A projected Hessian matrix for full waveform inversion

Yong Ma* and Dave Hale, Center for Wave Phenomena, Colorado School of Mines

SUMMARY

A Hessian matrix in full waveform inversion (FWI) is difficult to compute directly because of high computational cost and an especially large memory requirement. Therefore, Newton-like methods are rarely feasible in realistic large-size FWI problems. We modify the quasi-Newton BFGS method to use a projected Hessian matrix that reduces both the computational cost and memory required, thereby making a quasi-Newton solution to FWI feasible.

INTRODUCTION

Full waveform inversion (FWI) (Tarantola, 1984) is usually formulated as an optimization problem, in which we minimize a nonlinear objective function $E: \mathbb{R}^N \to \mathbb{R}$,

$$\min_{\mathbf{m} \in \mathbb{R}^N} E(\mathbf{m}), \quad (1)$$

where $\mathbf{m}$ denotes $N$ parameters, such as seismic velocities, for an earth model. In FWI, $E$ often takes a least-squares form: $E(\mathbf{m}) = \frac{1}{2} \| \mathbf{d} - \mathbf{F} (\mathbf{m}) \|^2$, where $\| \cdot \|$ denotes an L2 norm, $\mathbf{d}$ denotes the recorded data, and $\mathbf{F}$ is a forward data-synthesizing operator, a nonlinear function of the model $\mathbf{m}$.

We consider only iterative methods, such as Newton’s method and quasi-Newton methods, for solutions to FWI. In the $i$th iteration of such methods, we update the model $\mathbf{m}$ in the direction of a vector $\mathbf{p}_i$:

$$\mathbf{m}_{i+1} = \mathbf{m}_i + \alpha_i \mathbf{p}_i, \quad (2)$$

for some step length $\alpha_i$. Newton’s and quasi-Newton methods differ in the way in which they compute and use $\mathbf{p}_i$.

In Newton’s method, we ignore the higher-order ($> 2$) terms in the Taylor series of $E(\mathbf{m})$:

$$E(\mathbf{m}_i + \delta \mathbf{m}) \approx E(\mathbf{m}_i) + \delta \mathbf{m}^T \mathbf{g}_i + \frac{1}{2} \delta \mathbf{m}^T \mathbf{H}_i \delta \mathbf{m}_i, \quad (3)$$

and minimize this approximated $E(\mathbf{m})$ by solving

$$\mathbf{H}_i \delta \mathbf{m}_i = -\mathbf{g}_i, \quad (4)$$

where the gradient $\mathbf{g}_i \equiv \frac{\partial E}{\partial \mathbf{m}_i}$, and the Hessian matrix $\mathbf{H}_i \equiv \frac{\partial^2 E}{\partial \mathbf{m}_i^2}$. Therefore, the update direction in Newton’s method is simply

$$\mathbf{p}_i = -\mathbf{H}_i^{-1} \mathbf{g}_i, \quad (5)$$

and step length $\alpha_i = 1$.

Nevertheless, Newton’s method is suitable only for solving small- or medium-size optimization problems (Pratt et al., 1998; Virieux and Operto, 2009), in which the number $N$ of model parameters ranges from hundreds to thousands. For models with a large number $N$ of parameters, high costs for computing the Hessian $\mathbf{H}_i$ prevent the application of Newton-like methods. If, say, $N = 500000$ in 2D FWI, although we can use an efficient way in Pratt et al. (1998), we must solve the forward problem $\mathbf{F}(\mathbf{m})$ for 500000 times to compute the Hessian in every iteration of FWI. Furthermore, the memory required to store this Hessian matrix in single precision is 1 terabyte, or about 1/2 terabyte considering symmetry in the Hessian.

Quasi-Newton methods do not compute the Hessian matrix explicitly, but instead iteratively update a Hessian approximation. The BFGS method (Broyden, 1970; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970) is by far the most popular way (Nocedal and Wright, 2000) to update the Hessian matrix:

$$\mathbf{H}_{i+1} = \mathbf{H}_i + \frac{y_i y_i^T}{y_i^T \delta \mathbf{m}} - \frac{\mathbf{H}_i \delta \mathbf{m}_i (\mathbf{H}_i \delta \mathbf{m}_i)^T}{\delta \mathbf{m}_i^T \mathbf{H}_i \delta \mathbf{m}_i}, \quad (6)$$

where $y_i = \mathbf{g}_{i+1} - \mathbf{g}_i$, $\delta \mathbf{m}_i = \mathbf{m}_{i+1} - \mathbf{m}_i$. Although the BFGS method reduces the computation time required to approximate a Hessian matrix, it does not reduce the $O(N^3)$ computation time required to use the Hessian matrix to update models (equation 4), nor does it reduce the $O(N^2)$ memory required to store Hessian approximations.

The only way to reduce these costs is to reduce the number $N$ of model parameters. For this purpose, we introduce a projected Hessian matrix in a sparse model space. Tests of our projected Hessian on the Marmousi II model suggest that quasi-Newton FWI may be promising in practical applications.

