketing line search, and for L-BFGS we used a safeguarded backtracking line search. We now give an idea of the issues involved with both search procedures, and through numerical experiments examine their contribution to the overall cost of an inversion.

Downloaded from https://academic.oup.com/gji/article/206/3/1864/2583505 by Princeton University user on 11 September 2020

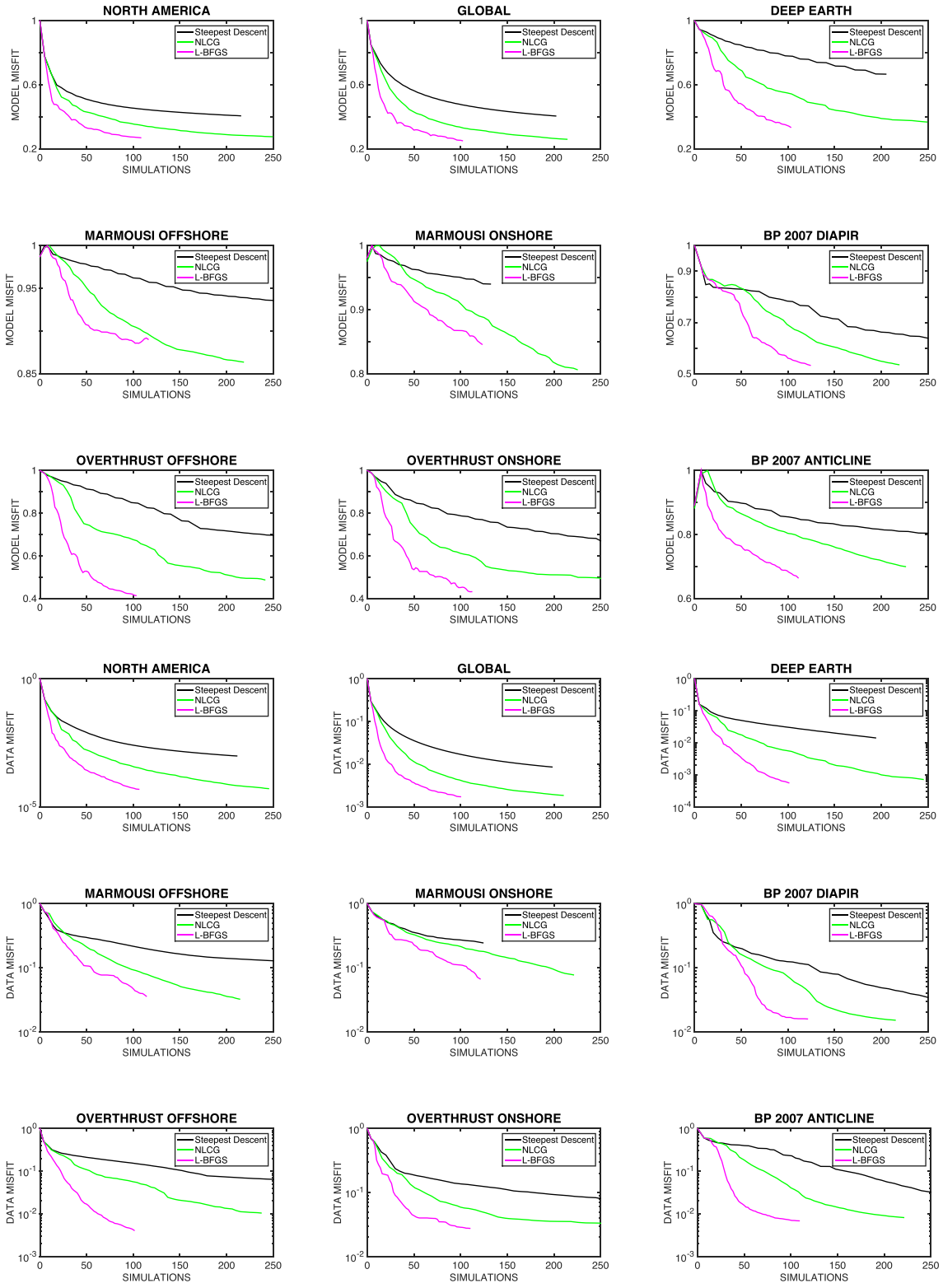

Figure 2. Comparison of nonlinear optimization algorithms. L-BFGS provides significant computational savings over NLCG in the waveform inversion test cases.

5.1 Bracketing line search

Acceptable NLCG step lengths can vary by several orders of magnitude from one model update to another. During the line search, a good strategy to deal with this lack of scaling is to first bracket the minimum of the objective function along the search direction and then choose a step length by polynomial interpolation between bracketing points.

Since the bracketing procedure can add considerably to the cost of an inversion, it must be carried out efficiently. To avoid unnecessary

Downloaded from https://academic.oup.com/gji/article/206/3/1864/2583505 by Princeton University user on 11 September 2020

gradient evaluations, we check the curvature condition only after the minimum has been bracketed and a polynomial interpolation has been performed. After this, if a given step length is found to satisfy both descent and curvature conditions, the associated gradient evaluation can be carried over to the next model update iteration, removing the need for any additional evaluations until the next line search.

5.2 Safeguarded backtracking line search

The L-BFGS algorithm with proper initialization returns a search direction that is well-scaled in the sense that a unit step length $\alpha = 1$ is an appropriate first choice. While most of the time a unit step satisfies the decrease condition mentioned above, occasionally one or more subsequent trial steps are required. The idea of a backtracking line search is to select any subsequent trial steps by interpolating backward, towards zero, within the unit interval. Because the search direction has to satisfy eq. (2), a reduction in the objective function relative to $\alpha = 0$ can always be found in the unit interval. Our use of the term 'safeguarded' relates to the fact that if the backtracking procedure fails to return a step length satisfying the curvature condition, we terminate the backtracking line search and switch to a bracketing line search.

For choosing backtracking step lengths, we use the quadratic and cubic interpolation algorithms given by Nocedal & Wright (2006). To ensure that the interpolation procedure does not select a step length too close to zero on the one hand or too close to the old step length on the other, we impose upper and lower bounds of the type described by Dennis & Schnabel (1996). Since well-scaled L-BFGS search directions are available only after at least two gradient evaluations have been performed, we switch from a bracketing line search to a backtracking line search starting with the second model update iteration.

5.3 Comparisons

Fig. 3 shows results from numerical experiments involving the line search. In the waveform inversion test cases, a typical bracketing line search is found to require 3.5 function evaluations, and a typical backtracking line search is found to require 1.2 function evaluations.

Importantly, in waveform inversion and other optimization procedures based on adjoint methods, the last function evaluation of the line search overlaps with the first function evaluation of the next model update iteration. Put another way, a forward simulation performed during the line search removes the need for a forward simulation as a prerequisite for the next adjoint simulation. As a result, a backtracking line search contributes little to the overall cost of an inversion, only about 0.2 function evaluations on average. At about 2.5 function evaluations, the effective cost of a bracketing line search in the waveform inversion is significantly higher. In the numerical optimization literature, Nash & Nocedal (1991) reported a similar cost per L-BFGS backtracking line search. Published cost estimates for bracketing line searches vary more widely, reflecting the greater diversity of algorithms in use. Liu & Nocedal (1989) report an average cost of around 2.5 function evaluations per NLCG bracketing line search, less expensive than in the waveform test problems. We note that Fig. 3 of this paper presents essentially the same method comparisons as table 13 of Liu and Nocedal, though with expensive, high-dimensional waveform inversion problems considered in this study and comparatively inexpensive, low-dimensional test problems considered in the other.

6 PRECONDITIONING

The performance of the nonlinear optimization algorithms discussed above can be improved through preconditioning. In this section, we begin with a general overview and move on to descriptions and numerical comparisons of waveform inversion diagonal preconditioners.

6.1 Overview

Preconditioning is a way of rescaling or recombining model parameters to provide favourable numerical properties. Despite upfront computational and storage costs, such a procedure can provide significant overall savings by accelerating the convergence of the optimization algorithm.

Underlying most preconditioning methods is a change of variables via a linear transformation, say $\hat{m} = Cm$. In preconditioning conjugate gradient methods, the change of variables C does not enter the computations directly, except through the action of $P = C^T C$ or its inverse. P is typically called the preconditioner, even though its inverse is usually what is implemented in practice. Importantly, a good trade-off between inexpensive computation of P and fast convergence of the optimization algorithm is required for the preconditioner to provide an overall reduction in computational cost.

Typically, the term 'preconditioning' is used in connection with conjugate gradient methods and 'rescaling' in connection with quasi-Newton algorithms.

Most preconditioning strategies involve direct or indirect connections to the Hessian. Even for computationally demanding problems, the Hessian can be made to play a useful role through numerical approximations. Given the huge dimensionality of the model space in waveform inversion, it is important that such approximations allow for inexpensive computation and affordable storage. Examples of this kind of approach include the following.

(1) Quasi-Newton preconditioners, in which the approximation to the Hessian varies from one model update to another through an updating process.

(2) Higher-dimensional but still relatively inexpensive preconditioners involving, for example, forward simulations with coarse numerical grids or approximate solvers.

(3) Diagonal preconditioners formed by exact or inexact computation of the diagonal elements of the Hessian.

While the first two categories have received some attention (Akçelik *et al.* 2003; Demanet *et al.* 2011; Métivier *et al.* 2014), most preconditioners in waveform inversion fall into the last category (Claerbout & Nichols 1994; Shin *et al.* 2001; Rickett 2003). In suggesting best practices below, we focus on diagonal preconditioners not because they are always the most cost effective, but because they are widely used, easily implemented, and can serve as a starting point for more sophisticated techniques.

6.2 Waveform inversion preconditioners

Two types of diagonal preconditioners are prevalent in waveform inversion: scalings that account geometric spreading away from the

Downloaded from https://academic.oup.com/gji/article/206/3/1864/2583505 by Princeton University user on 11 September 2020

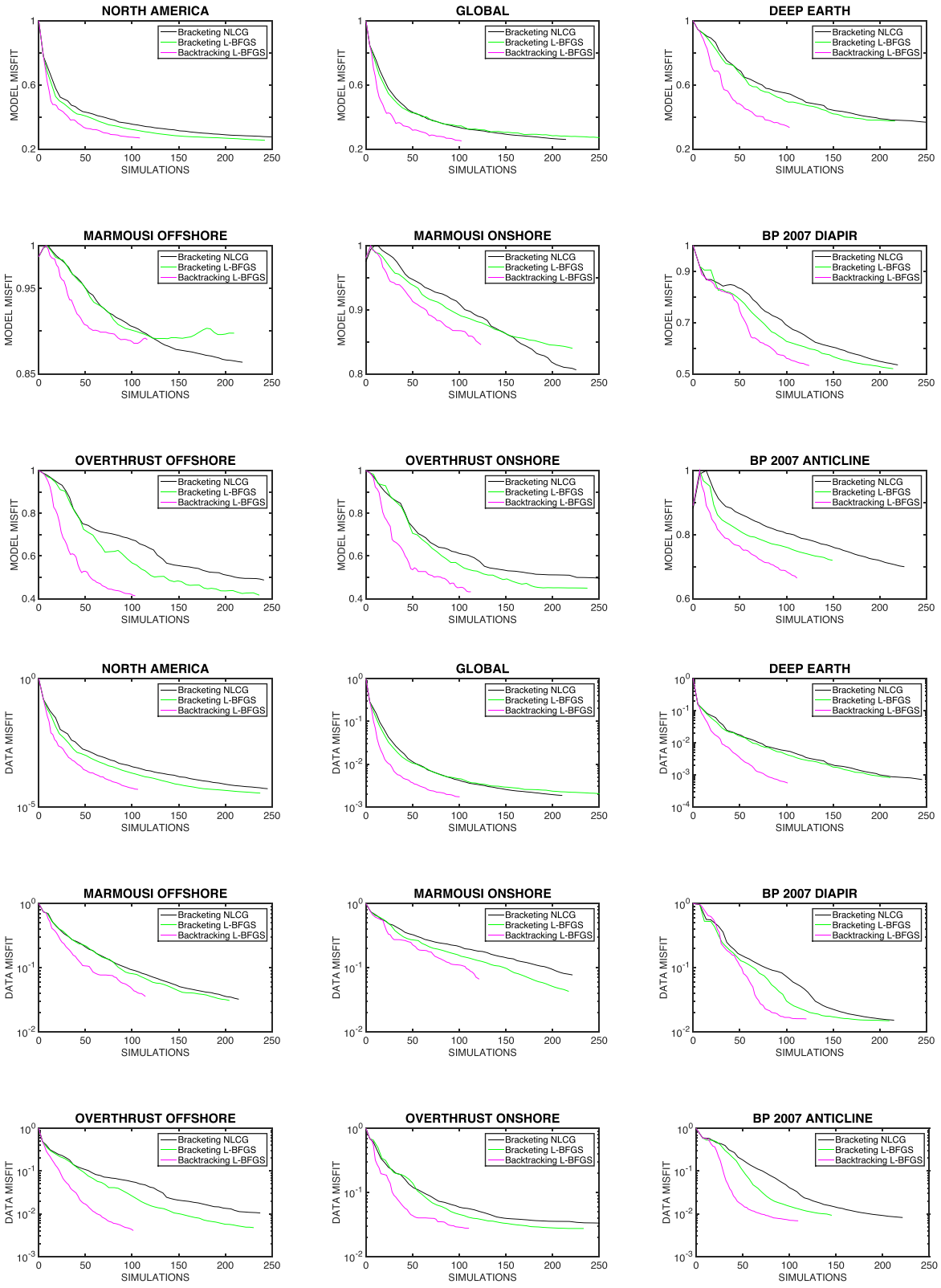

Figure 3. Comparison of line search algorithms. A safeguarded backtracking line search contributes significantly to the overall efficiency of L-BFGS. Because NLCG search directions are not well-scaled, a backtracking line search is not effective with NLCG, and a more expensive bracketing line search is required instead.

