Implicit and explicit preconditioning for least squares nonstationary phase shift

Marcus R. Wilson and Robert J. Ferguson

ABSTRACT

An implicit preconditioned conjugate gradient scheme is derived to implement nonstationary phase shift for irregularly sampled seismic data using least squares. This implicit scheme gives fast convergence at all frequencies at the cost of an approximation to the evanescent filter. This results in some error in the resulting phase shifted and regularized data. Our implicit scheme suggests an explicit scheme which unfortunately does not perform any better than the standard unconditioned scheme. The fast implicit scheme suggests that an appropriate preconditioner can be found that will reduce the runtime of the algorithm without sacrificing accuracy, and this will result in a robust trace regularization and statics algorithm for use in heterogeneous media.

INTRODUCTION

To correct for surface statics and irregular trace spacing before migration, Ferguson (2006) presents an inversion algorithm based on the phase shift method of Gazdag (1978). Acquired seismic data is extrapolated recursively through the near surface by weighted-damped least squares. The result is a regularly sampled wavefield at a flat datum, which can then be imaged using migration techniques that use the fast Fourier transform.

Implementation of this operator as a matrix is extremely costly to compute. The associated Hessian is constructed at the cost of matrix-matrix multiplication, with complexity $O(n^3)$, where n is the number of traces. Inversion of the Hessian by Gaussian elimination also has complexity $O(n^3)$. These computations have to be repeated for every depth step and every frequency.

We can reduce these costs by recasting the problem in a conjugate gradient framework, replacing matrices with function calls, where the Hessian is applied as a forward operator, and the extrapolated wavefield can be computed by an iterative search. The cost of the resulting inversion scheme is the cost of applying the forward operator times the number of iterations required for an acceptable approximation. Wilson and Ferguson (2010) presents an application of this inversion scheme. The cost of applying the forward operator can be reduced to $O(vnlogn)$, where v is the number of reference velocities in the velocity model. The algorithm converges in under $\sqrt{n}$ iterations for large frequencies, but fails to converge quickly for lower frequencies.

Wilson and Ferguson (2010) postulate that the poor convergence in the lower frequencies is caused by the evanescent filter embedded in the phase shift extrapolator. Here we will derive two preconditioning schemes by which the effects of this filter can be mitigated, and we observe the effects of this change on the convergence rate of the conjugate gradient method.

THEORY

A wave equation inversion for seismic data given by Ferguson (2006) simultaneously corrects for velocity variation in the near surface and irregular trace spacing using non-stationary phase shift operators. Here we discuss the development of these operators, and their applications to statics and trace regularization. We then build a framework by which to design preconditioning operators to improve the inversion of this operator by conjugate gradients.

Non-stationary Phase Shift Operators

In a layered medium, the phase shift operator acts within a layer on a monochromatic wavefield φ_z at depth z by way of a spatial fast Fourier transform, followed by multiplication by an extrapolation function, then an inverse fast Fourier transform. Written as matrices, we have

$$P_{\Delta z}(\varphi_z) = [IFT]\,[\alpha_{\Delta z}]\,[FT]\,\varphi_z. \tag{1}$$

Here $[\alpha_{\Delta z}]$ is a diagonal matrix that applies the phase shift operator in the wavelike region, where $|\frac{\omega}{v_z}| \leq |k_x|$, and attenuates energy in the evanescent region, where $|\frac{\omega}{v_z}| > |k_x|$. The diagonal elements of $[\alpha_{\Delta z}]$ are computed from the layer velocity v_z and the input wavenumber k_x using the formula,

$$\alpha_{\Delta z}(k_x, v_z) = e^{i\Delta z k_z}, \tag{2}$$

where the wavenumber k_z must satisfy the dispersion relation,

$$k_x^2 + k_z^2 = \left(\frac{\omega}{v_z}\right)^2, \tag{3}$$

where v_z is the layer velocity. We can choose the sign of k_z so that the operator propagates the wavefield in the direction of Δz in the wavelike region, where $|\frac{\omega}{v}| \leq |k_x|$, and attenuates energy in the evanescent region, where $|\frac{\omega}{v}| > |k_x|$. These conditions are satisfied in Ferguson (2010), where k_z is given by,

$$k_z = \mathrm{Re}\left\{\sqrt{\left(\frac{\omega}{v}\right)^2 - k_x^2}\right\} + i\,\mathrm{sgn}(\Delta z)\,\mathrm{Im}\left\{\sqrt{\left(\frac{\omega}{v}\right)^2 - k_x^2}\right\}. \tag{4}$$