PROJECTED HESSIAN MATRIX

In an approach similar to that used in subspace methods (Kennett et al., 1988; Oldenburg et al., 1993) and alternative parameterizations (Pratt et al., 1998, Appendix A), we construct a finely- and uniformly-sampled (dense) model $\mathbf{m}$ from a sparse model $\mathbf{s}$ that contains a much smaller number $n < < N$ of model parameters:

$$\mathbf{m} = \mathbf{R} \mathbf{s}, \quad (7)$$

where $\mathbf{R}$ is an interpolation operator that interpolates model parameters from the sparse model $\mathbf{s}$ to the dense model $\mathbf{m}$.

FWI is then reformulated as a new sparse optimization problem (Pratt et al., 1998, Appendix A; Ma et al., 2010), in which we minimize a new nonlinear objective function $E: \mathbb{R}^n \to \mathbb{R}$,

$$\min_{\mathbf{s} \in \mathbb{R}^n} E(\mathbf{R} \mathbf{s}), \quad (8)$$

In the optimization problem posed in equation 8, model parameters are not determined independently as in equation 1. Instead, the $n$ sparse parameters in $\mathbf{s}$ constrain the other $N - n$ parameters in $\mathbf{m}$. Therefore, equation 8 is equivalent to a linearly constrained optimization, with $N - n$ constraints. In the context of constrained optimization, the projected Hessian matrix is suggested by Gill et al. (1981) and several other authors.
Rewrite equation 3 as
\[
E (\mathbf{R}s_0 + \mathbf{R}\delta s_i) \approx E (\mathbf{R}s_i) + \frac{1}{2} \delta s_i^T \mathbf{R}^T \mathbf{R} \delta s_i + \frac{1}{2} \mathbf{R}^T \mathbf{H}_i \mathbf{R} \delta s_i, \quad (9)
\]
where we refer to \( \mathbf{R}^T \mathbf{H}_i \mathbf{R} \) as a *projected Hessian* matrix (Gill et al., 1981), and \( \mathbf{R}^T \mathbf{g}_i \) as a projected gradient. Here \( \mathbf{R}^T \) denotes the adjoint of the interpolation operator \( \mathbf{R} \).

We then minimize this approximated \( E \) by solving a set of \( n \) linear equations:
\[
\mathbf{R}^T \mathbf{H}_i \mathbf{R} \delta s_i = -\mathbf{R}^T \mathbf{g}_i, \quad (10)
\]
with a solution for the \( n \) unknowns
\[
\delta s_i = -\left( \mathbf{R}^T \mathbf{H}_i \mathbf{R} \right)^{-1} \mathbf{R}^T \mathbf{g}_i. \quad (11)
\]
Therefore, in Newton’s method the update direction vector becomes
\[
\mathbf{p}_i = -\left( \mathbf{R}^T \mathbf{H}_i \mathbf{R} \right)^{-1} \mathbf{R}^T \mathbf{g}_i, \quad (12)
\]
for a step length \( \alpha_i = 1 \).

The projected Hessian \( \mathbf{R}^T \mathbf{H}_i \mathbf{R} \) is a \( n \times n \) symmetric matrix. Figure 1 describes this projected Hessian in a schematic fashion: the tall and thin rectangle denotes the interpolation operator \( \mathbf{R} \), the short and fat rectangle denotes the adjoint operator \( \mathbf{R}^T \), and the large and small squares denote the original and projected Hessian matrices, respectively. Figure 1 illustrates how memory consumption is reduced from \( O(N^2) \) to \( O(n^2) \). Computation is likewise reduced from \( O(N^3) \) to \( O(n^3) \).

### Projected BFGS method

A projected Hessian matrix can be approximated iteratively with an approach similar to the classic BFGS method. Apply the interpolation operator \( \mathbf{R} \) and its adjoint operator \( \mathbf{R}^T \) to both sides of equation 6 to obtain
\[
\mathbf{R}^T \mathbf{H}_{i+1} \mathbf{R} = \mathbf{R}^T \mathbf{H}_i \mathbf{R} + \frac{\mathbf{R}^T \mathbf{y}_i \mathbf{R} \mathbf{R}^T \mathbf{H}_i \mathbf{m}_i \delta \mathbf{m}_i}{\mathbf{y}_i^T \delta \mathbf{m}_i} - \frac{\mathbf{R}^T \mathbf{H}_i \mathbf{m}_i \delta \mathbf{m}_i}{\delta \mathbf{m}_i^T \mathbf{H}_i \delta \mathbf{m}_i}. \quad (13)
\]