Downloaded from https://academic.oup.com/gji/article/206/3/1864/2583505 by Princeton University user on 11 September 2020

sources, and scalings obtained by applying the adjoint of the forward solver to the data (Rickett 2003). Using perturbation analysis, Luo (2012) showed that both types of preconditioners are related, but not exactly equivalent, to second order variations of a waveform-difference misfit function. We now briefly restate Luo's results.

By expanding the displacement field, u, as a function of the model, m, in a perturbation series

$$u(m+\delta m) \approx u(m) + \delta u_1(m) + \delta u_2(m), \tag{3}$$

the variation in waveform-difference misfit can be written as

$$\delta\chi \approx \delta\chi_0 + \delta\chi_1 + \delta\chi_2, \tag{4}$$

where $\delta\chi_1$ and $\delta\chi_2$ correspond to first- and second-order scattering terms δu_1 and δu_2. If H_1 is the positive semi-definite first-order contribution to the waveform-difference Hessian and H_2 is the remaining second-order contribution, the gradient g and Hessian $H = H_1 + H_2$ are related to the variation of the data misfit via

$$\delta\chi_0 = \int g(x)\,\delta m(x)\,\mathrm{d}V, \tag{5}$$

$$\delta\chi_1 = \frac{1}{2}\iint \delta m(x)\,H_1(x,x')\,\delta m(x')\,\mathrm{d}V\,\mathrm{d}V', \tag{6}$$

$$\delta\chi_2 = \frac{1}{2}\iint \delta m(x)\,H_2(x,x')\,\delta m(x')\,\mathrm{d}V\,\mathrm{d}V'. \tag{7}$$

By taking

$$\lim_{x\to x'} H(x,x') \tag{8}$$

various diagonal scalings can be derived. Referring to Luo (2012) for details, we state the main result of the perturbation analysis, namely, that diagonal preconditioners

$$P_1(x) = \sum_{i=1}^{N_s}\int \partial_t^2\,u(x,t)\,\partial_t^2\,u(x,t)\,\mathrm{d}t \tag{9}$$

$$P_2(x) = \sum_{i=1}^{N_s}\int \partial_t^2\,u(x,t)\,\partial_t^2 v(x,-t)\,\mathrm{d}t \tag{10}$$

are related to the data misfit variations (6) and (7), respectively, through the limit (8). If G is the Green's function of the medium, then in the above expressions

$$u(x,t) = \int G(x,s,t-t')\,f(t')\,\mathrm{d}t' \tag{11}$$

is the wavefield originating from the source located at s with wavelet $f(t)$, and

$$v(x,t) = \sum_{j=1}^{N_r}\int G(x,r_j,t-t')\,[d_j(t') - u(r_j,t')]\,\mathrm{d}t', \tag{12}$$

is the data residual wavefield that arises from backprojecting the differences between observed data $d_j(t)$ and simulated data $u(r_j, t)$ from the source located at s and the receivers located at $r_j, j = 1, \ldots, N_r$.

While both P_1 and P_2 contribute to the variation in data misfit through the diagonal of the Hessian, each behaves in a different way and it is useful to consider them separately. P_1 involves only the wavefield originating from the sources, so it does a better job than P_2 accounting for amplitude effects such a geometric spreading, focusing and defocusing. P_2 involves wavefields originating from both the sources and receivers, so it is more effective than P_1 in compensating for uneven data coverage. Because P_1 and P_2 correspond to the first- and second- order contributions the Hessian, respectively, it sensible to precondition using either P_1 or $P_1 + P_2$ but not P_2 alone. We show later that P_1 and $P_1 + P_2$ provide almost identical performance.

Given their direct relation to the waveform-difference Hessian, P_1 and P_2 can be viewed as variations on a theme developed by exploration geophysicists first from the perspective of migration, and later from the perspective of waveform inversion. In migration, Rickett (2003) compared the performance of a source-only diagonal scaling, similar to P_1, with diagonal scalings derived to the adjoint of the forward solver, similar to P_2.

Meanwhile, in regional and global seismology, the existence of well-behaved crust, mantle and core phases led to the adoption of wave-equation based phase and traveltime misfit functions (Dahlen *et al.* 2000; Tromp *et al.* 2005). Such methods combine the robustness of full waveform modelling with the reduced nonlinearity of phase and traveltime measurements compared with amplitude and waveform-difference measurements. A preconditioner that arises naturally in this context is

$$P_3(x) = \sum_{i=1}^{N_s}\int \partial_t^2 u(x,t)\,w(x,-t)\,\mathrm{d}t, \tag{13}$$

where

$$w(x,t) = \sum_{j=1}^{N_r}\int G(x,r_j,t-t')\,\partial_t u(r_j,t')\,\mathrm{d}t'. \tag{14}$$

In the terminology of Marquering *et al.* (1999), P_3 is simply the unweighted sum of 'banana-doughnut kernels' from all source–receiver pairs.

The similarity between this and the other two preconditioners suggests the possibility of using P_3 as an alternative diagonal approximation to the waveform-difference Hessian. As shown through the checkerboard example below, P_3 has properties that make it appear quite promising.

The main work of generating P_1, P_2 and P_3, we note, consists of propagating the forward wavefields u and the reverse wavefields v and w from eqs (9)–(13). For illustration, Fig. 4 shows these preconditioners computed using a checkerboard test case. To drive v, data residuals were obtained by subtracting traces generated from the checkerboard model shown in panel (a) with traces generated from a homogeneous model. The wavefields u, v and w themselves were propagated within the same homogeneous model.

In panels (b) and (c) of Fig. 4, we consider two source–receiver distributions. The first consists of a single source–receiver pair; the corresponding P_1 has a radially symmetric pattern, and P_2 and P_3 have alternating positive and negative fringes typical of wave-equation sensitivity kernels (Woodward 1992; Dahlen *et al.* 2000). The second source–receiver distribution consists of 25 sources and 132 receivers in a typical regional seismology layout; in this case, P_1 has a notably smaller condition number, or 'spread' of values than the other two preconditioners because it involves only wavefields originating from the sources.

Importantly, for the case of a single source–receiver pair, P_2 has more pronounced negative fringes than P_3. It follows that, for multiple source–receiver pairs, P_2 has a mix of positive and negative values and P_3 has mostly positive values. Given numerical problems

Downloaded from https://academic.oup.com/gji/article/206/3/1864/2583505 by Princeton University user on 11 September 2020

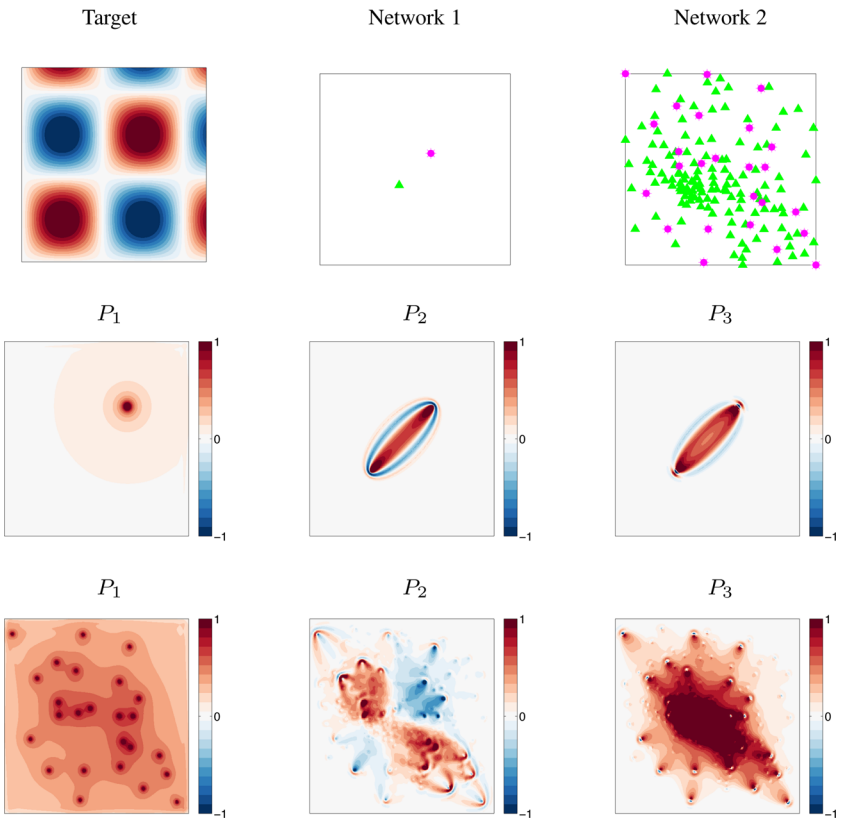

Figure 4. Illustration of waveform inversion diagonal preconditioners using a checkerboard example. Top: experimental setup. Middle: preconditioners corresponding to Network 1. Bottom: preconditioners corresponding to Network 2.

associated negative eigenvalue with distributions (Fletcher 1976), P_3 is expected to compare favourably to other preconditioners derived from the action of the adjoint of the forward solver.

6.3 Numerical issues

Even with diagonal approximations to the Hessian of the type illustrated in Fig. 4, a number of non-trivial implementation questions remain. Here we focus two issues, smoothing and updating, with significant practical effects.

6.3.1 Smoothing

Away from a minimum or maximum of the objective function, the Hessian may have both positive and negative eigenvalues. As Fig. 5 shows, the diagonal preconditioners $P_1 + P_2$ and P_3 may exhibit problematic eigenvalue distributions of this kind, which can cause numerical instability of the type described by Fletcher (1976). A simple and effective remedy, we find, is to replace negative values with zero and then smooth the resulting discontinuities.

Even if a diagonal preconditioner has no negative values, a large ratio of maximum to minimum values can result in slow or unstable convergence. While Rickett (2003) recommended damping to deal with such problems in migration, we recommend smoothing in waveform inversion. Use of a damped preconditioner $P + \lambda I$ works well in migration because it preserves the ability of the preconditioner to bring out fine details. In waveform inversion, it is desirable to bring out such details gradually to ensure that small-scale structures are not systematically mislocated as a result of errors in overlying large-scale structures. For this reason, smoothing the preconditioner, or some combination of smoothing and damping, works better than damping alone.

In practice, the amount of smoothing needed for good numerical performance is quite large. To investigate this issue quantitatively, we convolved preconditioners with Gaussian kernels with standard deviation $\sigma = \sigma_x = \sigma_z$ measured in terms of grid spacing h. Since grid spacing is related to dominant wavelength of the excited wavefield through a numerical condition, e.g., five grid points per wavelength, this way of looking at things is directly related to thinking about smoothing in terms of dominant wavelength. Comparing the performance of the resulting smoothed preconditioners in the waveform inversion test cases, we found that $\sigma = 80h$ provides the best results on average, with somewhat lower smoothing required in the regional and global test cases and somewhat higher smoothing required in the near-surface test cases. A great deal of supporting information is provided in the online supplement.

Fig. 7 gives a sense of the visual appearance of preconditioners after smoothing. For the near-surface test cases, preconditioners display both depth and lateral dependence, with shallow regions below the centre of the array weighted differently than deep regions below the edges of the array. For the regional and global test cases, the lateral variations are even more pronounced as a result of uneven earthquake and seismic station distributions.

Though we do not perform such experiments here, we note that the convolution procedure described above can easily be modified to allow dip-dependent smoothing for near-surface problems or radial smoothing for regional and global problems. In advocating smoothing, our motivation is purely numerical. Without it, preconditioning with the diagonal of the Hessian can result in slower than expected

Downloaded from https://academic.oup.com/gji/article/206/3/1864/2583505 by Princeton University user on 11 September 2020

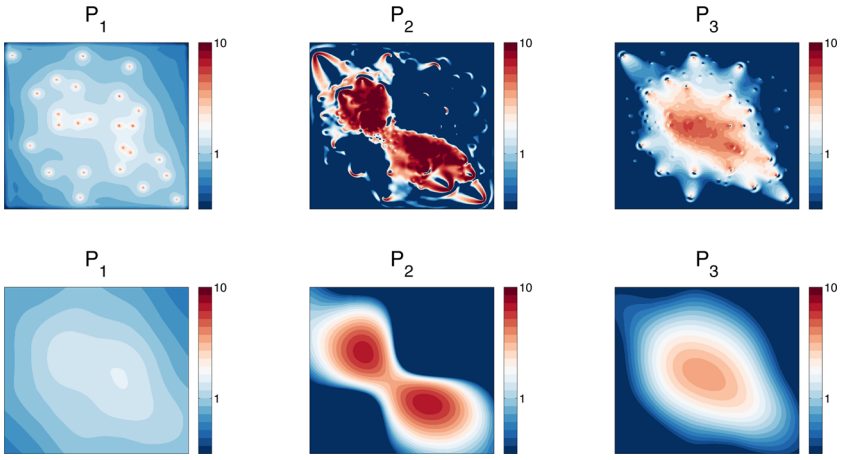

Figure 5. With most diagonal preconditioners, smoothing or damping is required to avoid numerical problems. Damping helps bring out small structures quickly, which can be useful in non-iterative migration. Smoothing bring out such details gradually, helping avoid local minima in inversion. The smoothing parameter has significant effects on numerical performance, as discussed in the text. Top: diagonal preconditioners before smoothing. Bottom: after smoothing.