To accommodate lateral velocity variation, we use a set of constant velocity windows, defined for a given reference velocity v_j by

$$\Omega_j(x) = \begin{cases} 1 & \text{if } v(x) = v_j \\ 0 & \text{if } v(x) \neq v_j \end{cases}, \tag{5}$$

and the phase shift operator becomes

$$P_{\Delta z}(\varphi_z) = \sum_j [\Omega]_j [IFT][\alpha_{\Delta z}]_j [FT]\varphi_z. \tag{6}$$

Evanescent Filter

The wavefield extrapolator $\alpha_{\Delta z}$ can be factored into two parts: a complex exponential that performs the phase shift, and a negative real exponential, which acts as an evanescent filter. Wilson and Ferguson (2010) note that convergence of the least squares inversion of $P_{\Delta z}$ was fast for high frequencies, and slow for low frequencies, where the data vector crosses into the evanescent region. It was postulated that this slow convergence was the result of the evanescent filter. To overcome this difficulty, we can factor the filter out of the least-squares Hessian. To that end, express $\alpha_{\Delta z}$ as,

$$\begin{aligned} \alpha_{\Delta z} &= \exp\left(2\pi i \Delta z k_z\right) \\ &= \exp\left(2\pi i \Delta z \operatorname{Re}\{k_z\} - |\Delta z| \operatorname{Im}\{k_z\}\right) \\ &= \exp\left(2\pi i \Delta z \operatorname{Re}\{k_z\}\right) \exp\left(-|\Delta z| \operatorname{Im}\{k_z\}\right). \\ &= \alpha^P_{\Delta z} \alpha^F_{\Delta z}. \end{aligned} \tag{7}$$

So we can factor $[\alpha_{\Delta z}]$ into two diagonal matrices, one that applies the phase shift, and one that applies the filter.

$$[\alpha_{\Delta z}] = [\alpha^P_{\Delta z}][\alpha^F_{\Delta z}] \tag{8}$$

Now if we alter each $[\alpha^F_{\Delta z}]$ to filter with respect to the highest reference velocity, we can factor the matrix $P_{\Delta z}$.

$$\begin{aligned} P_{\Delta z}(\varphi_z) &= \sum_j [\Omega]_j [IFT][\alpha^P_{\Delta z}][\alpha^F_{\Delta z}][FT]\varphi_z \\ &= \{\sum_j [\Omega]_j [IFT][\alpha^P_{\Delta z}]\}\{[\alpha^F_{\Delta z}][FT]\}\varphi_z \\ &= \bar{P}_{\Delta z} F \varphi_z \end{aligned} \tag{9}$$

We can then factor the operator F out of the Hessian matrix, which results in a better conditioned system for our conjugate gradient framework.

Least Squares Minimization

To correct for surface statics and irregular trace spacing, Ferguson (2006) models seismic data as follows: given a recorded wavefield φ_z at depth z, assume that $\varphi_z = W_e P_{-\Delta z} \varphi_{z+\Delta z} + \eta$, where $P_{-\Delta z}$ is an upward phase shift, as in Equation 6, W_e is a weighting operator that models irregular trace spacing and topography, as in Reshef (1991), and η is an additive noise term. This is a mixed-determined linear system (Menke, 1989), so the least-squares approximation of $\varphi_{z+\Delta z}$ can be recovered by solving the normal equations,

$$P^*_{-\Delta z} W_e \varphi_z = \left[P^*_{-\Delta z} W_e P_{-\Delta z} + \varepsilon W_m\right] \varphi_{z+\Delta z}. \tag{10}$$

Here $P^*_{-\Delta z}$ is the adjoint of $P_{-\Delta z}$, W_m is a smoothing operator, and ϵ is a user parameter that controls the amount of smoothing. However, since we have $P_{-\Delta z} = \bar{P}_{-\Delta z} F$, we can write $P^*_{-\Delta z} = F^* \bar{P}^*_{-\Delta z}$, and our normal equations become,

$$F^* \bar{P}^*_{-\Delta z} W_e \varphi_z = \left[F^* \bar{P}^*_{-\Delta z} W_e \bar{P}_{-\Delta z} F + \varepsilon W_m\right] \varphi_{z+\Delta z}. \tag{11}$$

