IMPLEMENTATION OF THE GAUSS-NEWTON METHOD FOR FREQUENCY-DOMAIN FULL WAVEFORM INVERSION USING A LOGARITHMIC OBJECTIVE FUNCTION

SUKJOON PYUN[1], WOOHYUN SON[2] and CHANGSOO SHIN[2]

[1] *Dept. of Energy Resources Engineering, Inha University, 253 Yonghyun-Dong, Nam-Gu, Incheon 402-751, South Korea. pyunsj@inha.ac.kr*
[2] *Dept. of Energy Systems Engineering, Seoul National University, 599 Gwanak-ro, Gwanak-gu, Seoul 151-744, South Korea.*

(Received July 6, 2010; revised version accepted January 4, 2011)

ABSTRACT

Pyun, S., Son, W. and Shin, C., 2011. Implementation of the Gauss-Newton method for frequency-domain full waveform inversion using a logarithmic objective function. *Journal of Seismic Exploration*, 20: 193-206.

The use of a logarithmic misfit function has proven to be advantageous for full waveform inversion of field data in noisy environments. However, few efforts have been carried out to define an efficient and effective updating procedure. In this paper, we utilize an approximated Gauss-Newton method, which is called CGLS or Gauss-Newton-CG method, using a logarithmic misfit for the frequency-domain full waveform inversion. The Gauss-Newton inversion via a conjugate gradient algorithm is an efficient method because the Hessian matrix does not need to be explicitly calculated. In this method, the logarithmic objective function is not applied even though it is robust against noise and able to resolve strongly damped wavefields such as the Laplace-domain wavefields. In this context, we use the logarithmic objective function to formulate the normal equation for implementation in the Gauss-Newton inversion. This results in an unstable inverse problem, which is mitigated by applying a heuristic filtering method to the gradient direction and the Hessian matrix. A simple damping term was added to the approximated Hessian matrix to stabilize the inversion. To verify our algorithm, we compare the proposed Gauss-Newton method with the gradient method. The Marmousi model and the SEG/EAGE salt model were used for this comparison. The gradient method produces distorted images in the deeper parts of the velocity model, while our method gives improved results and shows noticeable improvement for recovering the structure beneath the salt layer in the SEG/EAGE model.

KEY WORDS: Gauss-Newton method, conjugate gradient, waveform inversion, frequency-domain, logarithmic wavefield.

0963-0651/11/$5.00

INTRODUCTION

The basic theory of the seismic full waveform inversion (FWI) was established a long time ago. However, it has not been used until recently due to computational limitations. After Tarantola (1984) suggested the efficient gradient calculation algorithm based on the adjoint property of the wave equation, the seismic full waveform inversion became a feasible approach. The detailed history of FWI can be found in Virieux and Operto (2009). Although Tarantola's method is efficient since it does not calculate the Jacobian directly, it has limits inherited from the gradient method. In order to properly scale the gradient direction and to avoid a prohibitively large memory requirement and computation size, an advanced method based on the conjugate gradient algorithm was suggested by many researchers. This method is called the CGLS (conjugate gradient least squares) method (Golub and Loan, 1996; Hu et al., 2009) or Newton-CG method (Epanomeritakis et al., 2008). In this method, the Jacobian matrix does not need to be explicitly calculated or saved in computer memory. For the magnetotelluric problem, Mackie and Madden (1993) proposed the linear conjugate gradient relaxation method for solving a least-squares inverse problem. They used the linear conjugate gradient method to solve the normal equation without forming the Hessian and Jacobian matrices. The same conjugate gradient method for the magnetotelluric inversion problem was also applied to the seismic full waveform inversion (Chen et al., 2007; Epanomeritakis et al., 2008, Hu et al., 2009). Through this approach, we can implement the Gauss-Newton inversion scheme for the full waveform inversion.

However, there has been no attempt to exploit the logarithmic objective function (Shin and Min, 2006) to implement the Gauss-Newton or quasi-Newton method in the published literature. This is partly because the logarithmic objective function leads to a complicated and unstable sensitivity matrix. As a result, it is not possible to apply advanced regularization techniques to the logarithmic objective function, despite the various applications of the logarithmic misfit. The logarithmic objective function has been shown to be robust to noise and was successfully applied to field data (Shin and Min, 2006). Furthermore, the logarithmic objective function is needed to resolve the strongly damped wavefields in the Laplace-domain or the Laplace-Fourier-domain waveform inversion (Shin and Cha, 2008, 2009; Shin et al., 2010). Thus, we need to develop a more sophisticated optimization method for the logarithmic objective function. In this paper, we focus on the implementation of the Gauss-Newton method for the full waveform inversion using a logarithmic wavefield. Although more recent techniques such as BFGS (Broyden-Fletcher-Goldfarb-Shanno) and L-BFGS (Nocedal, 1980) have already been developed and applied to the FWI (Sambridge et al., 1991; Brossier et al., 2009), we limit our scope to the Gauss-Newton method to concentrate on the successful implementation of the logarithmic objective function. Since the logarithmic objective function has an instability problem, the logarithmic wavefield should be heuristically

manipulated for both the gradient direction and the approximated Hessian. We introduce the empirical filtering technique and regularization method for the stable inversion. We verify our algorithm through the inversion examples of the Marmousi model and the SEG/EAGE salt model. We also compare the inversion results using the proposed method with those using the gradient method to show the advantages of our algorithm.