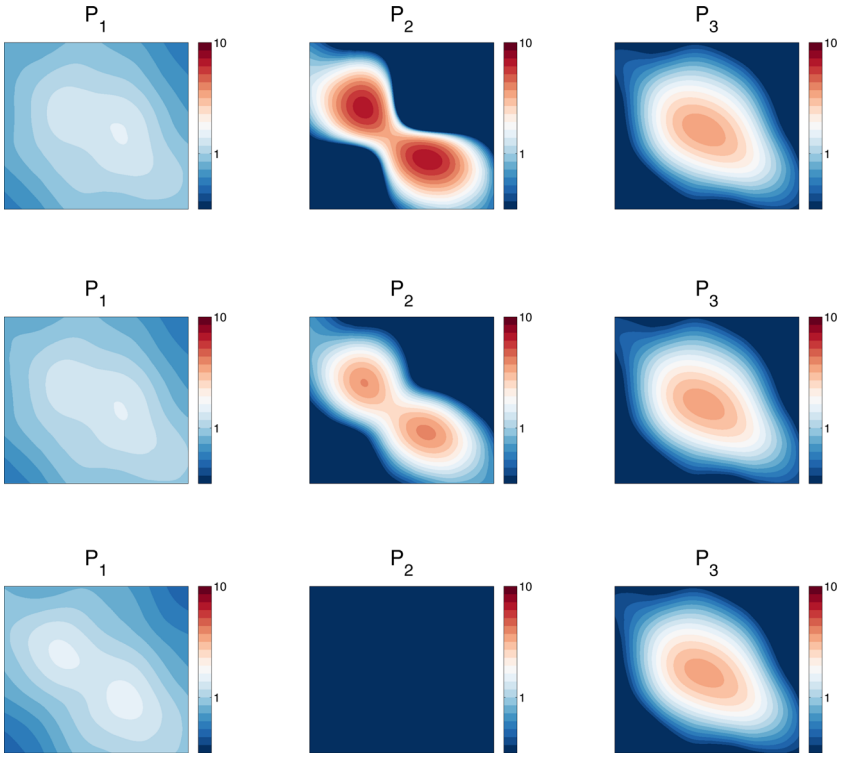

Figure 6. How often is it necessary to update preconditioners? The answer depends on how much the Hessian varies throughout the model space. Top: preconditioners computed using a homogeneous model. Middle: preconditioners computed using a smoothed target model. Bottom: preconditioners computed using an unsmoothed target model.

convergence, or even cause the inversion to fail. Interestingly, Symes (2008) describes a sense in which a filtering operation, similar to smoothing, provides a better underlying approximation to the Hessian.

6.3.2 Updating

Besides deciding how much to smooth, practitioners must choose how often to update a preconditioner to account for variations in the Hessian from one part of the model space to another. While updates improve the approximation to the Hessian, they also require forward and adjoint simulations whose cost might not be offset by faster convergence. Because updating the diagonal approximation Hessian amounts to variable preconditioning, it can also cause numerical difficulties of the type described by Knyazev & Lashuk (2007), which can be resolved by restarting the optimization procedure at the expense of slower convergence.

For diagonal preconditioners based on the waveform-difference Hessian, the question of how often to update depends on the relative size of the positive semi-definite first-order approximation P_1 and the remaining second-order contribution P_2. As shown in Fig. 7, P_1 varies a little and P_2 varies a lot throughout the model space. From the numerical experiments below, we find it is not necessary

Downloaded from https://academic.oup.com/gji/article/206/3/1864/2583505 by Princeton University user on 11 September 2020

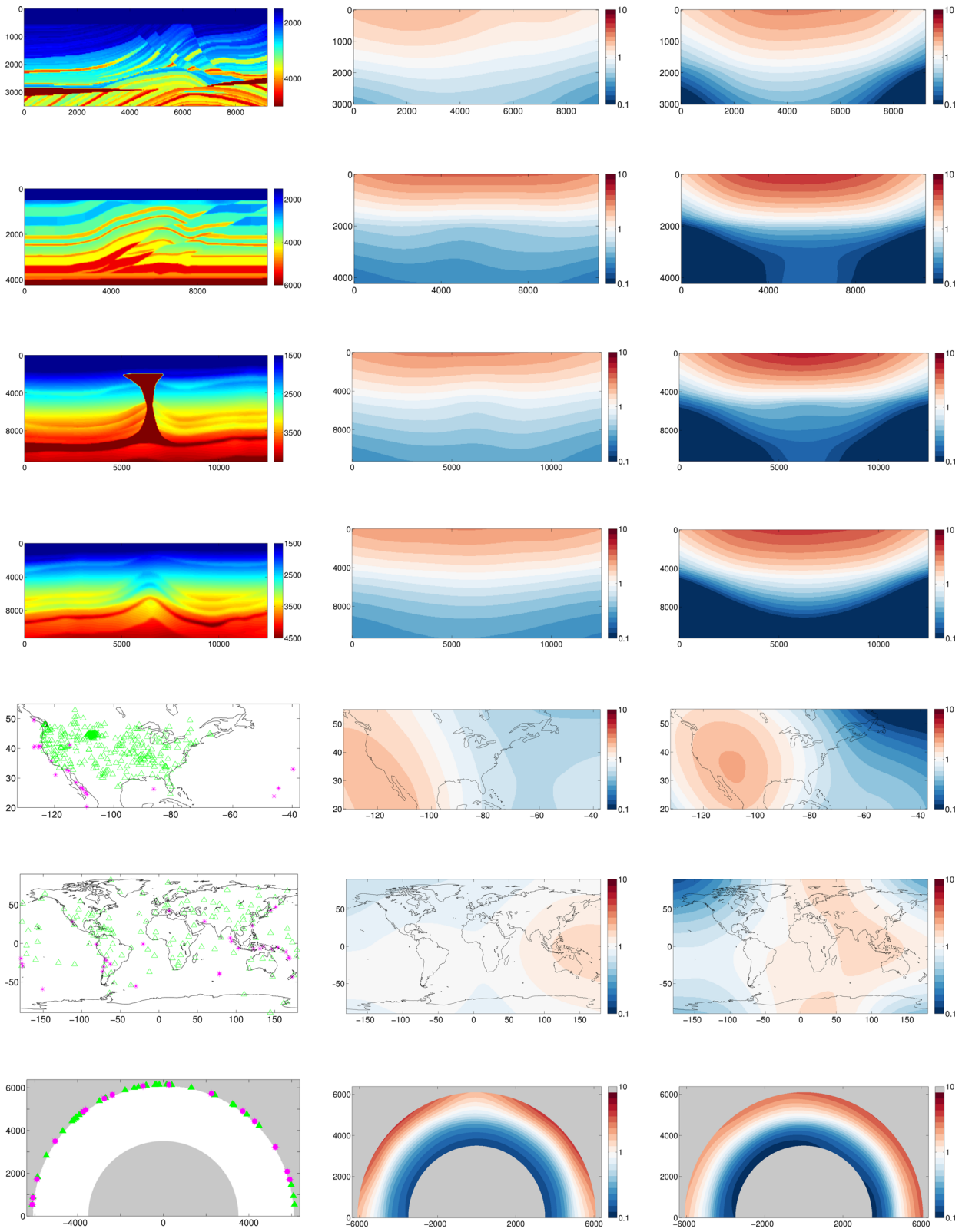

Figure 7. Left: target model and/or network. Middle: diagonal preconditioners $P_1 + P_2$. Right: diagonal preconditioners P_3.

Downloaded from https://academic.oup.com/gji/article/206/3/1864/2583505 by Princeton University user on 11 September 2020

to update either P_1, $P_1 + P_2$, or P_3 very often. In our experience, it is most effective to update diagonal preconditioners only at multiscale transitions, if at all.

6.4 Comparisons

After the smoothing procedure described above, we compared the numerical performance of preconditioners P_1, $P_1 + P_2$ and P_3 in the waveform inversion test cases. Figs 8 and 9 show the results of these experiments. In terms of convergence rate, P_1 and $P_1 + P_2$ provide virtually identical performance. The new scaling P_3 provides computational savings, sometimes quite significant, relative to the other two. In Figs 9–12, we use the label 'L-BFGS' for the case $D = I$, and 'rescaled L-BFGS' for the case $D = P_3^{-1}$, with D being the diagonal scaling described in Appendix A and I being the identity matrix. Likewise, we use the label 'NLCG' for the case $P = I$ and 'preconditioned NLCG' for the case $P = P_3$, with P being the preconditioner described in Appendix B.

7 REGULARIZATION

Without robust measures to mitigate nonconvexity and non-uniqueness in waveform inversion, the optimization algorithms and preconditioning strategies described above would be of little use. An important way of promoting convexity and suppressing non-uniqueness is to regularize, that is, to impose smoothness or other constraints on the model directly through the objective function or indirectly by other means.

Conceptually, regularization differs from preconditioning in a fundamental way: as a change of variables, preconditioning leaves the underlying optimization problem unchanged, while regularization, in effect, trades one problem for another more tractable problem (Engl *et al.* 2000).

Below, we describe several common regularization methods, discuss practical complications that arise during their use and compare their performance through waveform inversion test cases. Although the following techniques are all to one degree or another classical, some interesting new facts emerge from analysing them in the waveform inversion context.

7.1 Tikhonov regularization

In Tikhonov regularization, a preference for smooth models enters through the addition of a penalty term

$$R_2(m) = \frac{\lambda}{2} \int_V \nabla m \cdot \nabla m \, \mathrm{d}V, \tag{15}$$

to the objective function, where λ is a user-supplied parameter that controls the weight relative to the data misfit function. More commonly, the right-hand side is written in discretized form as

$$\frac{\lambda}{2} \sum_i \left[(\partial_x m)_i^2 + (\partial_z m)_i^2 \right], \tag{16}$$

where the sum is over numerical grid points. Classic references on Tikhonov regularization include Hansen (1998) and Vogel (2002).

As an alternative to penalizing first-order spatial derivatives of the model, one can choose to work with higher-order spatial derivatives or some other measure of model roughness. In our experience, higher-order Tikhonov regularization often provides good results in regional and global inversions but not in near-surface problems.

Sometimes it makes sense to apply the penalty term not to the model itself, but to the difference between the model and some reference. We use such an approach for the deep Earth test case, applying the regularization term to the difference between the model and a radial reference model to avoid penalizing boundaries between the crust, mantle and core.

Although the theory is classical, application of Tikhonov regularization to waveform inversion is not straightforward. The behaviour of the penalty term can be quite different than in other seismic inverse problems, as illustrated in Figs 13 and 14 through a checkerboard example. The use of a fine model discretization, as required for the solver computations, leads to abrupt variations in the derivatives of the penalty term. While the penalty term still acts to smooth the updated model, it effectiveness is reduced. To work around this difficulty, it is possible to project from a fine grid for the solver to a coarse grid for computing spatial derivatives of the model and back again. The smoothness of the updated model in such an approach derives from a combination of the regularization term and the projection operator. Projection can also be used as a regularization method in its own right, as described later in Section 7.3.

7.2 Total variation regularization

Total variation (TV) regularization also acts directly through the objective function. The TV penalty term

$$R_{1,\epsilon}(m) = \lambda \int_V \sqrt{\nabla m \cdot \nabla m + \epsilon} \, \mathrm{d}V \tag{17}$$

is essentially the L_1 norm of the spatial derivatives of the model. In discretized form, the right-hand side becomes

$$\lambda \sum_i \sqrt{(\partial_x m)_i^2 + (\partial_z m)_i^2 + \epsilon}. \tag{18}$$

Because the TV penalty terms act through the L_1 norm, discontinuous transitions are not penalized any more than smooth transitions. As a result, the method is well-suited for recovering layered geologic structures.

While the damping parameter ϵ makes the above expressions differentiable, the effect of TV regularization on the gradient of the objective function is still problematic. To illustrate, Fig. 13 shows the contribution to the gradient from the TV penalty term for different ϵ values. Even with large damping, abrupt variations in the derivatives of the penalty term remain that add considerably to the numerical difficulty of an inversion. Numerical problems associated with the TV regularization, it turns out, are common to a range of scientific applications (Vogel 2002). Goldstein & Osher (2009) reviewed methods that have been proposed to make TV regularization numerically tractable. Recently, Lin & Huang (2015) applied one such workaround to acoustic and elastic waveform inversion with promising results.

7.3 Projection and convolution

Another way to apply regularization is through the basis functions used to represent the model (Engl *et al.* 2000; Mathé & Pereverzev 2003). The choice of a smooth basis, for example, imposes a degree of smoothness on the model.

In waveform inversion with finite-difference or finite-element solvers, stability conditions are usually too strict for the numerical

Downloaded from https://academic.oup.com/gji/article/206/3/1864/2583505 by Princeton University user on 11 September 2020

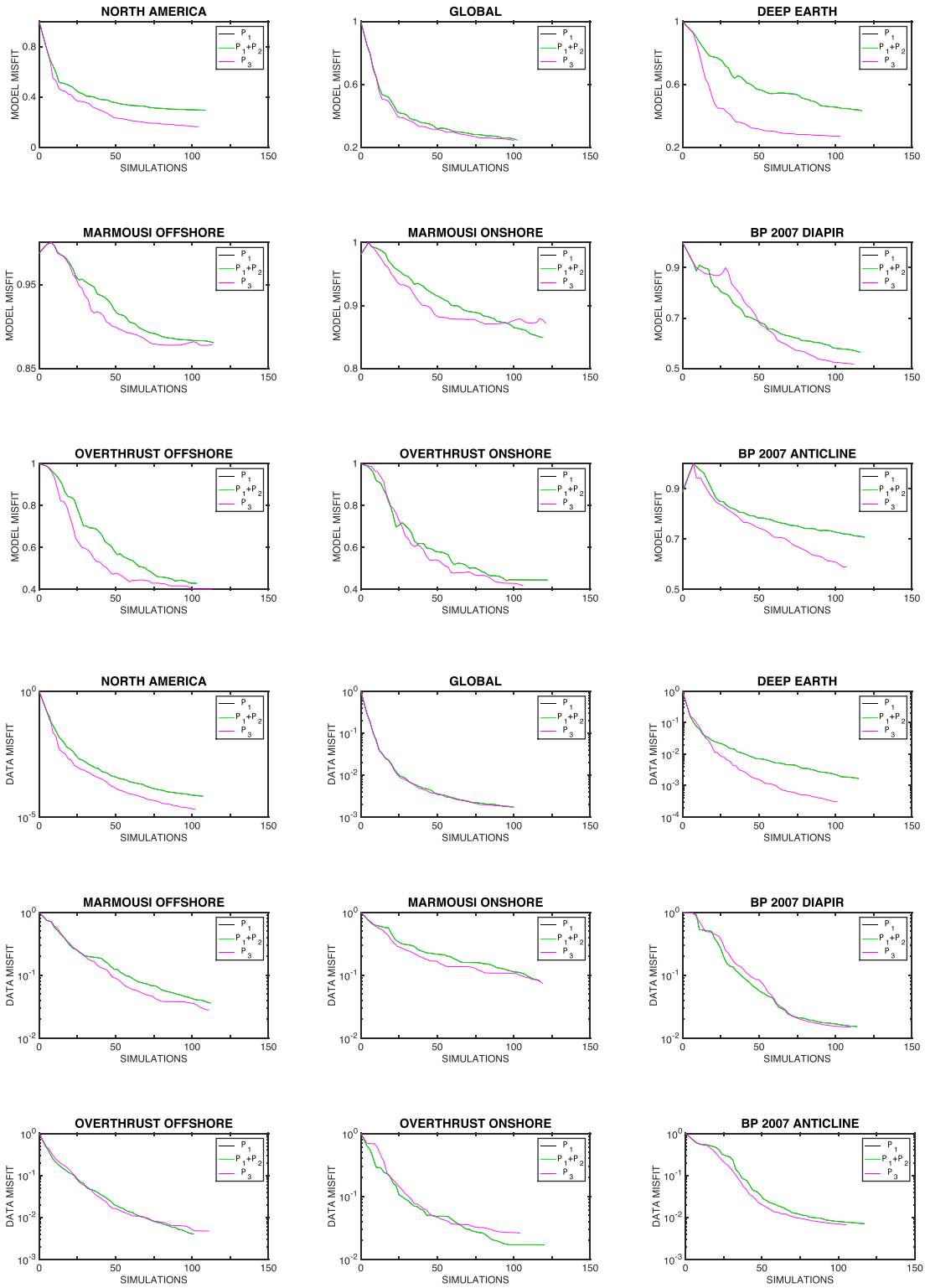

Figure 8. Comparison of diagonal preconditioners. P_1 and $P_1 + P_2$ provide almost identical performance. P_3 performs better than the other two.