Factoring out the Fs and cancelling F^* on the right gives us,

$$\bar{P}^*_{-\Delta z} W_e \varphi_z = \left[\bar{P}^*_{-\Delta z} W_e \bar{P}_{-\Delta z} + \varepsilon (F^{-1})^* W_m F^{-1}\right] F \varphi_{z+\Delta z} \tag{12}$$

Equation 12 can be solved for $F\varphi_{z+\Delta z}$ by conjugate gradients, and the extrapolated wavefield can then be computed by inverting F, which is fast as F is just a diagonal operator followed by a fast Fourier transform.

Complexity

Many techniques exist to solve linear systems, and the cost of solving the normal equations (Equation 10) will vary depending on the inversion method used. A summary of the complexities of various options can be found in Table 1 on page 5. Direct matrix methods such as Gaussian Elimination and LU Factorization are widely used due to their versatility and ease of use, but require explicit computation of the Hessian matrix, and generally carry the largest cost (Burden and Faires, 2001). When speed is desired, iterative methods can be used to seek an approximate solution with fewer computations.

For ease of notation, denote by H the Hessian operator on the right-hand side of Equation 10, and b the transformed wavefield vector on the left-hand side. The problem can then be written as a linear system given by,

$$Hx = b, \tag{13}$$

where we wish to compute the unknown vector x. To evaluate x by Gaussian Elimination, we would first need to compute the matrix form of H. The simplest way to do this is to first compute the matrix form of $P_{-\Delta z}$. For a survey with n trace locations, we compute $P_{-\Delta z}$ by applying the phase shift operator to the columns of the $n \times n$ identity matrix. The outputs would become the respective columns of $P_{-\Delta z}$. The cost of applying $P_{-\Delta z}$ to a single column is $\mathcal{O}(vn \log n)$, where v is the number of different velocities found in our model at the current depth step (Ferguson and Margrave, 2002). Therefore, full evaluation of $P_{-\Delta z}$ is $\mathcal{O}(vn^2 \log n)$. We can then take the adjoint of this matrix, and compute $P^*_{-\Delta z} W_e P_{-\Delta z}$ by matrix-matrix multiplication, which is $\mathcal{O}(n^3)$.

To reduce this cost, Ferguson (2006) computes only a limited number of diagonals of the matrices for $P_{-\Delta z}$ and $P^*_{-\Delta z}$, and sets the remaining entries to zero. This can be done in $\mathcal{O}(dn^2)$, where d is the number of diagonals computed. Furthermore, multiplying two d-diagonal matrices together can be accomplished in $\mathcal{O}(d^2 n)$. This constraint forces a dip limitation on the data can be handled, as we are asserting that the behaviour of a given point in space at one depth level cannot be affected by points in space at adjacent depth levels that are more than d spaces away.

We could eliminate the need for matrix-matrix multiplication here by applying the full Hessian operator to the columns of the identity matrix, which would reduce the total cost to n applications each of $P_{-\Delta z}$ and its adjoint, plus n applications of the weight matrix, for a total complexity of $\mathcal{O}(2vn^2 \log n + n^2)$, or simply $\mathcal{O}(vn^2 \log n)$. Proceeding in this manner takes advantage of the fact that $P_{-\Delta z}$ uses the fast Fourier transform, so applying $P_{-\Delta z}$ is faster than matrix-vector multiplications for large enough values of n.

Operation	**Type**	**Algorithm**	**Complexity**
Hessian Construction	Matrix	Phase Shift function plus Matrix Multiplication	$\mathcal{O}(n^3)$
		d-Diagonal Matrix Multiplication	$\mathcal{O}(dn^2)$
		Full Hessian Function	$\mathcal{O}(vn^2 \log n)$
		Series Approximation	$\mathcal{O}(n^2 \log n)$
	Function Call	Prewritten Function	$\mathcal{O}(0)$
Hessian Application	Matrix	Matrix-Vector Multiplication	$\mathcal{O}(n^2)$
	Function Call	Full Hessian Function	$\mathcal{O}(vn \log n)$
Inversion	Matrix	Gaussian Elimination	$\mathcal{O}(n^3)$
		LU Decomposition	$\mathcal{O}(n^3)$
		Conjugate Gradients	$\mathcal{O}(C(n)n^2)$
	Function Call	Conjugate Gradients	$\mathcal{O}(C(n)H(n))$

Table 1. Complexities of computational options for algorithm construction. Due to the variety of Hessian application types, $H(n)$ is used to denote the complexity of the Hessian construction and application used. v is the number of distinct velocities in the model, and $C(n)$ is the number of conjugate gradient iterations required for an acceptable approximation.