THE GAUSS-NEWTON METHOD USING THE LOGARITHMIC OBJECTIVE FUNCTION

In the frequency-domain waveform inversion, the logarithmic objective function can be defined as

$$E = \mathbf{e}^T\mathbf{e}^* \quad , \tag{1}$$

where

$$\mathbf{e} = [\ln(d_1/u_1) \quad \ln(d_2/u_2) \quad \ldots \quad \ln(d_r/u_r)]^T \quad , \tag{2}$$

and d is the observed wavefield, u is the modeled wavefield, the subscript denotes the receiver index, r is the last receiver index, and the superscript * denotes the complex conjugate. In order to obtain the parameter vector to minimize the objective function in eq. (1), we choose the Gauss-Newton method to solve the linearized minimization problem iteratively.

Expanding the modeled wavefields u to the first order term of a Taylor's series gives

$$E = [\mathbf{e}_0 - \mathbf{J}\Delta\mathbf{p}]^T[\mathbf{e}_0 - \mathbf{J}\Delta\mathbf{p}]^* \quad , \tag{3}$$

where

$$\mathbf{J} = \begin{bmatrix} \frac{1}{u_1}\frac{\partial u_1}{\partial p_1} & \frac{1}{u_1}\frac{\partial u_1}{\partial p_2} & \cdots & \frac{1}{u_1}\frac{\partial u_1}{\partial p_m} \\ \frac{1}{u_2}\frac{\partial u_2}{\partial p_1} & \frac{1}{u_2}\frac{\partial u_2}{\partial p_2} & \cdots & \frac{1}{u_2}\frac{\partial u_2}{\partial p_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{u_r}\frac{\partial u_r}{\partial p_1} & \frac{1}{u_r}\frac{\partial u_r}{\partial p_2} & \cdots & \frac{1}{u_r}\frac{\partial u_r}{\partial p_m} \end{bmatrix} , \tag{4}$$

$$\Delta\mathbf{p} = [\Delta p_1 \quad \Delta p_2 \quad \ldots \quad \Delta p_m]^T \quad . \tag{5}$$

$\mathbf{e}_0$ is the logarithmic residual wavefield vector using an initial velocity model, $\mathbf{p}$ is the model parameter vector and m is the number of model parameters.

J is the Jacobian matrix, representing the sensitivity of wavefields with respect to the model parameters, and $\Delta\mathbf{p}$ is the perturbation to the parameter vector. Based on the least-squares principle, we differentiate the objective function in eq. (3) with respect to each element of $\Delta\mathbf{p}$ and let it be zero. Then, we arrive at

$$\mathbf{J}^{\mathrm{T}}\mathbf{J}^{*}\Delta\mathbf{p} = \mathbf{J}^{\mathrm{T}}\mathbf{e}_{0}^{*} \quad . \tag{6}$$

We can obtain the perturbation vector of model parameters by solving the least-squares problem in eq. (6). Since the full waveform inversion is a nonlinear inverse problem, eq. (6) is recalculated and solved for each iteration and the model parameter is iteratively updated.

IMPLEMENTATION OF THE GAUSS-NEWTON-CG METHOD

In order to show how the Gauss-Newton inversion using the logarithmic objective function can be formulated as CGLS, we first rewrite eq. (6) as

$$\begin{bmatrix} \frac{1}{u_1}\frac{\partial u_1}{\partial p_1} & \frac{1}{u_1}\frac{\partial u_1}{\partial p_2} & \cdots & \frac{1}{u_1}\frac{\partial u_1}{\partial p_m} \\ \frac{1}{u_2}\frac{\partial u_2}{\partial p_1} & \frac{1}{u_2}\frac{\partial u_2}{\partial p_2} & \cdots & \frac{1}{u_2}\frac{\partial u_2}{\partial p_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{u_r}\frac{\partial u_r}{\partial p_1} & \frac{1}{u_r}\frac{\partial u_r}{\partial p_2} & \cdots & \frac{1}{u_r}\frac{\partial u_r}{\partial p_m} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \frac{1}{u_1}\frac{\partial u_1}{\partial p_1} & \frac{1}{u_1}\frac{\partial u_1}{\partial p_2} & \cdots & \frac{1}{u_1}\frac{\partial u_1}{\partial p_m} \\ \frac{1}{u_2}\frac{\partial u_2}{\partial p_1} & \frac{1}{u_2}\frac{\partial u_2}{\partial p_2} & \cdots & \frac{1}{u_2}\frac{\partial u_2}{\partial p_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{u_r}\frac{\partial u_r}{\partial p_1} & \frac{1}{u_r}\frac{\partial u_r}{\partial p_2} & \cdots & \frac{1}{u_r}\frac{\partial u_r}{\partial p_m} \end{bmatrix}^{*} \begin{bmatrix} \Delta p_1 \\ \Delta p_2 \\ \vdots \\ \Delta p_m \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{u_1}\frac{\partial u_1}{\partial p_1} & \frac{1}{u_1}\frac{\partial u_1}{\partial p_2} & \cdots & \frac{1}{u_1}\frac{\partial u_1}{\partial p_m} \\ \frac{1}{u_2}\frac{\partial u_2}{\partial p_1} & \frac{1}{u_2}\frac{\partial u_2}{\partial p_2} & \cdots & \frac{1}{u_2}\frac{\partial u_2}{\partial p_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{u_r}\frac{\partial u_r}{\partial p_1} & \frac{1}{u_r}\frac{\partial u_r}{\partial p_2} & \cdots & \frac{1}{u_r}\frac{\partial u_r}{\partial p_m} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \ln\frac{d_1}{u_1} \\ \ln\frac{d_2}{u_2} \\ \vdots \\ \ln\frac{d_r}{u_r} \end{bmatrix}^{*} . \tag{7}$$