grid to suppress non-uniqueness. It becomes necessary, if the goal is to mitigate non-uniqueness through the choice of basis functions, to project back and forth from the fine grid used by the solver to a coarser basis used for the model update procedure. The projection can be performed either explicitly, or as described by Peters *et al.* (2015), implicitly by formulating the inversion as a constrained optimization problem. While the primary use of projection is as a regularization method, faster convergence may be a beneficial side effect since the efficiency of gradient-based optimization methods improves with decreasing model space dimensionality (Sigmund & Petersson 1998).

A closely related method of regularization involves smoothing the gradient by convolving it with a Gaussian function or other kernel. While retaining mathematical properties of projection, convolution

Downloaded from https://academic.oup.com/gji/article/206/3/1864/2583505 by Princeton University user on 11 September 2020

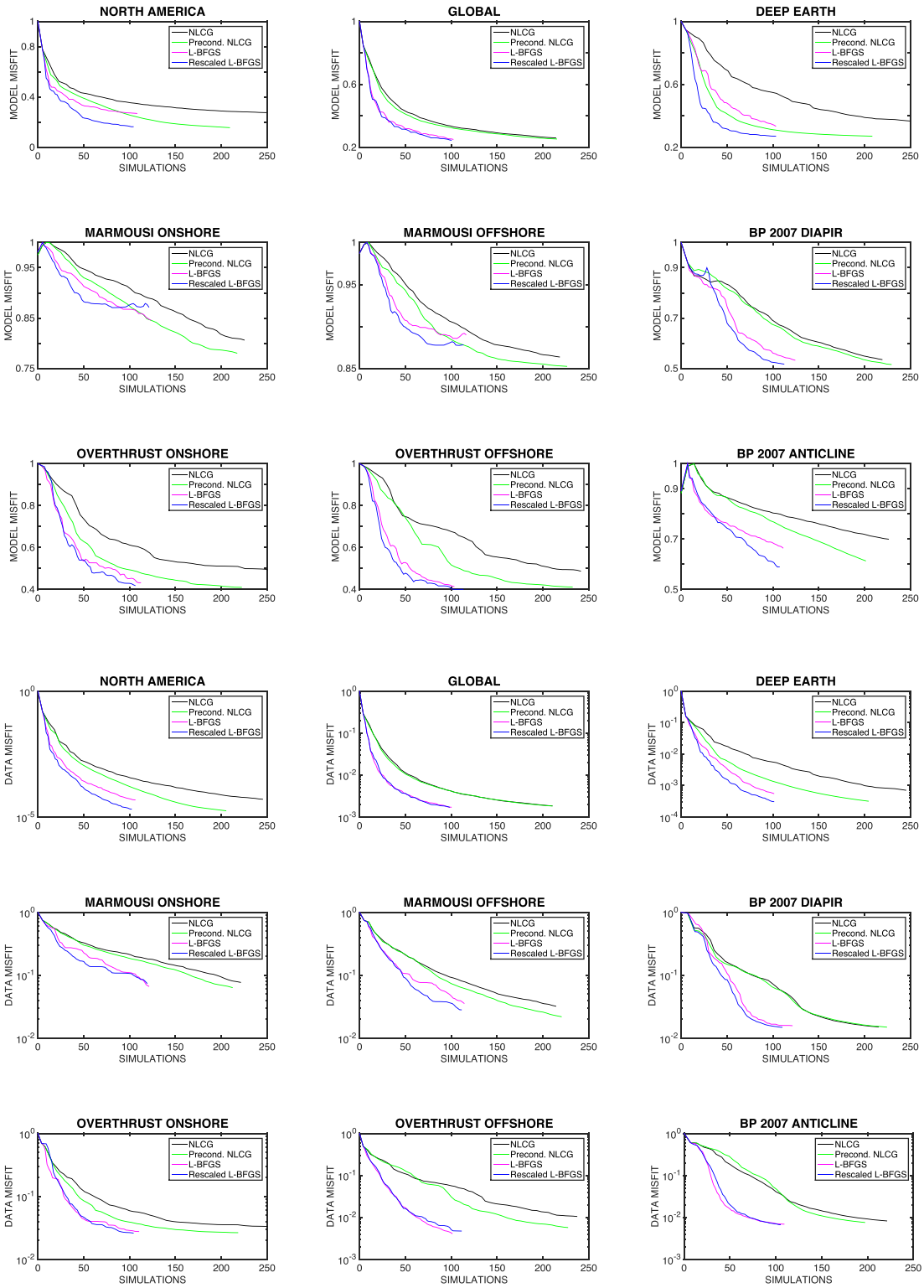

Figure 9. Comparison of optimization algorithms with and without preconditioning NLCG or rescaling the L-BFGS initial Hessian.

avoids the need to convert back and forth between two sets of basis functions, thus simplifying the inversion machinery. To see the connection between projection and convolution, consider a set of basis functions that differ only by spatial offset. Projection onto such a basis is equivalent to convolution with one of the basis elements. Convolution and other methods for smoothing the gradient have a long history in optimal control and design (Jameson 1988), and a more recent history in geophysical parameter estimation (Brenders & Pratt 2007; Oh & Min 2013; Alkhalifah 2015).

7.4 Choice of regularization parameters

The extent to which regularization affects the result of a given model update can be adjusted through user-supplied parameters.

Downloaded from https://academic.oup.com/gji/article/206/3/1864/2583505 by Princeton University user on 11 September 2020

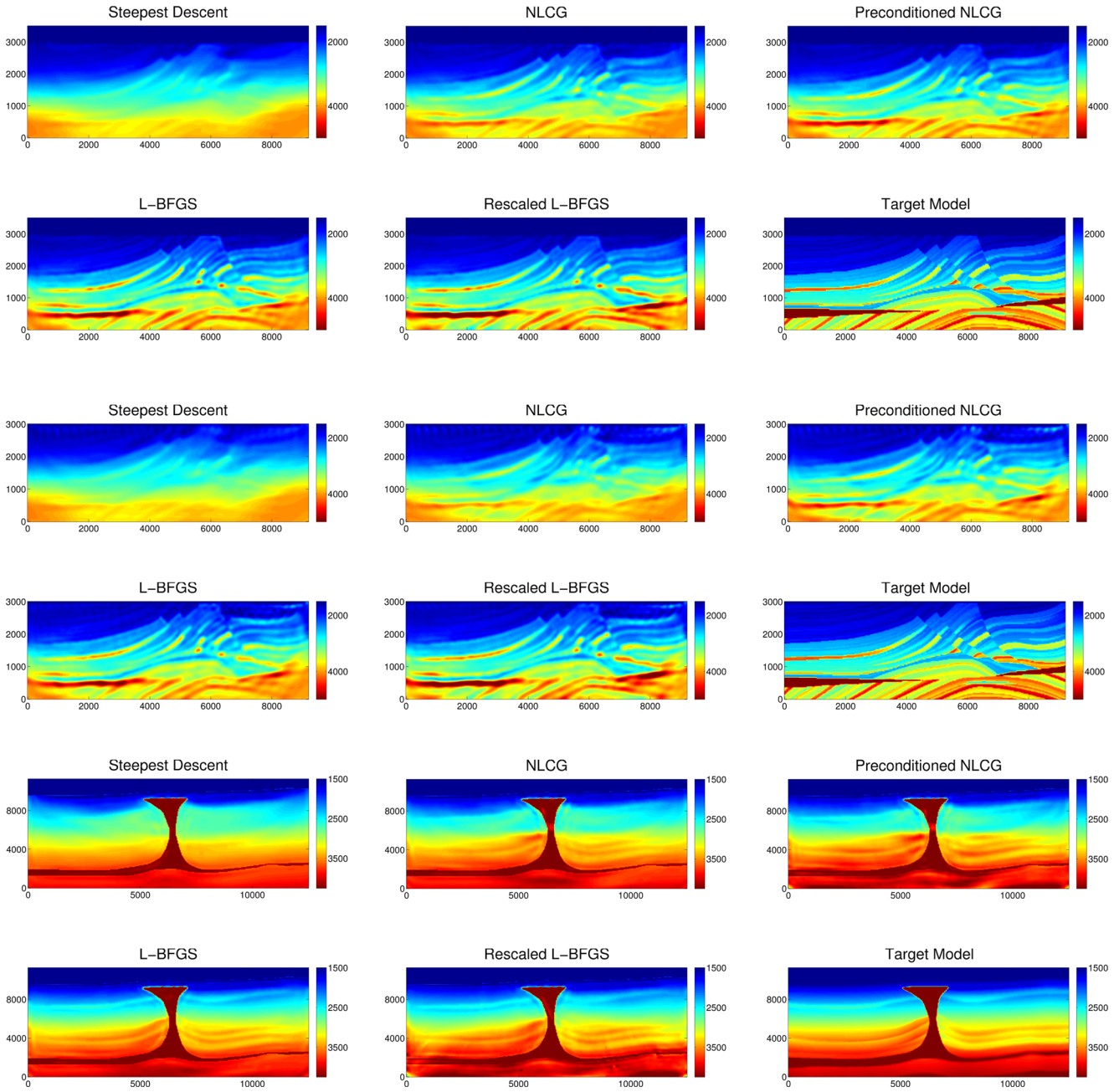

Figure 10. Comparison of nonlinear optimization algorithms. From top to bottom: *Marmousi offshore*, *Marmousi onshore* and *salt diapir* inversion results after 100 function/gradient evaluations.

In Tikhonov and TV regularization, the smoothness of the updated model is controlled in a straightforward way by the weight of the penalty term relative to the data misfit function. In projection and convolution, smoothness is a function of the spacing of the basis elements or the width of the convolution kernel. Such parameter choices are important for managing how quickly structure is added to the model and, in turn, for helping the updated model remain in the basin of convergence.

For testing the projection method of regularization, we explicitly convert back and forth between the fine solver basis and a coarser model update basis, using a set of spatially offset Gaussian functions as the elements of the latter. Relative to the solver grid spacing, we increase the spacing between Gaussian functions (as well as the radius of each Gaussian function) by a uniform factor f, so that in the two-dimensional waveform inversion test cases the ratio N of the number of basis functions to the number solver grid points is $N = f^{-2}$. For testing the convolution method of regularization, we use a Gaussian kernel to smooth the gradient prior to passing it to the search direction algorithm. The standard deviation $\sigma = \sigma_x = \sigma_z$ of the Gaussian function determines the amount smoothing.