The complexity of computing and adding εW_m varies with the choice of smoother, but can be chosen, as in Ferguson (2006) and Ferguson (2010), so that the cost is low compared to the previous steps, so we will assume the cost is negligible here. Finally, Gaussian Elimination itself is $\mathcal{O}(n^3)$ (Cohn et al., 2005). Each of these steps occurs in sequence, so the total complexity of the method is equivalent to the most expensive step, which is $\mathcal{O}(n^3)$.

To further reduce the cost, Ferguson (2010) expresses the Hessian as the composition of the forward and adjoint operators, and derives a truncated Taylor series expansion to reduce the cost of computing the matrix form. Computation of the resulting approximate Hessian is $\mathcal{O}(n^2 \log n)$, which is independent of the number of different velocities present in the layer. Smith et al. (2009) replaces Gaussian Elimination with conjugate gradients to reduce the runtime of the inversion by a factor of 10, but the number of conjugate gradient iterations required is not specified, so an asymptotic runtime can not be derived.

Conjugate Gradients

The conjugate gradient method is an iterative algorithm used to approximate a solution to a positive definite linear system of equations (Hestenes and Stiefel, 1952). In our case it can be used to recover the source wavefield $\varphi_{z+\Delta z}$ from Equation 10. In contrast to direct matrix methods, the conjugate gradient algorithm uses an iterative search technique that can obviate the need to compute the Hessian matrix explicitly. This is desirable when an application of the operator is much faster than standard matrix-vector multiplication, as is the case in our algorithm.

To solve a linear system by conjugate gradients involves choosing an initial value x_0, and computing a residual vector $r_0 = b - Hx_0$. The cost of this step is dominated by the cost of applying the operator H to x_0. This residual vector defines a search direction that we use to refine our guess. Subsequent iterations are similar, except the search directions are adjusted to take advantage of the positive definite structure of the operator H. For an $n \times n$ matrix H, assuming perfect arithmetic, this method is guaranteed to produce an exact solution to the system after n iterations (Hestenes and Stiefel, 1952), and an acceptable approximation can be attained using machine arithmetic in fewer iterations if the matrix is well conditioned. In this case, we should be able to solve the matrix form of the system in $\mathcal{O}(C(n)n^2)$, and the functional form in $\mathcal{O}(C(n)vn\log(n))$, where $C(n)$ is the number of conjugate gradient iterations required. This is a significant cost decrease when n is very large, and the number of reference velocities in the model is small.

If the system is particularly sensitive to rounding errors, this method might not find a solution to the system quickly, and may fail to find an acceptable approximation at all. We call such a system "ill conditioned." In fact, Wilson and Ferguson (2009) implements this algorithm with no preconditioning, and notes that the algorithm tends to converge quickly in the high frequencies, where no evanescent filter is applied, and very slowly or not at all in the lower frequencies. Such a system must first be preconditioned by selecting an invertible conditioning operator C so that the operator H_C given by

$$H = CH_CC^* \tag{14}$$

is better conditioned (Burden and Faires, 2001). Then we can apply conjugate gradients to solve an alternate system given by,

$$H_C\varphi_C = \tilde{\varphi}_C, \tag{15}$$

where $\varphi_C = C^*\varphi$ and $\tilde{\varphi}_C = C^{-1}\tilde{\varphi}$, and compute $(C^*)^{-1}\varphi_C$ to obtain the desired result.