Then, we decompose the Jacobian matrix in eq. (7), finding

$$\begin{bmatrix} \frac{\partial u_1}{\partial p_1} & \frac{\partial u_1}{\partial p_2} & \cdots & \frac{\partial u_1}{\partial p_m} \\ \frac{\partial u_2}{\partial p_1} & \frac{\partial u_2}{\partial p_2} & \cdots & \frac{\partial u_2}{\partial p_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial u_r}{\partial p_1} & \frac{\partial u_r}{\partial p_2} & \cdots & \frac{\partial u_r}{\partial p_m} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \frac{1}{u_1} & 0 & \cdots & 0 \\ 0 & \frac{1}{u_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{u_r} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \frac{1}{u_1} & 0 & \cdots & 0 \\ 0 & \frac{1}{u_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{u_r} \end{bmatrix}^{*} \begin{bmatrix} \frac{\partial u_1}{\partial p_1} & \frac{\partial u_1}{\partial p_2} & \cdots & \frac{\partial u_1}{\partial p_m} \\ \frac{\partial u_2}{\partial p_1} & \frac{\partial u_2}{\partial p_2} & \cdots & \frac{\partial u_2}{\partial p_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial u_r}{\partial p_1} & \frac{\partial u_r}{\partial p_2} & \cdots & \frac{\partial u_r}{\partial p_m} \end{bmatrix}^{*} \begin{bmatrix} \Delta p_1 \\ \Delta p_2 \\ \vdots \\ \Delta p_m \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\partial u_1}{\partial p_1} & \frac{\partial u_1}{\partial p_2} & \cdots & \frac{\partial u_1}{\partial p_m} \\ \frac{\partial u_2}{\partial p_1} & \frac{\partial u_2}{\partial p_2} & \cdots & \frac{\partial u_2}{\partial p_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial u_r}{\partial p_1} & \frac{\partial u_r}{\partial p_2} & \cdots & \frac{\partial u_r}{\partial p_m} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \frac{1}{u_1} & 0 & \cdots & 0 \\ 0 & \frac{1}{u_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{u_r} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \ln \frac{d_1}{u_1} \\ \ln \frac{d_2}{u_2} \\ \vdots \\ \ln \frac{d_r}{u_r} \end{bmatrix}^{*} . \tag{8}$$

Rearranging eq. (8) results in

$$\begin{bmatrix} \frac{\partial u_1}{\partial p_1} & \frac{\partial u_1}{\partial p_2} & \cdots & \frac{\partial u_1}{\partial p_m} \\ \frac{\partial u_2}{\partial p_1} & \frac{\partial u_2}{\partial p_2} & \cdots & \frac{\partial u_2}{\partial p_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial u_r}{\partial p_1} & \frac{\partial u_r}{\partial p_2} & \cdots & \frac{\partial u_r}{\partial p_m} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \frac{1}{u_1 u_1^*} & 0 & \cdots & 0 \\ 0 & \frac{1}{u_2 u_2^*} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{u_r u_r^*} \end{bmatrix} \begin{bmatrix} \frac{\partial u_1}{\partial p_1} & \frac{\partial u_1}{\partial p_2} & \cdots & \frac{\partial u_1}{\partial p_m} \\ \frac{\partial u_2}{\partial p_1} & \frac{\partial u_2}{\partial p_2} & \cdots & \frac{\partial u_2}{\partial p_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial u_r}{\partial p_1} & \frac{\partial u_r}{\partial p_2} & \cdots & \frac{\partial u_r}{\partial p_m} \end{bmatrix}^{*} \begin{bmatrix} \Delta p_1 \\ \Delta p_2 \\ \vdots \\ \Delta p_m \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\partial u_1}{\partial p_1} & \frac{\partial u_1}{\partial p_2} & \cdots & \frac{\partial u_1}{\partial p_m} \\ \frac{\partial u_2}{\partial p_1} & \frac{\partial u_2}{\partial p_2} & \cdots & \frac{\partial u_2}{\partial p_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial u_r}{\partial p_1} & \frac{\partial u_r}{\partial p_2} & \cdots & \frac{\partial u_r}{\partial p_m} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \frac{1}{u_1}\left(\ln \frac{d_1}{u_1}\right)^{*} \\ \frac{1}{u_2}\left(\ln \frac{d_2}{u_2}\right)^{*} \\ \vdots \\ \frac{1}{u_r}\left(\ln \frac{d_r}{u_r}\right)^{*} \end{bmatrix} . \tag{9}$$

In order to obtain the Jacobian matrix in eq. (9), we define the wave equation modeling as a set of linear algebraic equations (Marfurt, 1984; Pratt et al., 1998):

$$\begin{bmatrix} \\ \mathbf{S}_0 \\ \\ \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} \\ \mathbf{f} \\ \\ \end{bmatrix}, \tag{10}$$

where $\mathbf{S}_0$ is the complex impedance matrix using an initial velocity model, $\mathbf{f}$ is the source vector, and the subscript n is the total number of the grid points of the model. If we differentiate both sides of eq. (10) with respect to p, we have

$$\begin{bmatrix} \\ \mathbf{S}_0 \\ \\ \end{bmatrix} \begin{bmatrix} \frac{\partial u_1}{\partial p_1} \\ \frac{\partial u_2}{\partial p_1} \\ \vdots \\ \frac{\partial u_n}{\partial p_1} \end{bmatrix} = -\frac{\partial}{\partial p_1} \begin{bmatrix} \\ \mathbf{S}_0 \\ \\ \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}. \tag{11}$$