To furnish values for the regularization method comparisons below, we adopt a 'brute force' approach, running inversions multiple times to determine the most effective parameter values for a given test case. Results from these experiments are shown in the online supplement. Far from suggesting a routine method for parameter selection, the goal of these experiments is to build intuition and to ensure that in the comparisons below, each regularization method performs at its best. Since regularization involves a trade-off

Downloaded from https://academic.oup.com/gji/article/206/3/1864/2583505 by Princeton University user on 11 September 2020

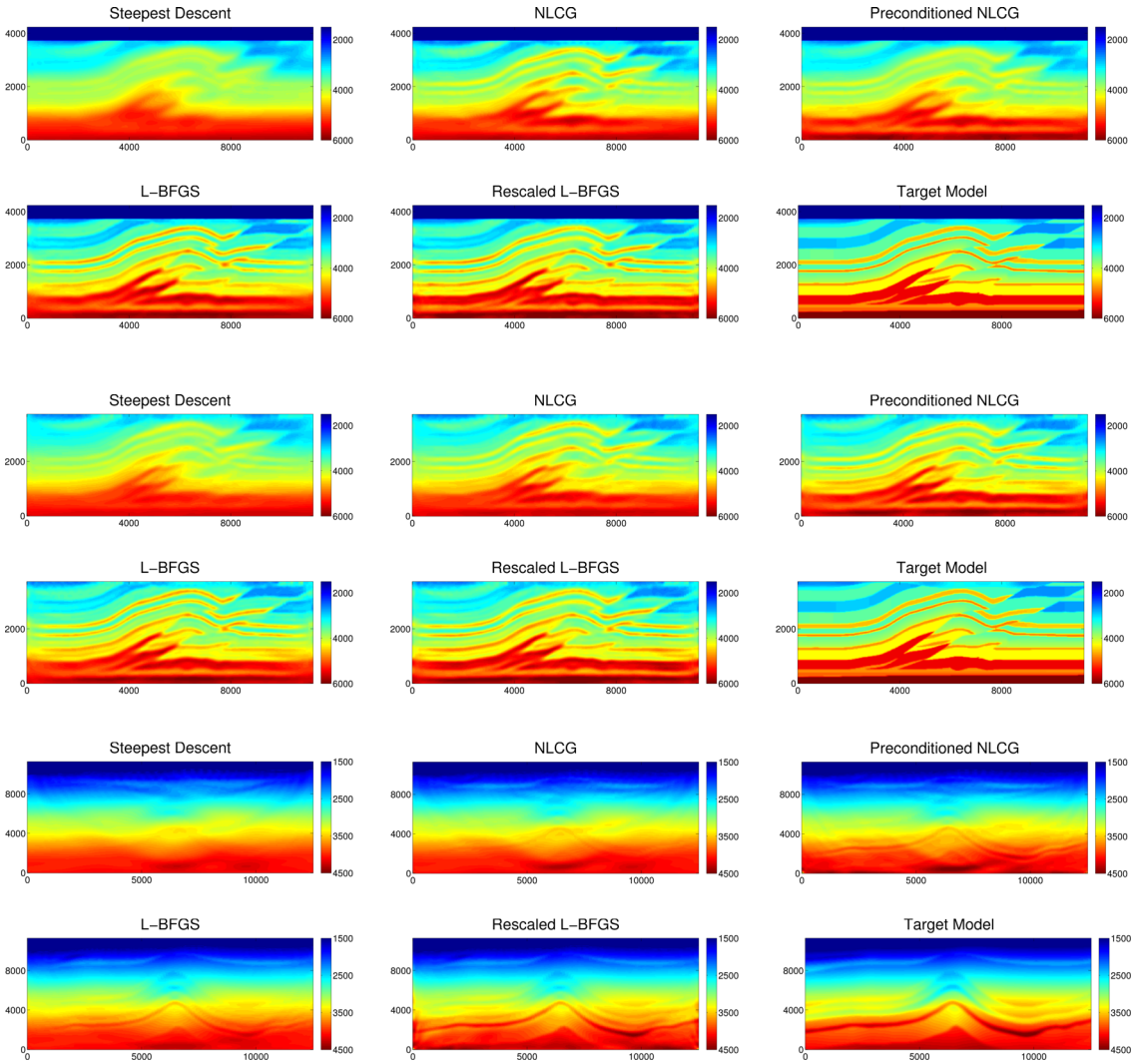

Figure 11. Comparison of nonlinear optimization algorithms. From top to bottom: *overthrust offshore*, *overthrust onshore* and *anticline* inversion results after 50, 50 and 100 function/gradient evaluations, respectively. Because the *overthrust* inversions converge more quickly, results are shown after half the usual number of simulations.

between fitting the data and suppressing non-uniqueness, increased convergence rate in one norm is often accompanied by decreased convergence rate in the other. As a result, decisions about which parameter value is best, in our approach, are made solely on the basis of model misfit. A number of less exhaustive, more practical methods for parameter selection are discussed by Vogel (2002).

7.5 Comparisons

Results from numerical experiments using the inversion test cases, regularization methods and parameter selection procedure described above are shown in Figs 15–18.

Of all approaches, the performance of Tikhonov regularization is perhaps the most surprising. Since it acts through L_2 norm of the spatial derivatives of the model, Tikhonov regularization favours smooth transitions over discontinuous transitions in the model. Given this preference, it might be expected to perform well in recovering the regional model, which contains only smoothly varying structures, but this is not the case. Poor performance on the regional test case appears closely related to the model discretization. Because the spatial derivatives of the model are computed using the fine numerical grid, Tikhonov regularization is much less effective than projection or convolution in smoothing the updated model. Increasing the weight on the regularization term does not reliably solve the problem because with gradient-based methods, the weight on the penalty term cannot exceed a certain value or the optimization algorithm becomes unstable.

While performing poorly on the regional example, Tikhonov regularization does well on all other test problems, especially the near-surface problems. The success of the method in these cases has a lot to do with improved accuracy at depth. By smoothing layer interfaces, Tikhonov regularization allows deep layers corresponding to reflected phases to shift position as shallow layers become more accurately recovered. Such a mechanism is important early in an inversion, when incomplete recovery of shallow structure leads to systematic errors in deep structure. Crucially, the smoothing effect is not large enough to prevent the emergence of layered structures, so updated models are able to generate reflections.

Not surprisingly, TV regularization performs well in test cases involving discontinuous structures, especially the salt test case. What

Downloaded from https://academic.oup.com/gji/article/206/3/1864/2583505 by Princeton University user on 11 September 2020

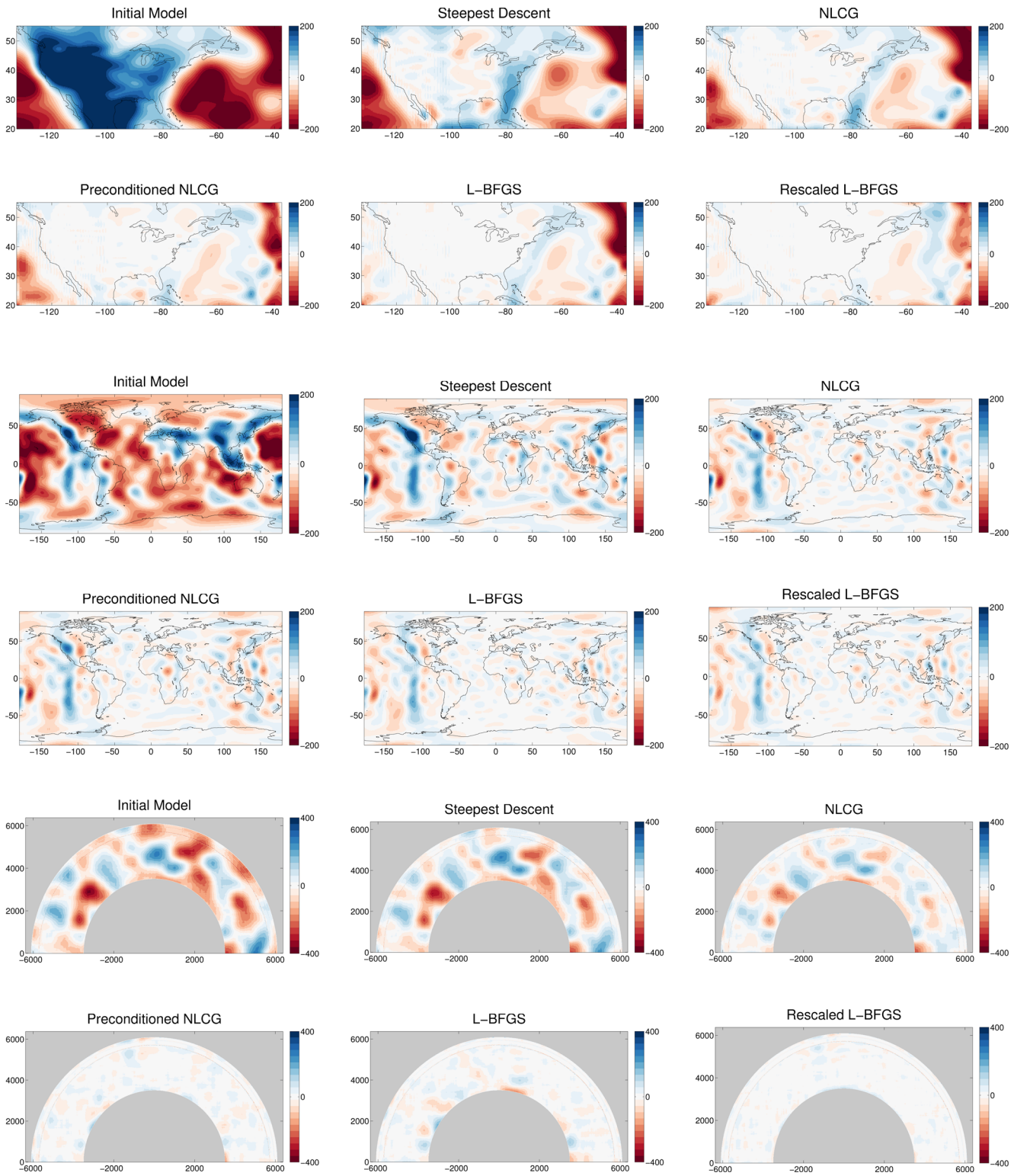

Figure 12. Comparison of nonlinear optimization algorithms. From top to bottom: *regional*, *global* and *Deep Earth* inversion results after 100 function/gradient evaluations. Because regional and global models lack the kind of coherent geologic structures found in the other test cases, we plot the difference between inverted and target models, rather than inverted models themselves.

is surprising is that, despite its advantage in resolving layered structures, TV regularization performs worse than Tikhonov regularization in all near-surface test cases, which can be attributed to the well-known nonlinearity and ill-conditioning of the TV penalty function (Goldstein & Osher 2009).

In most test cases, projection and convolution provide outcomes that, while similar to one another, differ markedly from Tikhonov and TV regularization results. In the regional test case, projection and convolution provide considerably better results than either penalty function method, with about two times greater reduction in

Downloaded from https://academic.oup.com/gji/article/206/3/1864/2583505 by Princeton University user on 11 September 2020

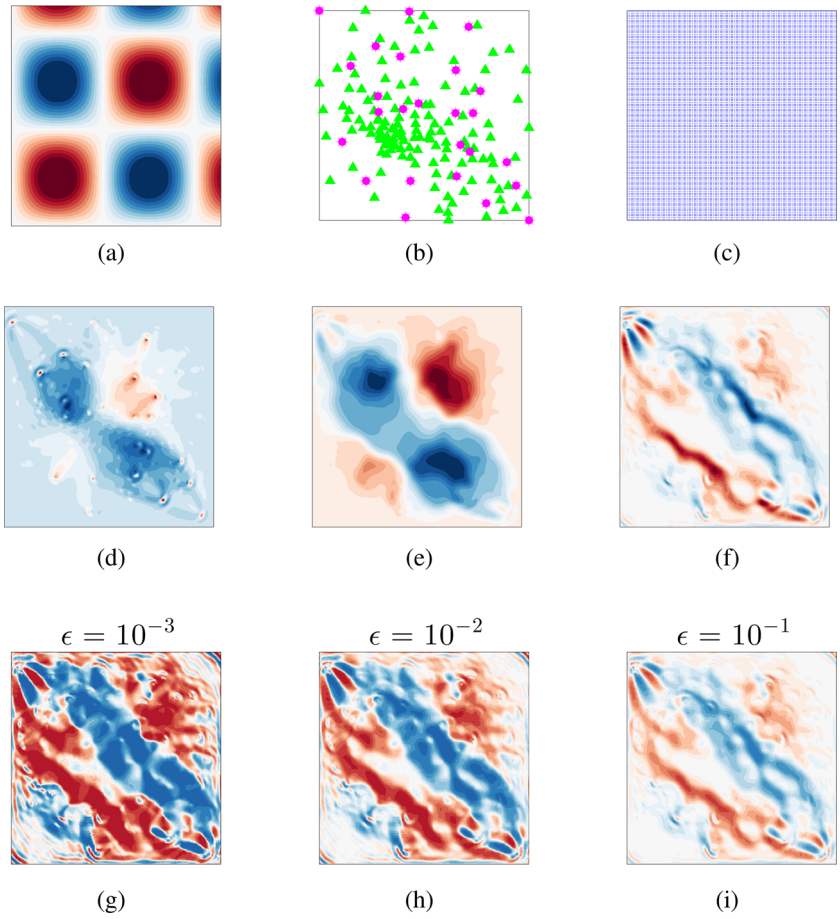

Figure 13. Tikhonov and total variation regularization illustrated using a checkerboard example. (a) Target model. (b) Network. (c) Numerical mesh. (d) Gradient of data misfit function. (e) Gradient of data misfit function after applying source–receiver corrections described in Section 8.2, which are essential for avoiding instability from the regularization penalty term. (f) Contribution to the gradient of the objective function from the Tikhonov penalty term. (g–i) Contribution to the gradient of the objective function from the total variation penalty term, with various choices of damping parameter ϵ.

model error and 10^2 times greater reduction in data misfit. Close inspection of Figs 16 and 17 shows that convolution performs better than projection in the near-surface examples. The reason, we believe, is that while convolution allows small-scale structures to emerge over multiple model updates (even if no individual update contains such structures), projection to a coarser basis unavoidably limits recovery of such details. Close inspection of Fig. 18 suggests that projection is more effective than convolution at suppressing non-uniqueness in regional and global inversions.

Looking at all test cases together, the importance of a problem-dependent perspective on regularization comes across strongly. While all four regularization methods are found to be useful in one way or another, clear differences emerge between problems. Tikhonov regularization performs well when challenging small-scale structures are present, as in the near-surface problems. TV regularization provides slower convergence than Tikhonov regularization in most cases as a result of well-known numerical difficulties. Although we do not experiment with workarounds of the type described by Goldstein & Osher (2009), Lin & Huang (2015) showed that TV regularization can be successfully adapted to waveform inversion problems through such measures. Rather than simply being beneficial, projection or convolution may be all but required for dealing with highly uneven source–receiver distributions. Extrapolating from these results, we predict projection or convolution is most effective far away from the global minimum, Tikhonov regularization is effective closer to the global minimum and TV regularization without numerical workarounds is only effective very close to the global minimum. In practice, use of two or more regularization methods at once may provide some advantages. A combination of Tikhonov and TV penalty terms in the objective function may be particularly beneficial, helping avoid problems associated with either method alone (Lin & Huang 2015).

8 OTHER CONSIDERATIONS

Next, we describe considerations that, although important to the success or efficiency of an inversion, do not fit well into any of the previous categories.

8.1 Multiscale transitions

In Section 7, we compared regularization methods under controlled conditions, without varying the data misfit, data filtering, or regularization parameters from one model update to another. While good for building intuition, such an approach is not enough to get reliably to the global minimum of a waveform inversion objective function. To avoid problems along the way, robust multiscale procedures are needed (Bunks *et al.* 1995).