Implicit Preconditioner

Noting the similarities between Equation 14 and Equation 11 suggests a first choice of preconditioner. If we think of C has being factored out of the original operator on the left, and C^* as being factored out on the right, the factorization in Equation 11 immediately gives us

$$\begin{aligned} H &= \left[P^*_{-\Delta z}W_eP_{-\Delta z} + \varepsilon W_m\right] && (16)\\ H_C &= \left[\bar{P}^*_{-\Delta z}W_e\bar{P}_{-\Delta z} + \varepsilon(F^{-1})^*W_mF^{-1}\right] && (17)\\ C &= F^*. && (18) \end{aligned}$$

Since H is our poorly conditioned system, the implication that H_C is better conditioned would lead us to try F^* as a preconditioner, where F^* is given by

$$\begin{aligned} F^* &= \left([\alpha^F_{\Delta z}][FT]\right)^* \\ &= [FT]^*[\alpha^F_{\Delta z}]^* \\ &= [IFT][\alpha^F_{\Delta z}]. \end{aligned} \tag{19}$$

Noting that $[FT]$ is unitary, and $[\alpha^F_{\Delta z}]$ is diagonal and real valued, the inverse and inverse adjoints of F are simple to derive.

As a result, we have an implicitly preconditioned inversion scheme that we can use to solve our problem. We simply construct the normal equations as usual, but using the transformed phase shift operator with no evanescent attenuation. Then we feed the resulting operator H_C and input data b into the standard conjugate gradient algorithm, which will return an approximation of Fx. Then we can derive the result of the original problem by applying F^{-1} to this approximation.

Explicit Preconditioner

In order to derive the implicit scheme we were forced to make an approximation; we had to assume that all the phase shift symbols were acting with respect to the same velocity, so that each evanescent filter was the same. This introduces an error into the calculation that may cause undesirable artifacts in the result. However, the conjugate gradient algorithm can be adapted to handle preconditioning explicitly (Burden and Faires, 2001). To apply explicit preconditioning, we would pass the original operator H, along with the data vector b, and the preconditioning operator $C = F^*$ into the algorithm. We expect to see a similar speedup to that of the implicit scheme here, without the artifacts created by the approximation.

EXAMPLES

To test these preconditioners, we forward propagate a trace gather of 256 synthetic traces (Figure 1(b)) using the operator $P_{\Delta z}$ defined in Equation 6 using the reference velocity model shown in Figure 1(a), and with $\Delta z = -100$m. A random noise term is added to the result, set to 40db below the signal level, and a random sample of 30% of the traces are set to zero. The transformed wavefield is shown in Figure 1(c). This data is then run through the inversion, first with no preconditioning, then using the implicit scheme, followed by the explicit scheme.

With no preconditioning, we can see in Figure 2(a) that the wavefield is effectively recovered, with some artifacts arising where there were significant gaps in trace coverage, although missing traces were effectively interpolated. However we can see in Figure 2(c) that the inversion was slow, and failed to converge to an acceptable solution in the low frequencies. The residual error (Figure 2(d)) correlates with the convergence rates.

Using the implicit preconditioning scheme, we can see in Figure 3(a) that the wavefield is likewise effectively recovered, but some additional numerical error is introduced because of our approximation of the evanescent filter. However the convergence rate is dramatically improved, as the algorithm converged everywhere in under 30 iterations (Figure 3(c)). This is not as good as the 16 iterations implied by Burden and Faires (2001), but it remains to see how this number scales as n increases. The residual error is below the tolerance of 10^{-6} that was allowed during the inversion.

We would expect that some of this improvement would carry over to the explicitly preconditioned scheme, but this was not the case. Convergence in the low frequencies was worse than that of the standard scheme (Figure 4(c)), and the residual error is much higher in the 20-50Hz range than either of the previous two schemes (Figure 4(d)). The image quality of the recovered data is acceptable, however, and comparable to the standard and implicit schemes.

The fast convergence of the implicit scheme suggests that there is a choice of preconditioning operator that would improve convergence in the explicit scheme without damaging the accuracy of the solution, although the clear choice for this operator does not give us the results we want, and a better operator has yet to be determined. Such an operator would give us a robust method for performing trace regularization on seismic data that may be feasible for very large trace gathers.

CONCLUSION

We have demonstrated the possibility of a preconditioning operator that will make this algorithm run quickly and accurately on large trace gathers, although the exact nature of this operator has yet to be determined. The implicit scheme is much faster than the standard scheme, but requires us to make an approximation that causes artifacts in the resulting output wavefield. The explicit scheme failed to live up to the expectations generated by the implicit scheme, but other choices of preconditioner exist that have yet to be explored. The implicit scheme gives us a fast, although not very accurate, implementation of the statics and trace regularization method, that we could use to perform a large scale test of the algorithm, to see how the runtime scales with the size of the input.