Since we can assume that the right hand side of eq. (11) is a type of source vector, we define the right hand side vector of eq. (11) as a virtual source $\mathbf{v}_1$. By doing the same process for the other parameters from p_2 to p_m, we define the Jacobian matrix as

$$\begin{bmatrix} \frac{\partial u_1}{\partial p_1} & \frac{\partial u_1}{\partial p_2} & \cdots & \frac{\partial u_1}{\partial p_m} \\ \frac{\partial u_2}{\partial p_1} & \frac{\partial u_2}{\partial p_2} & \cdots & \frac{\partial u_2}{\partial p_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial u_n}{\partial p_1} & \frac{\partial u_n}{\partial p_2} & \cdots & \frac{\partial u_n}{\partial p_m} \end{bmatrix} = \begin{bmatrix} & \mathbf{S}_0 & \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & & \mathbf{v}_m \end{bmatrix}. \tag{12}$$

Using eq. (12), we can recalculate the Jacobian matrix as

$$\mathbf{J} = \mathbf{A}\mathbf{S}_0^{-1}\mathbf{V} \ , \tag{13}$$

where $\mathbf{A}$ is the $r \times n$ matrix to limit the elements within the receiver points as follows:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & \cdots & 0 \end{bmatrix}. \tag{14}$$

The matrix $\mathbf{V}$ consists of the virtual source vectors

$$\mathbf{V} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_m \end{bmatrix}. \tag{15}$$

Substituting eq. (13) into eq. (9) results in

$$(\mathbf{A}\mathbf{S}_0^{-1}\mathbf{V})^{\mathrm{T}}\mathbf{U}(\mathbf{A}\mathbf{S}_0^{-1}\mathbf{V})^{*}\Delta\mathbf{p} = (\mathbf{A}\mathbf{S}_0^{-1}\mathbf{V})^{\mathrm{T}}\mathbf{e}_{\mathrm{r}} \ , \tag{16}$$

where

$$\mathbf{U} = \begin{bmatrix} \frac{1}{u_1 u_1^{*}} & 0 & \cdots & 0 \\ 0 & \frac{1}{u_2 u_2^{*}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{u_r u_r^{*}} \end{bmatrix} \ , \tag{17}$$

and

$$\mathbf{e}_r = \begin{bmatrix} \frac{1}{u_1}\left(\ln\frac{d_1}{u_1}\right)^* \\ \frac{1}{u_2}\left(\ln\frac{d_2}{u_2}\right)^* \\ \vdots \\ \frac{1}{u_r}\left(\ln\frac{d_r}{u_r}\right)^* \end{bmatrix} \quad . \tag{18}$$

Since the impedance matrix $\mathbf{S}_0$ is symmetric, eq. (16) is expressed as

$$\mathbf{V}^T\mathbf{S}_0^{-1}\mathbf{A}^T\mathbf{U}\mathbf{A}(\mathbf{S}_0^{-1}\mathbf{V})^*\Delta\mathbf{p} = \mathbf{V}^T\mathbf{S}_0^{-1}\mathbf{A}^T\mathbf{e}_r \quad . \tag{19}$$

Eq. (19) can be rewritten as

$$\mathbf{V}^T\mathbf{S}_0^{-1}\mathbf{U}_a(\mathbf{S}_0^{-1}\mathbf{V})^*\Delta\mathbf{p} = \mathbf{V}^T\mathbf{S}_0^{-1}\mathbf{e}_a \quad , \tag{20}$$

where the matrix $\mathbf{U}_a$ has a size of n $\times$ n and is expressed as

$$\mathbf{U}_a = \begin{bmatrix} \mathbf{U} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad . \tag{21}$$

The vector $\mathbf{e}_a$ has n elements and is expressed as

$$\mathbf{e}_a = \begin{bmatrix} \mathbf{e}_r \\ \mathbf{0} \end{bmatrix} \quad . \tag{22}$$

We can rewrite eq. (20) in the following simple form

$$\mathbf{H}\Delta\mathbf{p} = \mathbf{g} \quad , \tag{23}$$

where $\mathbf{H} = \mathbf{V}^T\mathbf{S}_0^{-1}\mathbf{U}_a(\mathbf{S}_0^{-1}\mathbf{V})^*$ and $\mathbf{g} = \mathbf{V}^T\mathbf{S}_0^{-1}\mathbf{e}_a$. The perturbation $\Delta\mathbf{p}$ can be obtained by solving eq. (23) using the conjugate gradient method (Golub and Loan, 1996).

MODIFICATION OF THE GRADIENT DIRECTION AND THE JACOBIAN

When we construct the normal equation shown in eq. (20), we encounter a troublesome task stemming from division operations. As shown in eqs. (17) and (18), both the Jacobian and the gradient direction should be divided by the initial wavefield, causing the solution of the normal equation to blow up. This instability originates from the logarithmic objective function. However, there has been no overt discussion about this problem since Shin and Min (2006)

suggested the logarithmic objective function. In this paper, we use a heuristic method to avoid the instability by setting a threshold and setting elements of the gradient direction and Jacobian matrix with amplitudes less than this threshold to zero. Based on trial and error, we chose a threshold of 1.e-32. In the case of the Jacobian, we removed the division operation from the initial wavefield instead of setting the element to zero. Although this heuristic manipulation can lead to an inaccurate solution, the inaccuracy is negligible and it makes the solution more stable.