Downloaded from https://academic.oup.com/gji/article/206/3/1864/2583505 by Princeton University user on 11 September 2020

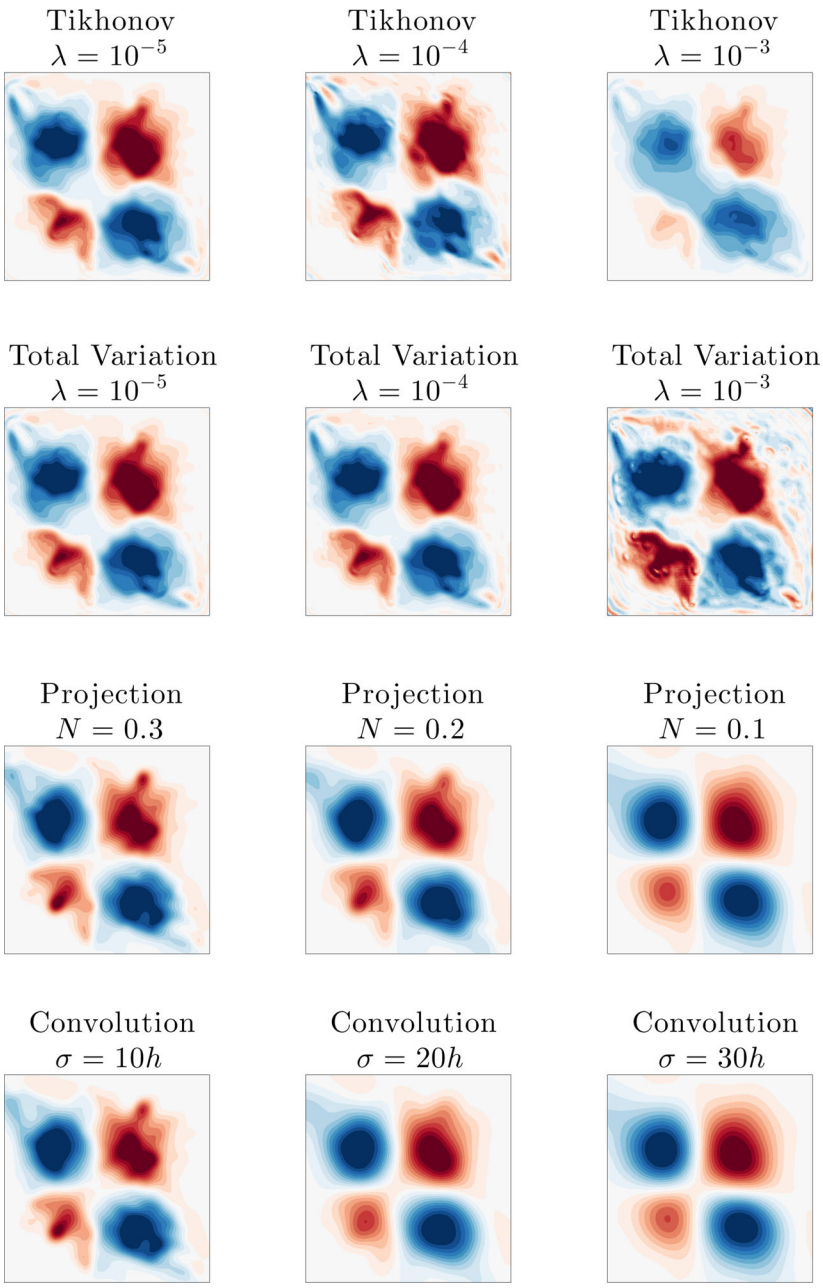

Figure 14. Behaviour of regularization methods illustrated through a checkerboard example. Each panel above shows the inversion result after five model updates from a homogeneous starting model, with the regularization method and weight varied from one panel to another. Perhaps surprisingly, Tikhonov regularization is less effective than projection or convolution at smoothing the updated model. Numerical difficulties from the use of total variation regularization are apparent even at this early stage in the inversion.

Many multiscale procedures involve modification of the objective function. It may be beneficial to restart the optimization algorithm following such a change. By restarting, one avoids making comparisons in the NLCG and L-BFGS algorithms between the current and previous gradient, which may not be valid if the objective function has changed.

To determine best practices, we examined the performance of the L-BFGS algorithm in a two-level multiscale procedure with and without restarting. Modifying the objective function through the data filtering parameters, we carried out 25 model updates at low frequency and 50 model updates at high frequency. In terms of dominant frequency, the two multiscale levels differed by a factor of two.

The online supplement shows the performance of L-BFGS in these experiments. In all cases, restarting the algorithm at the transition between multiscale levels led to faster convergence. In three cases, the effect was relatively small (less than a one or two model updates' difference in computational cost), in two cases it was larger and in the remaining four cases not restarting caused the optimization algorithm to fail outright.

8.2 Near field artefacts

From inaccuracy in the numerical treatment of wave propagation in the close vicinity of sources and receivers, the gradient of the data

Downloaded from https://academic.oup.com/gji/article/206/3/1864/2583505 by Princeton University user on 11 September 2020

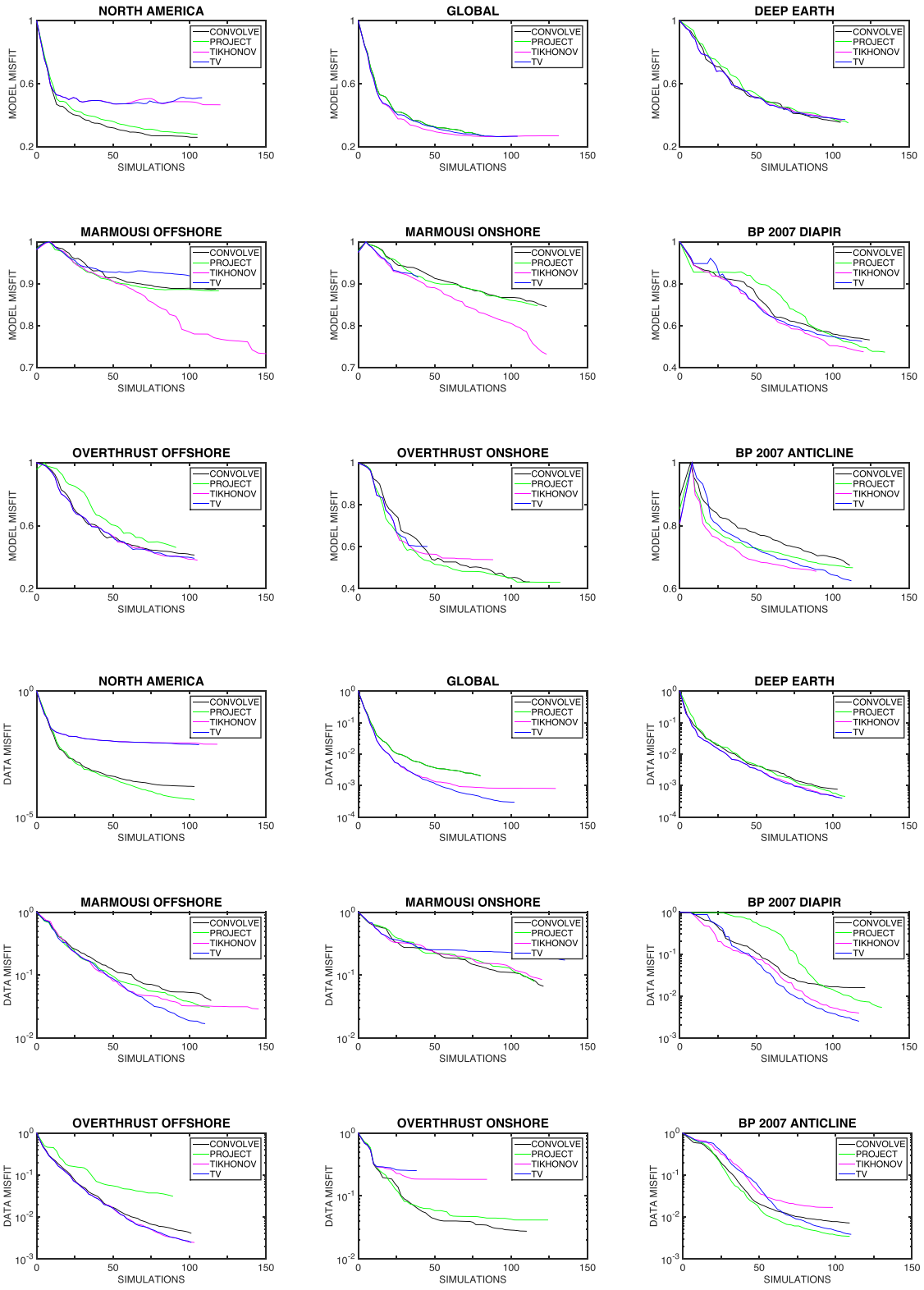

Figure 15. Comparison of regularization methods. To ensure that each method performed at its best, we adopted a 'brute force' approach to the selection of regularization parameters. The effectiveness of projection, convolution, Tikhonov and total variation methods of regularization is found to be highly problem dependent.

misfit function computed using the adjoint of the forward solver is commonly found to contain spurious near field features. Whether or not a correction is required depends on the regularization method employed. Convolution with a Gaussian kernel or projection onto a coarse basis tend to smooth out near field artefacts, so an additional correction is not generally required in these cases. Tikhonov or TV regularization, on the other hand, is not effective in smoothing out such features, so a correction is required in these cases.

Downloaded from https://academic.oup.com/gji/article/206/3/1864/2583505 by Princeton University user on 11 September 2020

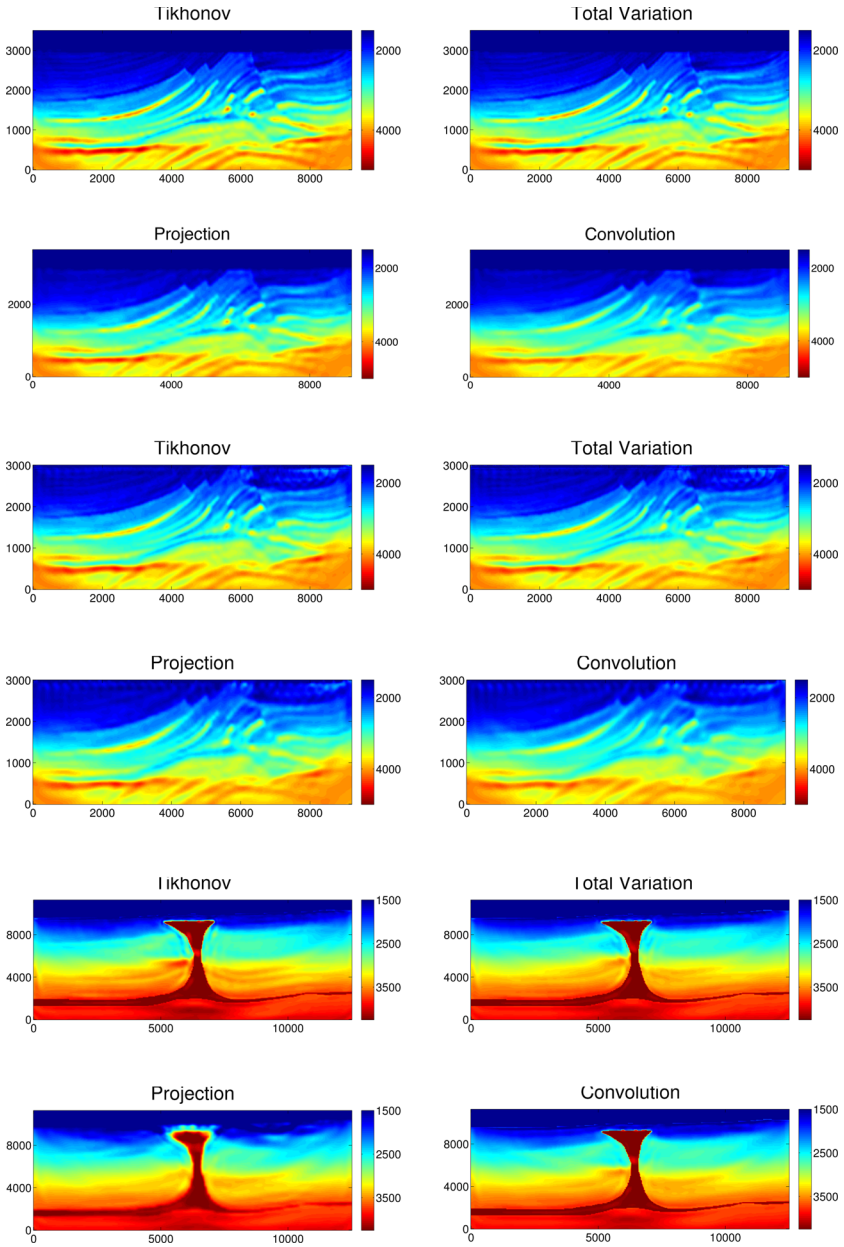

Figure 16. Comparison of regularization methods. From top to bottom: *Marmousi offshore*, *Marmousi onshore* and *salt diapir* inversion results after 50 function/gradient evaluations.

To illustrate the problem, consider the checkerboard example in Fig. 13. Panel (a) shows the true model, panel (b) the locations of sources and receivers and panel (d) the gradient with respect to a homogenous initial model. Numerical artefacts around sources and receivers give the gradient a pockmarked appearance. In finite-element or finite-difference forward modelling, such artefacts can be removed by smoothing within a radius of one or two elements or grid points around each source and receiver. The use of this of type of procedure in waveform-difference inversion, it turns out, strongly parallels the use of source and receiver corrections of the type described by Tian *et al.* (2007) in traveltime inversion.

Let x_i, $i = 1, \ldots, N_r + N_s$ denote the location of the ith source or receiver. To correct the raw gradient of the data misfit function, we compute for each x_i

$$g_i = \int_V g_{\text{raw}}(x) \exp\left[-\left(\frac{x - x_i}{h}\right)^2 \right] \mathrm{d}V, \tag{19}$$

using a quadrature rule that is appropriate for the given numerical discretization and a value h that is one or two times the grid or element spacing. In subsequent computations, we use the corrected gradient given by

$$g(x) = \frac{1}{N_r + N_s} \sum_{i=1}^{N_r+N_s} \left\{ g_{\text{raw}}(x) + [g_i - g_{\text{raw}}(x)] \times \exp\left[-\left(\frac{x - x_i}{h}\right)^2 \right] \right\}. \tag{20}$$

In Fig. 13, the result of applying this correction to the raw gradient in panel (d) is shown in panel (e).