ACKNOWLEDGEMENTS

The authors wish to thank the sponsors, faculty, staff and students of the Consortium for Research in Elastic Wave Exploration Seismology (CREWES), and the Natural Sciences and Engineering Research Council of Canada (NSERC, CRDPJ 379744-08) for their support of this work.

REFERENCES

Burden, R., and Faires, J. D., 2001, Numerical Analysis: Brooks/Cole, 511 Forest Lodge Road, Pacific Grove, CA 93950 USA.

Cohn, H., Kleinberg, R., Szegedy, B., and Umans, C., 2005, Group-theoretic algorithms for matrix multiplication, *in* FOCS '05: Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society, Washington, DC, USA, 379–388.

Ferguson, R. J., 2006, Regularization and datuming of seismic data by weighted, damped least squares: Geophysics, **71**, No. 5, U67–U76.

Ferguson, R. J., 2010, Efficient simultaneous wave-equation statics and trace regularization by series approximation: Geophysics, **75**, No. 3, U19–U28.

Ferguson, R. J., and Margrave, G. F., 2002, Prestack depth migration by symmetric nonstationary phase shift: Geophysics, **67**, No. 2, 594–603.

Gazdag, J., 1978, Wave equation migration with the phase-shift method: Geophysics, **43**, No. 7, 1342–1351.

Hestenes, M. R., and Stiefel, E., 1952, Methods of conjugate gradients for solving linear systems: Journal of Research of the National Bureau of Standards, **49**, No. 6.

Menke, W., 1989, Geophysical Data Analysis: Academic Press, 525B Street, Suite 1900, San Diego, California 92101-4495.

Reshef, M., 1991, Depth migration from irregular surfaces with depth extrapolation methods: Geophysics, **56**, No. 1, 119–122.

Smith, D. R., Sen, M. K., and Ferguson, R. J., 2009, Optimized regularization and redatum by conjugate gradients: 71st EAGE Conference & Exhibition, **71**.

Wilson, M. R., and Ferguson, R. J., 2009, Trace interpolation and elevation statics by conjugate gradients: CREWES Research Report, **21**.

Wilson, M. R., and Ferguson, R. J., 2010, Trace interpolation and elevation statics by conjugate gradients: GeoCanada 2010 Conference Abstracts.