DETERMINATION OF THE DAMPING FACTOR

In general, a damping factor or regularization term is added to the Hessian matrix because the matrix is ill-posed. To determine this damping factor, we need to know the magnitudes of the diagonal elements of the Hessian matrix. However, we do not calculate the Hessian matrix explicitly when we use the conjugate gradient method. Therefore, we need another approach to determine the proper damping factor. If we have the Hessian matrix, an arbitrary element of the matrix can be calculated as

$$H_{ij} = \mathbf{b}_i^T \mathbf{H} \mathbf{b}_j \quad , \tag{24}$$

where $\mathbf{b}_i$ is a unit vector whose i-th element is unity and other elements are zero. Using eq. (20), we can calculate the i-th diagonal element of the Hessian matrix as follows:

$$H_{ii} = \mathbf{b}_i^T \mathbf{V}^T \mathbf{S}_0^{-1} \mathbf{U}_a (\mathbf{S}_0^{-1} \mathbf{V}) * \mathbf{b}_i \quad . \tag{25}$$

After calculating a diagonal element of the Hessian matrix using eq. (25), we can determine the magnitude of the damping factor λ and add the damping factor to the Hessian matrix implicitly as follows

$$(\mathbf{H} + \lambda \mathbf{I}) \Delta \mathbf{p} = \mathbf{H} \Delta \mathbf{p} + \lambda \Delta \mathbf{p} \quad . \tag{26}$$

Since the Hessian matrix does not change during the CG iterations for a single inversion step, we only need to compute the damping factor just once per iteration of the inversion.

NUMERICAL EXAMPLES

We tested our algorithm using the Marmousi model and the SEG/EAGE salt model. In both cases, we use a finite-element modeling technique in the frequency domain to generate a synthetic data set. First, we selected 50 frequencies, ranging from 0.3 to 15 Hz, to perform the frequency-domain full

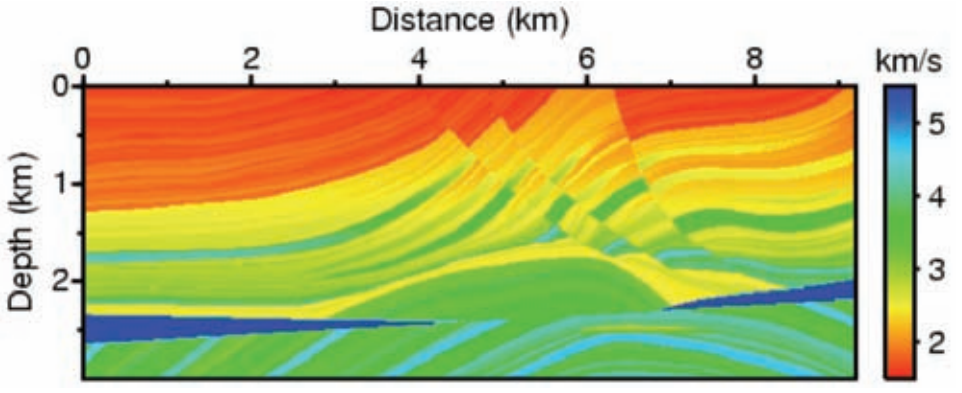
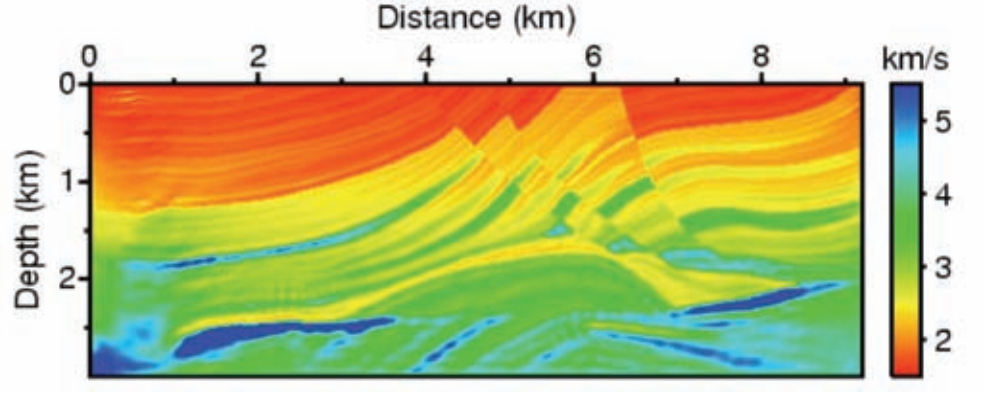


Fig. 1. The Marmousi velocity model.

a)



b)