Downloaded from https://academic.oup.com/gji/article/206/3/1864/2583505 by Princeton University user on 11 September 2020

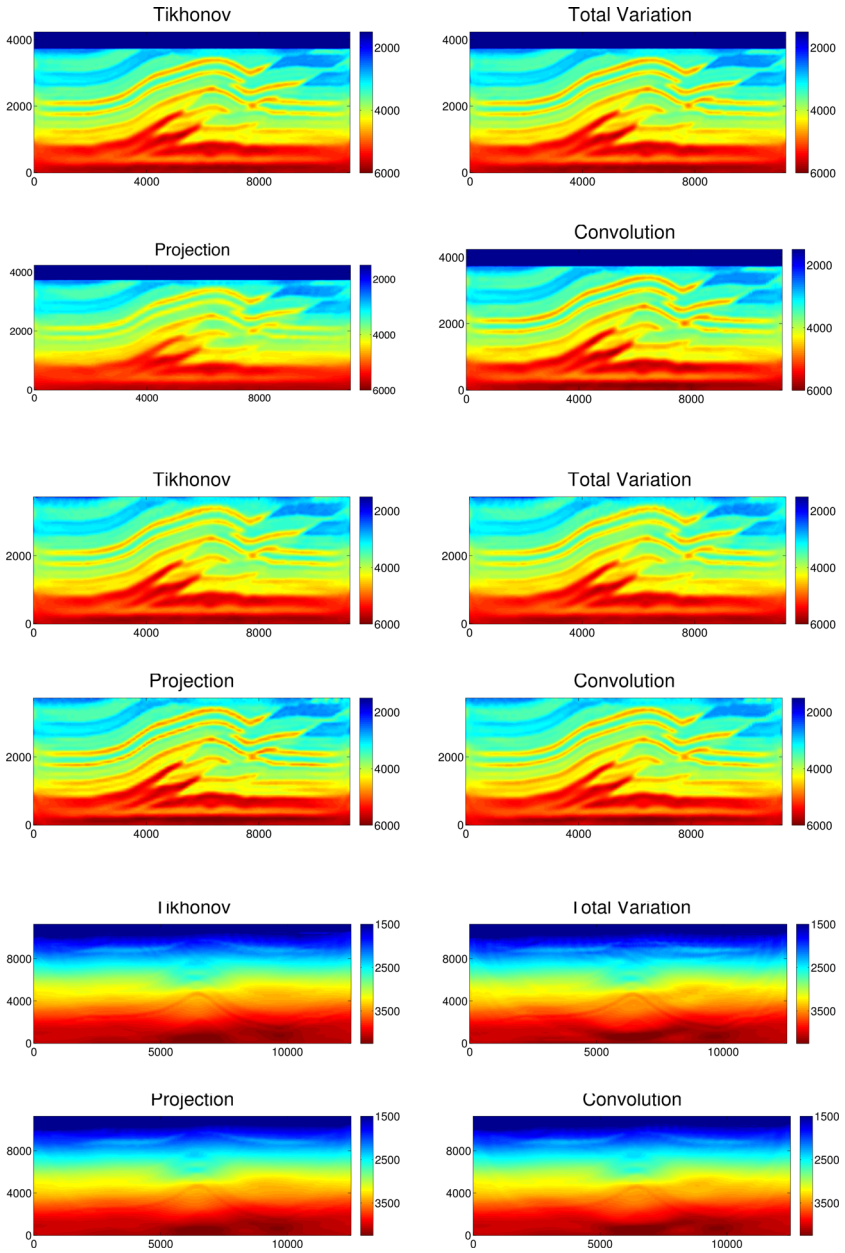

Figure 17. Comparison of regularization methods. From top to bottom: *overthrust offshore*, *overthrust onshore* and *anticline* inversion results after 50 function/gradient evaluations.

8.3 Masking strategies

Many inversions involve a water layer or other well-constrained region. If the properties of such an area are known with certainty, the corresponding model parameters can be excluded from the inversion. In other cases, it is better to include them in some way, for example, because there is uncertainty regarding the lower boundary of a salt structure or the properties of a water layer.

Bayesian methods, while arguably the most natural approach for incorporating such constraints, can add considerable complexity to an inversion. To find a simpler alternative, we compare two other strategies. The first involves modifying the NLCG preconditioner or the L-BFGS initial Hessian. In this approach, diagonal elements corresponding to well-constrained model parameters are scaled away from zero. The second strategy, which we call masking, involves ad hoc scaling of the gradient of the objective function. In this approach, gradient values corresponding to well-constrained model parameters are scaled toward zero.

We tested both methods using offshore exploration test cases. Masking the gradient performs better than rescaling the preconditioner, sometimes significantly better. By limiting changes to the water layer, both methods perform better than if no scaling is applied. Supporting results are provided in the online supplement. Although we experimented with water layers, the main usefulness of such methods, we anticipate, would be in dealing with salt structures.

9 CONCLUSIONS

Drawing on all of the results above, we suggest the following ‘best practices’ for seismic waveform inversion.

Downloaded from https://academic.oup.com/gji/article/206/3/1864/2583505 by Princeton University user on 11 September 2020

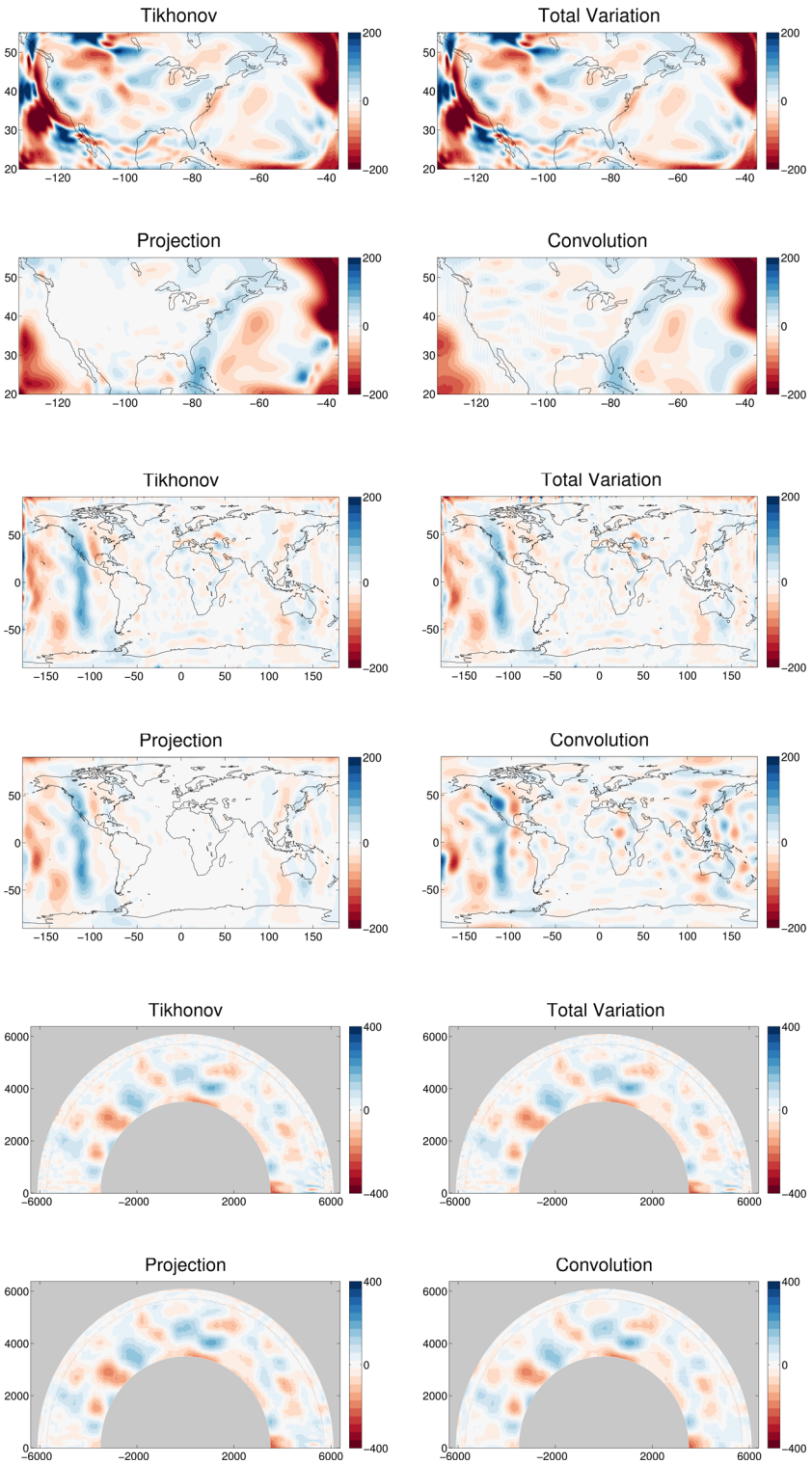

Figure 18. Comparison of regularization methods. From top to bottom: *regional*, *global* and *deep Earth* inversion results after 50 function/gradient evaluations. Because regional and global models lack the kind coherent geologic structures found in the other test cases, we plot the difference between inverted and target models, rather than inverted models themselves.

For nonlinear optimization, we recommend L-BFGS over NLCG. Average savings of 1/3 to 1/2 in regional, global and near-surface test cases make L-BFGS the clear winner in terms of computational efficiency, on which the choice of one optimization algorithm over another primarily depends. L-BFGS provides other advantages as well, including the potential for a safeguarded backtracking search in place of a more complicated bracketing line search. Even with robust regularization and multiscale strategies, numerical problems can occur with both L-BFGS and NLCG. Restart conditions provide a way of detecting and recovering from such difficulties.

Downloaded from https://academic.oup.com/gji/article/206/3/1864/2583505 by Princeton University user on 11 September 2020

In place of diagonal scalings obtained by application of the adjoint operator to the data or to the difference between data and synthetics, we recommend a new scaling, which is shown to provide faster, more reliable convergence. To avoid numerical problems, it is necessary to smooth or damp diagonal preconditioners; we recommend smoothing as part of a strategy for bringing out details gradually over many model updates. The amount of smoothing required in practice can be quite large. Regularly updating preconditioners to account for variations in the Hessian from one part of the model space to another does not appear to be cost effective.

For regularization, we stress the importance of a problem dependent perspective. Tikhonov regularization performs well in near-surface test cases, the effect being large enough to promote convexity and suppress non-uniqueness, but not large enough to prevent recovery of layered structures. TV regularization suffers from well-known numerical difficulties, but workarounds have been developed that make the method competitive. Regularization by projection or convolution is often required for dealing with the types of source–receiver distributions and starting models that commonly occur in regional and global seismology.

ACKNOWLEDGEMENTS

We thank Carl Tape and Andrea Morelli for very useful comments and suggestions. This research was partially supported by NSF grant 1112906.

REFERENCES

Akçelik, V. *et al.*, 2003. High resolution forward and inverse earthquake modeling on terascale computers, in *Proceedings of the 2003 ACM/IEEE Conference on Supercomputing,* New York, doi:10.1145/1048935.1050202.

Alkhalifah, T., 2015. Scattering-angle based filtering of the waveform inversion gradients, *Geophys. J. Int.,* **200,** 363–373.

eds Biegler, L., Ghattas, O., Heinkenschloss, M. & van Bloemen Waanders, B., 2003. *Large-Scale PDE-Constrained Optimization,* Springer.

Brenders, A. & Pratt, R., 2007. Full waveform tomography for lithospheric imaging: results from a blind test in a realistic crustal model, *Geophys. J. Int.,* **168,** 133–151.

Bunks, C., Fatimetou, M., Zaleski, S. & Chavent, G., 1995. Multiscale seismic waveform inversion, *Geophysics,* **60,** 1457–1473.

Burstedde, C. & Ghattas, O., 2009. Algorithmic strategies for full waveform inversion: 1D experiments, *Geophysics,* **74,** WCC37–W3346.

Castellanos, C., Métivier, L., Operto, S., Brossier, R. & Virieux, J., 2015. Fast full waveform inversion with source encoding and second-order optimization methods, *Geophys. J. Int.,* **200,** 718–742.

Claerbout, J. & Nichols, D., 1994. Spectral preconditioning, Stanford Exploration Project, Report 82, 183–186.

Dahlen, F., Hung, S.-H. & Nolet, G., 2000. Fréchet kernels for finite-frequency traveltimes—I. Theory, *Geophys. J. Int.,* **141,** 157–174.

Demanet, L., Létourneau, P.-D., Boumal, N., Calandra, H., Chiu, J. & Snelson, S., 2011. Matrix probing: a randomized preconditioner for the wave-equation Hessian, http://adsabs.harvard.edu/abs/2011arXiv1101.3615D.

Dennis, J. & Schnabel, R., 1996. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations,* SIAM.

Ekström, G., Abers, G. & Webb, S., 2009. Determination of surface-wave phase velocities across USArray from noise and Aki's spectral formulation, *Geophys. Res. Lett.,* **36,** L18301, doi:10.1029/2009GL039131.

Engl, H., Hanke, M. & Neubauer, A., 2000. *Regularization of Inverse Problems,* Kluwer.

Fletcher, R., 1976. Conjugate gradient methods for indefinite systems, in *Lecture Notes in Mathematics,* vol. 506, pp. 73–89, ed. Alistair Watson, H., Springer.

Fletcher, R. & Reeves, C., 1964. Function minimization by conjugate gradients, *Comput. J.,* **7,** 149–154.

Gilbert, J. & Nocedal, J., 1992. Global convergence properties of conjugate gradient methods for optimization, *SIAM J. Optim.,* **2,** 21–42.