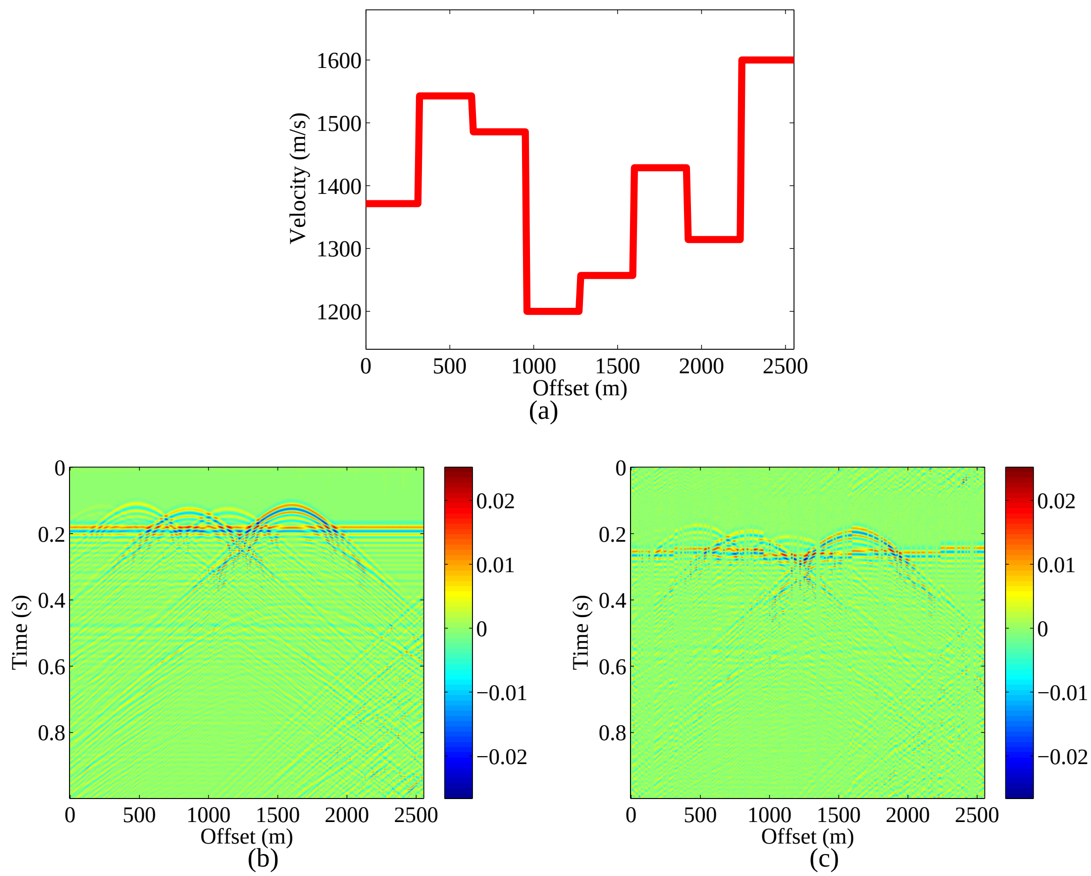

FIG. 1. (a) Laterally varying velocity model. (b) The unknown source wavefield. (c) Forward modelled synthetic data.