Now simplify equation 13 by defining \( \tilde{\mathbf{H}}_i = \mathbf{R}^T \mathbf{H}_i \mathbf{R}, \tilde{\mathbf{g}}_i = \mathbf{R}^T \mathbf{g}_i \), and \( \tilde{\mathbf{y}}_i = \mathbf{R}^T \mathbf{y}_i = \tilde{\mathbf{g}}_{i+1} - \tilde{\mathbf{g}}_i \) to obtain an update formula for the projected BFGS method:
\[
\tilde{\mathbf{H}}_{i+1} = \tilde{\mathbf{H}}_i + \frac{\tilde{\mathbf{y}}_i \tilde{\mathbf{y}}_i^T}{\tilde{\mathbf{y}}_i^T \delta s_i} - \frac{\tilde{\mathbf{H}}_i \delta s_i (\tilde{\mathbf{H}}_i \delta s_i)^T}{\delta s_i^T \tilde{\mathbf{H}}_i \delta s_i}. \quad (14)
\]

### EXAMPLES

We test our projected BFGS method on the Marmousi II model. Figure 2a shows the true model \( \mathbf{m}_0 \), and Figure 2b shows the initial model \( \mathbf{m}_0 \), a highly smoothed version of the true model. We use 11 shots uniformly distributed on the surface, and a 15Hz Ricker wavelet as the source for simulating wavefields. The source and receiver intervals are 0.76 km and 0.024 km, respectively. In this example, the dense model space has \( N = 391 \times 1151 \) parameters. Therefore, either computation or storage of the Hessian matrix for the dense model is infeasible.
Structured constrained sample selection

A properly constructed sparse model $s$ is essential for implementing the projected BFGS method. As suggested by Ma et al. (2011), we construct such a sparse model $s$ by using a structurally constrained sample selection scheme. This selection method considers structures apparent in migrated images. Figure 3a displays a metric tensor field (illustrated by ellipses) computed for a migrated image. As we can see, orientations and shapes of the ellipses in Figure 3a conform to the imaged structure.

Sparse samples should be representative, so that image-guided interpolation can reconstruct an accurate dense model $m$ from a sparse model $s$. In general, we must

- **locate samples between reflectors.** We should especially avoid putting samples on reflectors.
- **locate samples along structural features.** To reduce redundancy, we should place fewer samples along structural features than across them. Moreover, we should put more samples in structurally complex areas than in simple areas.

Figure 3b shows a total of 165 scattered sample locations. The chosen locations together with corresponding values at these locations comprise a sparse model space $s$ that will be used in the projected BFGS method.

Projected Hessian and inverse

In our projected BFGS method, we employ image-guided interpolation (IGI) (Hale, 2009) as the operator $R$ and the adjoint of image-guided interpolation (Ma et al., 2010) as the operator $R^T$. IGI interpolates values in the sparse model $s$ to a uniformly sampled dense model $m$, and the interpolant makes good geological sense because it accounts for structures in the model.

The Hessian update $\tilde{H}_1 - \tilde{H}_0$ in the first iteration is shown in Figure 4a. As we can see, the projected BFGS method adds significant off-diagonal components to the initial Hessian $\tilde{H}_0 = I$. Because the line search satisfies the strong Wolfe conditions (Nocedal and Wright, 2000), the projected Hessian is symmetric positive-definite (SPD). Therefore, the inverse of a projected Hessian matrix exists. Figure 4b shows the inverse Hessian $\tilde{H}_1^{-1}$. 

---

**Figure 3**: Metric tensor fields depicted by ellipses (a) and structurally constrained sample selections (b). A total of 165 samples are chosen for our projected BFGS method.

**Figure 4**: The Hessian update $\tilde{H}_1 - \tilde{H}_0$ in the 1st iteration (a) and the corresponding inverse Hessian $\tilde{H}_1^{-1}$ (b). An initial Hessian $\tilde{H}_0 = I$ is used for the projected BFGS method.
Projected quasi-Newton FWI
The projected BFGS method updates the model \( m_i \) in each iteration, and therefore the projected BFGS method can be directly used in quasi-Newton solutions to FWI. Figure 5a and 5b shows the update direction of the conjugate-gradient and the quasi-Newton methods in the 1st iteration, respectively. Compared with Figure 5a, the quasi-Newton update direction (Figure 5b) contains significant low wavenumbers. Furthermore, the inverse projected Hessian \( \tilde{H}_i^{-1} \) works as a filter applied to the gradient. As a consequence, the amplitudes of the update direction (Figure 5b) are more balanced.

Figures 5c and 5d show estimated models in the 10th iteration of the conjugate-gradient and quasi-Newton methods, respectively. Figure 6 shows the data misfit function of the quasi-Newton FWI, which shows a faster convergence rate than does the conjugate-gradient method.

CONCLUSION
In this paper we propose a projected BFGS method to iteratively approximate the Hessian matrix in FWI, thereby reducing both computation time and required memory. The projected BFGS method can be used to perform FWI using a quasi-Newton method. As demonstrated by the examples above, quasi-Newton FWI converges in fewer iterations.

Compared with the conjugate gradient method, the primary disadvantage of our projected BFGS method is the computational cost of a single iteration, which is relatively high because the line search must satisfy the Wolfe conditions. Therefore, a further investigation of the coefficients in the Wolfe conditions is necessary.

ACKNOWLEDGMENT
This work is sponsored by a research agreement between ConocoPhillips and Colorado School of Mines (SST-20090254-SRA).
Projected Hessian

REFERENCES