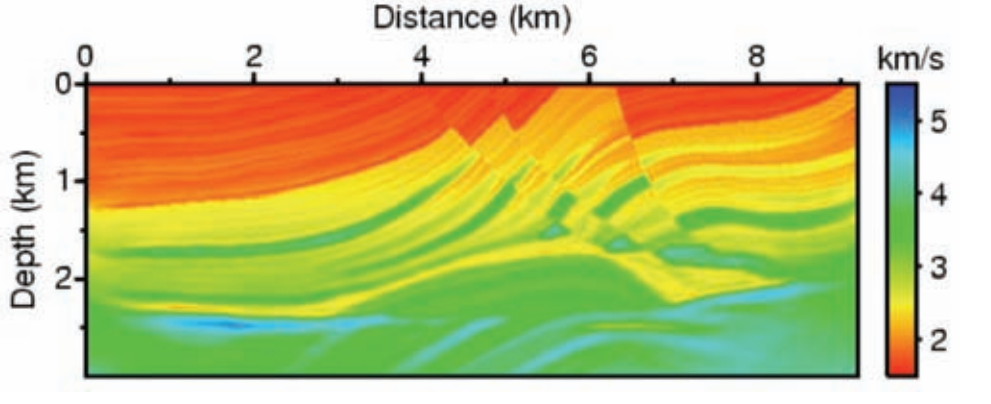


Fig. 2. Inverted velocity model using data based on the Marmousi velocity model obtained at (a) the 500-th iteration of the gradient method and (b) the 300-th iteration of the Gauss-Newton method.

waveform inversion for the Marmousi model (Fig. 1). We used a 576 × 188 model grid and a grid spacing of 16 m. We calculated synthetic data for 115 shots recorded by 576 receivers at the surface. The initial velocity model for the full waveform inversion linearly increases from 1.5 km/s to 5.5 km/s. To test our algorithm, we compared the inversion results of the proposed Gauss-Newton method with those of the gradient method. Figs. 2(a) and 2(b) show the inverted velocity models obtained with the gradient method at the 500-th iteration and with the Gauss-Newton method at the 300-th iteration, respectively. From Fig. 2(a), we confirm that the gradient method generates a distorted image in the lower left corner of the model and below 2 km depth. However, the Gauss-Newton method produced well-defined velocity structures [Fig. 2(b)]. Fig. 3 shows the history of the phase RMS error throughout the Marmousi model inversion. The Gauss-Newton method shows faster convergence than the gradient method and the two methods give similar results.

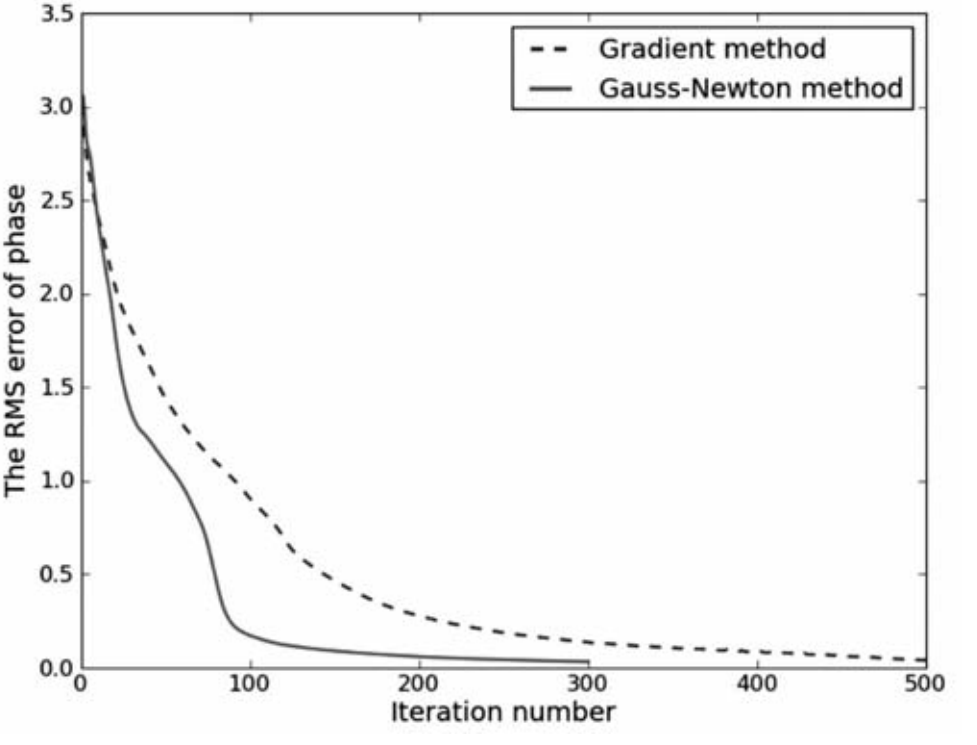


Fig. 3. The RMS error of phase for the Marmousi model inversion.

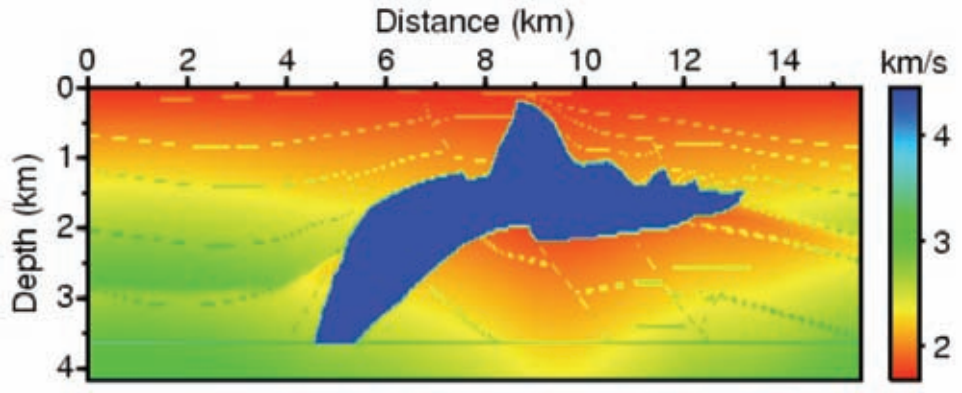


Fig. 4. A-A′ cross section of the SEG/EAGE salt model.