Goldstein, T. & Osher, S., 2009. The split Bregman method for *L*1-regularized problems, *SIAM J. Imaging Sci.,* **2,** 323–343.

Gould, N., Orban, D. & Toint, P., 2005. Numerical methods for large-scale nonlinear optimization, *Acta Numerica,* **14,** 299–361.

Gunzburger, M., 2000. *Perspectives in Flow Control and Optimization,* SIAM.

Hansen, P., 1998. *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion,* SIAM.

Jameson, A., 1988. Aerodynamic design via control theory, *J. Sci. Comput.,* **3,** 233–260.

Knyazev, A. & Lashuk, I., 2007. Steepest descent and conjugate gradient methods with variable preconditioning, *SIAM J. Matrix Anal. Appl.,* **29,** 1267–1280.

Kolda, T., O'Leary, D. & Nazareth, L., 1998. BFGS with update skipping and variable memory, *SIAM J. Optim.,* **8,** 1060–1083.

Komatitsch, D. & Vilotte, J.-P., 1998. The spectral element method: an efficient tool to simulate the seismic response of 2D and 3D geologic structures, *Bull. seism. Soc. Am.,* **88,** 368–392.

Lin, Y. & Huang, L., 2015. Acoustic- and elastic-waveform inversion using a modified total-variation regularization scheme, *Geophys. J. Int.,* **200,** 489–502.

Liu, D. & Nocedal, J., 1989. On the limited memory BFGS method for large scale optimization, *Math. Program.,* **45,** 504–528.

Luo, Y., 2012. Seismic imaging and inversion based on spectral-element and adjoint methods, *PhD thesis,* Princeton University.

Marquering, H., Dahlen, F. & Nolet, G., 1999. Three-dimensional sensitivity kernels for finite-frequency traveltimes: the banana–doughnut paradox, *Geophys. J. Int.,* **137,** 805–815.

Mathé, P. & Pereverzev, S., 2003. Discretization strategy for linear ill-posed problems in variable hilbert scales, *Inverse Probl.,* **19,** 1263–1277.

Métivier, L., Bretaudeau, F., Brossier, R., Operto, S. & Virieux, J., 2014. Full waveform inversion and the truncated Newton method: quantitative imaging of complex subsurface structures, *Geophys. Prospect.,* **62,** 1353–1375.

Moré, J., Garbow, B. & Hillstrom, K., 1981. Testing unconstrained optimization software, *ACM Trans. Math. Softw.,* **7,** 17–41.

Nash, S., 2000. A survey of truncated-Newton methods, *J. Comput. Appl. Math.,* **124,** 45–59.

Nash, S. & Nocedal, J., 1991. A numerical study of the limited memory BFGS method and the truncated-Newton method for large scale optimization, *SIAM J. Optim.,* **1,** 358–372.

Nazareth, L., 1979. A relationship between the BFGS and conjugate gradient algorithms and its implications for new algorithms, *SIAM J. Numer. Anal.,* **16,** 794–800.

Nocedal, J., 1992. Theory of algorithms for unconstrained optimization, *Acta Numerica,* **1,** 199–242.

Nocedal, J. & Wright, S., 2006. *Numerical Optimization,* Springer.

Oh, J.-W. & Min, D.-J., 2013. Spectral filtering of gradient for l_2-norm frequency-domain elastic waveform inversion, *Geophys. J. Int.,* **193,** 820–840.

Peters, B., Smithyman, B. & Herrmann, F., 2015. Regularizing waveform inversion by projection onto intersections of convex sets, *TR-EOAS-2015-4, Tech rep., UBC*. https://www.slim.eos.ubc.ca/Publications/Public/TechReport/2015/peters2015EAGErwi/peters2015EAGErwi.html.

Polak, E. & Ribière, G., 1969. Note sur la convergence de méthodes de directions conjugués, *Rev. Fr. Inform. Rech. Oper.,* **3,** 35–43.

Powell, M., 1977. Restart procedures for the conjugate gradient method, *Math. Program.,* **12,** 241–254.

Rickett, J., 2003. Illumination-based normalization for wave-equation depth migration, *Geophysics,* **68,** 1371–1379.

Downloaded from https://academic.oup.com/gji/article/206/3/1864/2583505 by Princeton University user on 11 September 2020

Shin, C., Jang, S. & Min, D.-J., 2001. Improved amplitude preservation for prestack depth migration by inverse scattering theory, *Geophys. Prospect.,* **49,** 592–606.

Sigmund, O. & Petersson, J., 1998. Numerical instabilities in topology optimization: A survey of procedures dealing with checkerboards, mesh-dependencies and local minima, *Struct. Optim.,* **16,** 68–75.

Symes, W., 2008. Approximate linearized inversion by optimal scaling of prestack depth migration, *Geophysics,* **73,** R23–R35.

Tian, Y., Hung, S.-H., Nolet, G., Montelli, R. & Dahlen, F., 2007. Dynamic ray tracing and traveltime corrections for global seismic tomography, *J. Comput. Phys.,* **226,** 672–687.

Trampert, J. & Woodhouse, J., 2003. Global anisotropic phase velocity maps for fundamental mode surface waves between 40 and 150 s, *Geophys. J. Int.,* **154,** 154–165.

Tromp, J., Tape, C. & Liu, Q., 2005. Seismic tomography, adjoint methods, time reversal and banana-doughnut kernels, *Geophys. J. Int.,* **160,** 195–216.

van Leeuwen, T. & Herrmann, F., 2013. Fast waveform inversion without source-encoding, *Geophys. Prospect.,* **61,** 10–19.

van Leeuwen, T., Aravkin, A. & Herrmann, F., 2011. Seismic waveform inversion by stochastic optimization, *Int. J. Geophys.,* **2011,** doi:10.1155/2011/689041.

Vogel, C., 2002. *Computational Methods for Inverse Problems,* SIAM.

Woodward, M., 1992. Wave-equation tomography, *Geophysics,* **57,** 15–26.

Zou, X., Navon, I., Berger, M., Phua, K., Schlick, T. & Le Dimet, F., 1993. Numerical experience with limited-memory quasi-Newton and truncated Newton methods, *SIAM J. Optim.,* **3,** 582–608.

SUPPORTING INFORMATION

Additional Supporting Information may be found in the online version of this paper:

Figure S1. Performance of L-BFGS with different memory values.
Figure S2. Performance of NLCG with P_1 preconditioners.
Figure S3. Performance of L-BFGS with P_1 initial scalings.
Figure S4. Performance of L-BFGS with P_3 initial scalings.
Figure S5. Brute force parameter selection experiments: effect of varying Gaussian smoothing parameter.
Figure S6. Brute force parameter selection experiments: effect of varying the number of Gaussian basis functions used to represent the model. The ratio *f* is the number of basis functions divided by the number of points in the numerical grid. For a given test case, the former is varied from one experiment to another and the latter is constant.
Figure S7. Brute force parameter selection experiments: effect of varying Tikhonov regularization weight.
Figure S8. Brute force parameter selection experiments: effect of varying total variation regularization parameter.
Figure S9. Effect of restarting optimization algorithm at multiscale transitions.
Figure S10. Strategies for masking a water layer.
Figure S11. Role of regularization in suppressing non-uniqueness illustrated through a checkerboard example. Each panel shows the error in the inversion result after 25 updates from a homogeneous starting model, with the standard deviation σ of the Gaussian 'regularization by convolution' kernel varied from one panel to another. (http://gji.oxfordjournals.org/lookup/suppl/doi:10.1093/gji/ggw202/-/DC1)

Please note: Oxford University Press is not responsible for the content or functionality of any supporting materials supplied by the authors. Any queries (other than missing material) should be directed to the corresponding author for the paper.

APPENDIX A: LIMITED-MEMORY BFGS ALGORITHM

Main algorithm

Given an initial model m_0, objective function f, diagonal scaling D, memory value l and stopping threshold $\delta > 0$, the L-BFGS algorithm is as follows:

(1) Evaluate $f_0 = f(m_0), g_0 = \nabla f(m_0)$.
(2) Set $p_0 = -g_0$, $k = 0$.
(3) If $k > 0$, compute p_k from recursion, below.
(4) Compute α_k by line search and set $m_{k+1} = m_k + \alpha_k p_k$.
(5) Evaluate $f_{k+1} = f(m_{k+1})$, $g_{k+1} = \nabla f(m_{k+1})$.
(6) Set $s_k = m_{k+1} - m_k$, $y_k = g_{k+1} - g_k$, $k = k + 1$.
(7) Repeat (3–6) until $g_{k+1}^T g_{k+1} < \delta$.

Recursion

(1) Set $q = g_k$, $i = k - 1$, $j = \min(k, l)$.
(2) Perform j times: $\lambda_i = \frac{s_i^T q}{y_i^T s_i}$, $q = q - \lambda_i y_i$, $i = i - 1$.
(3) Set $\gamma = \frac{s_{k-1}^T y_{k-1}}{y_{k-1}^T y_{k-1}}$, $r = \gamma D q$, $i = k - j$.
(4) Perform j times: $\mu = \frac{y_i^T r}{y_i^T s_i}$, $r = r + s_i(\lambda_i - \mu)$, $i = i + 1$.
(5) End with result $p_k = -r$.

Remarks

L-BFGS' relatively modest memory usage rests on the fact that if k is the current iteration number and l is the memory value, then vector pairs prior to $\{s_{k-l}, y_{k-l}\}$ are no longer needed and can be removed from storage.

The scaling factor γ, which accounts for differences between the true Hessian and the approximation thereto, is essential to the good performance of the algorithm. Of several choices proposed by Liu & Nocedal (1989), the expression for γ given above (step 3 of Recursion) has been found to provide the best results.

APPENDIX B: PRECONDITIONED NONLINEAR CONJUGATE GRADIENT METHOD

Given an initial model m_0, objective function f, preconditioner P and stopping threshold $\delta > 0$, the preconditioned NLCG method is as follows:

(1) Evaluate $f_0 = f(m_0), g_0 = \nabla f(m_0)$.
(2) Solve $P y_0 = g_0$.
(3) Set $p_0 = -y_0, k = 0$.
(4) Compute α_k by line search and set $m_{k+1} = m_k + \alpha_k p_k$.
(5) Evaluate $f_{k+1} = f(m_{k+1}), g_{k+1} = \nabla f(m_{k+1})$.
(6) Solve $P y_{k+1} = g_{k+1}$.
(7) Set $\beta_{k+1} = \frac{g_{k+1}^T (y_{k+1} - y_k)}{g_k^T y_k}$, $p_{k+1} = -y_{k+1} + \beta_{k+1} p_k$, $k = k + 1$.
(8) Repeat (5–8) until $g_{k+1}^T g_{k+1} < \delta$.

The precise form of the algorithm above is due to Polak & Ribière (1969). Besides this one, a number of other variants exist. One due to Fletcher & Reeves (1964) may be slightly worse, and another due to Gilbert & Nocedal (1992) may be slightly better. Since savings from the use of L-BFGS over NLCG are in general much larger than savings from the use of one NLCG algorithm over another, we have not investigated which of these variants performs best in the waveform inversion context.

Downloaded from https://academic.oup.com/gji/article/206/3/1864/2583505 by Princeton University user on 11 September 2020

APPENDIX C: RESTART CONDITIONS

Even with robust regularization and multiscale methods, numerical difficulties in waveform inversion are neither unexpected nor uncommon. Below, we describe two restart conditions that can be effective in addressing such problems.

Angle restart condition

In Section 4.4, we discussed the possibility that a search direction is not actually a descent direction. For poorly conditioned problems, it may be beneficial to impose even stricter restart conditions than $p_k^T g_k > 0$.

One possibility for such a condition is

$$\frac{p_k^T g_k}{\sqrt{p_k^T p_k}\sqrt{g_k^T g_k}} > \tau ,$$

where $-1 < \tau < 0$ is some user-supplied parameter. Values of τ of about -0.02, we find, are usually effective.

The above restart condition can be reformulated in terms of the angle θ between the gradient and the search direction using the relation $\theta = \arccos(\tau)$. For example, the above condition with $\tau = -0.087$ is equivalent to requiring that $\theta > 95$.

Powell restart condition

In Section 4.4, we mentioned the tendency of NLCG search directions to lose conjugacy. As described by Powell (1977), the restart condition

$$\frac{g_{k+1}^T g_k}{g_k^T g_k} > \tau ,$$

where $\tau > 0$ is some user-supplied parameter, provides an effective workaround. A common choice for τ is 0.2.

Since conjugacy is expected of NLCG search directions but not L-BFGS search directions, Powell restart conditions can be used in combination with the former but not the latter. While performance after Powell restarts can be poor in the short time, the possibility of substantially better long-term performance makes the procedure worthwhile.

APPENDIX D: LINE SEARCH TERMINATION CONDITIONS

Given an objective function f, model m and search direction p, and letting $\phi(\alpha) = f(m + \alpha p)$, the work of the line search is to find a step length α such that the updated model $m + \alpha p$ meets the termination conditions

$$\phi(\alpha) \leq \phi(0) + c_1 \alpha \phi'(0),$$

$$\phi'(\alpha) \geq c_2 \phi'(0),$$

where $c_1 > 0$ and $0 < c_2 < 1$ are user-supplied numerical parameters. The first condition above is called the *Armijo condition* or *sufficient decrease condition*. The second is called the *curvature condition*. Both together are known as the *Wolfe conditions*.

It had been found that L-BFGS is best implemented with a very loose line search. Following Liu & Nocedal (1989) and many other studies, a common choice of numerical parameters is $c_1 = 10^{-4}$ and $c_2 = 0.9$. NLCG is often implemented with a somewhat stricter line search, along the lines of $c_1 = 10^{-4}$ and $c_2 = 0.1$. With the NLCG bracketing line search described in Section 5.1, however, we find it more cost effective to use $c_1 = 10^{-4}$ and $c_2 = 0.9$ to reduce the number of gradient evaluations, the bracketing and interpolation requirements being enough, it seems, to ensure a sufficiently accurate step length.

Downloaded from https://academic.oup.com/gji/article/206/3/1864/2583505 by Princeton University user on 11 September 2020