The second test was carried out on using the A-A′ cross section of the SEG/EAGE salt model (Fig. 4). We selected 60 frequencies from 0.2 to 12 Hz to generate a synthetic data set. We used a 779 x 209 model mesh with a grid spacing of 20 m and calculated data for 155 shot gathers and 779 receivers at the surface. The initial velocity model linearly increases from 1.679 km/s to 4.45 km/s. Figs. 5(a) and 5(b) show the inverted velocity models obtained by the gradient method at the 2000-th iteration and by the Gauss-Newton method at the 300-th iteration, respectively. From Figs. 5(a) and 5(b), we can observe that the gradient method fails to recover the sub-salt velocity, while the Gauss-Newton method more accurately recovers the velocity structures below the salt layer. Fig. 6 shows the RMS error of phase for the SEG/EAGE salt model inversion. Unlike the first example, the Gauss-Newton method shows significant improvement in convergence speed when compared with the gradient method. This result implies that the proposed Gauss-Newton method is efficient and robust when inverted for a high-contrast velocity model.

CONCLUSIONS

In this paper, we test the approximated Gauss-Newton method using the linear CG method for the logarithmic objective function. While the gradient method needs a reasonable weighting scheme and a line search algorithm for determining step length, the Gauss-Newton method gives realistic velocity perturbations without additional manipulation. Although numerical instability must be treated properly when using the logarithmic objective function, instability can be easily manipulated with a heuristic filtering method.

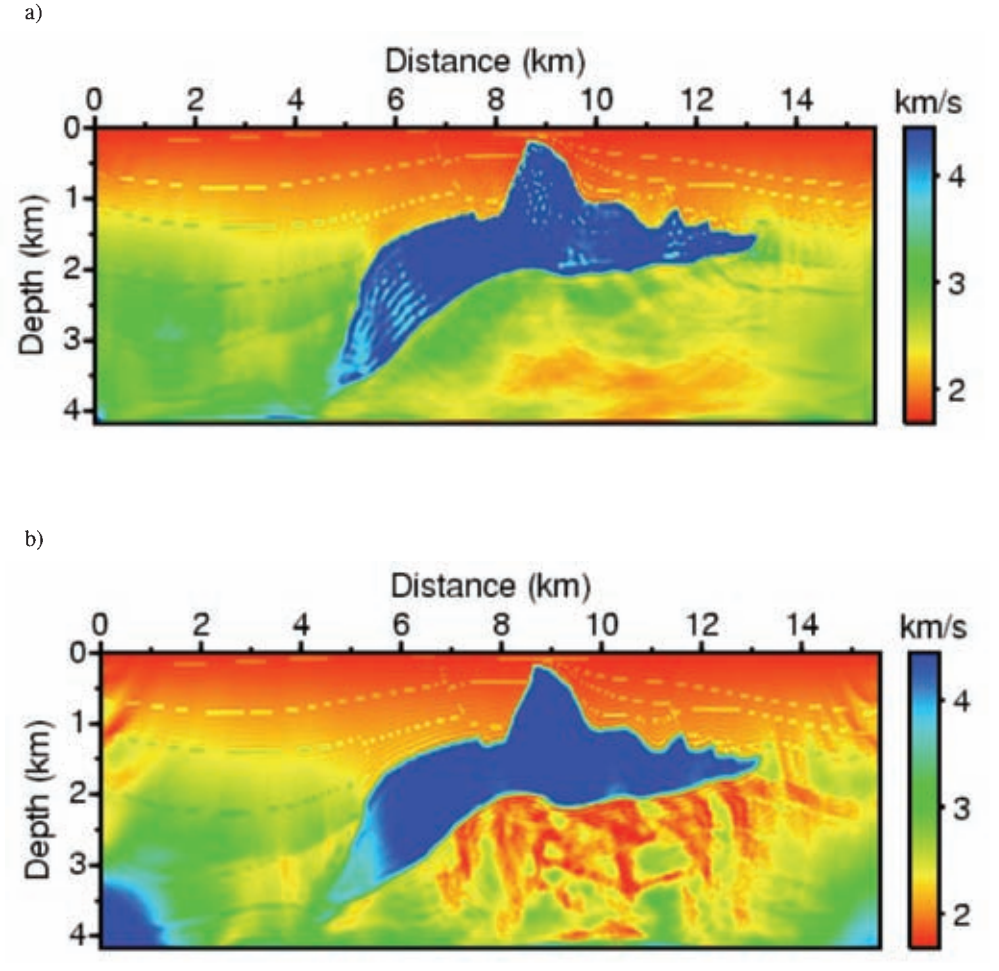


Fig. 5. Inverted velocity model for data based on the SEG/EAGE model obtained at (a) the 2000-th iteration of the gradient method and (b) the 300-th iteration of the Gauss-Newton method.

In order to stabilize the inverse problem, we introduce an additional regularization term into the objective function. A simple damping method was implemented at the cost of half of a CG iteration. Numerical examples show that the proposed Gauss-Newton method successfully deals with logarithmic wavefields and accelerates the convergence for a high-contrast salt structure velocity model. The Gauss-Newton method using logarithmic wavefields is able to recover a low velocity structure beneath the salt dome, while the gradient method could not. This implies that the proposed method can illuminate regions below salt layers and properly deal with amplitude recovery.

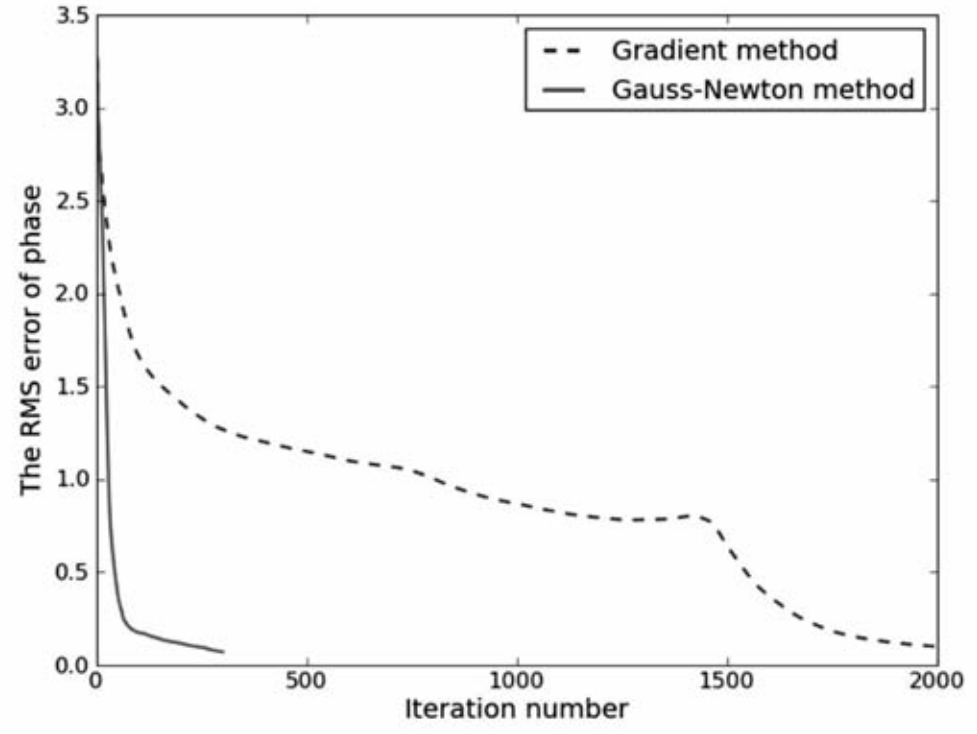


Fig. 6. The RMS error of phase for the SEG/EAGE salt model inversion.

ACKNOWLEDGEMENTS

This work was supported by the Energy Efficiency and Resources of the Korea Institute of Energy Technology Evaluation and Planning (KETEP) grant funded by the Korea government Ministry of Knowledge Economy (No. 2010T100200396).

REFERENCES

Brossier, R., Operto, S. and Virieux, J., 2009. Seismic imaging of complex onshore structures by 2D elastic frequency-domain full-waveform inversion. Geophysics, 74: WCC105-WCC118.

Chen, P., Jordan, T.H. and Zhao, L., 2007. Full three-dimensional tomography: a comparison between the scattering-integral and adjoint-wavefield methods. Geophys. J. Internat., 170: 175-181.

Epanomeritakis, I., Akcelik V., Ghattas, O. and Bielak, J., 2008. A Newton-CG method for large-scale three-dimensional elastic full-waveform seismic inversion. Inverse Problems, 24: 1-26.

Golub, G.H. and Van Loan, C.F., 1996. Matrix Computations. 3rd ed. Johns Hopkins University Press, Baltimore and London.

Hu, W., Abubakar A. and Habashy T.M., 2009. Simultaneous multifrequency inversion of full-waveform seismic data. Geophysics, 74: R1-R14.

Mackie, R.L. and Madden, T.R., 1993. Three-dimensional magnetotelluric inversion using conjugate gradients. Geophys. J. Int., 115: 215-229.

Marfurt, K.J., 1984. Accuracy of finite-difference and finite-element modeling of the scalar and elastic wave equations. Geophysics, 49: 533-549.

Nocedal, J., 1980. Updating quasi-Newton matrices with limited storage. Math. Comput., 35: 773-782.

Pratt, G.R., Shin, C. and Hicks, G.J., 1998. Gauss-Newton and full Newton methods in frequency-space seismic waveform inversion. Geophys. J. Int., 133: 341-362.

Sambridge, M., Tarantola, A. and Kennett, B.L.N., 1991. An alternative strategy for non-linear inversion of seismic waveforms. Geophys. Prosp., 39: 457-491.

Shin, C. and Cha, Y.H., 2008. Waveform inversion in the Laplace domain. Geophys. J. Internat., 173: 922-931.

Shin, C. and Cha, Y.H., 2009. Waveform inversion in the Laplace-Fourier domain. Geophys. J. Internat., 177: 1067-1079.

Shin, C., Koo, N.-H., Cha, Y.H. and Park, K.-P., 2010. Sequentially ordered single-frequency 2-D acoustic waveform inversion in the Laplace-Fourier domain. Geophys. J. Internat., 181: 935-950.

Shin, C. and Min, D.-J., 2006. Waveform inversion using a logarithmic wavefield. Geophysics, 71: R31-R42.

Tarantola, A., 1984. Inversion of seismic reflection data in the acoustic approximation. Geophysics, 49: 1259-1266.

Virieux, J. and Operto, S., 2009. An overview of full-waveform inversion in exploration geophysics. Geophysics, 74: WCC1-WCC26.