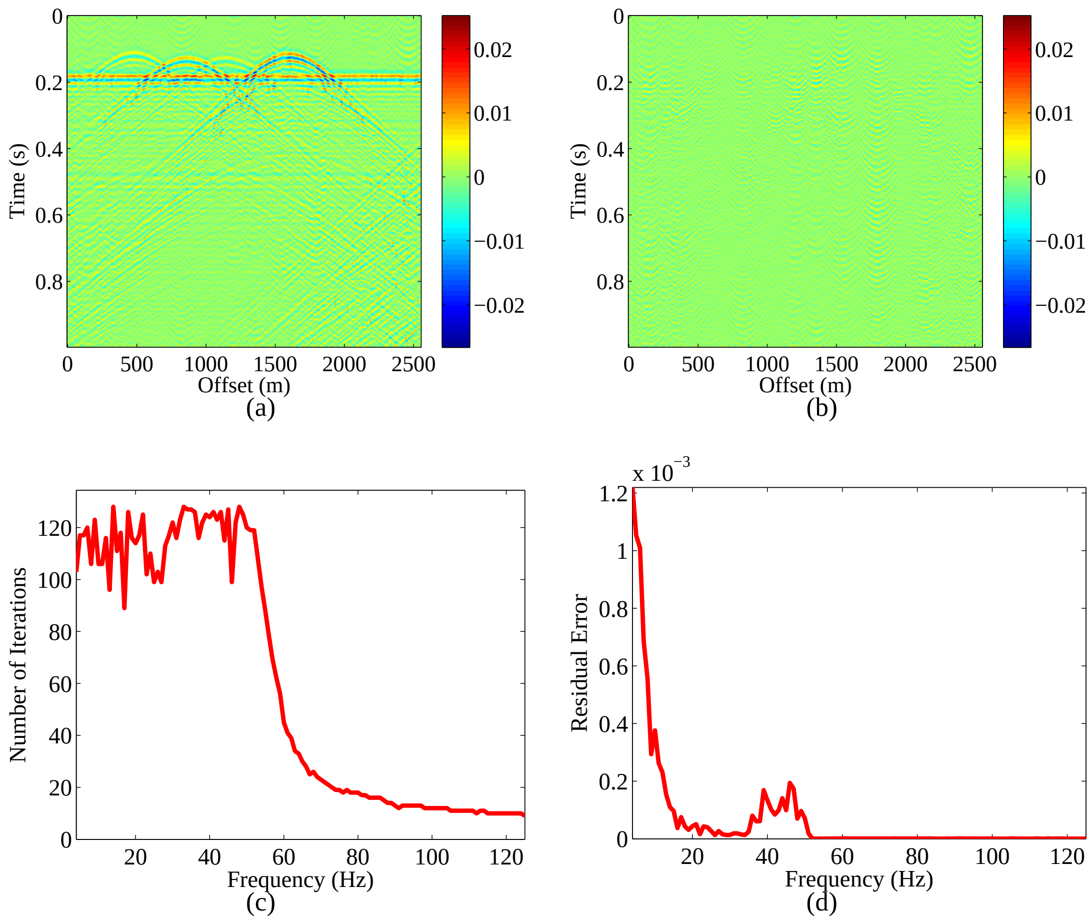

FIG. 2. Output for the nonpreconditioned scheme. (a) The recovered wavefield. (b) The difference between the recovered wavefield and the source. (c) The number of CG iterations required at each frequency. (d) The residual error of the output at each frequency.

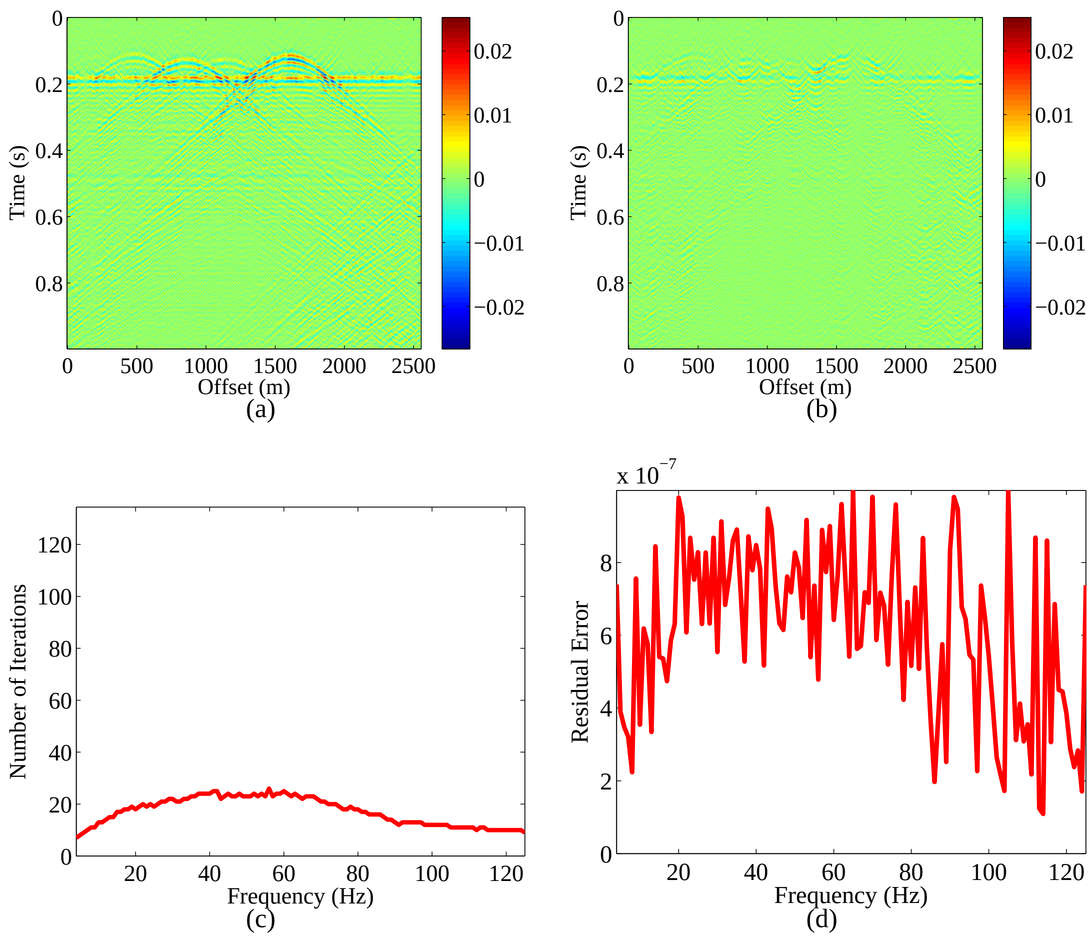

FIG. 3. Output for the implicit preconditioning scheme. (a) The recovered wavefield. (b) The difference between the recovered wavefield and the source. (c) The number of CG iterations required at each frequency. (d) The residual error of the output at each frequency.

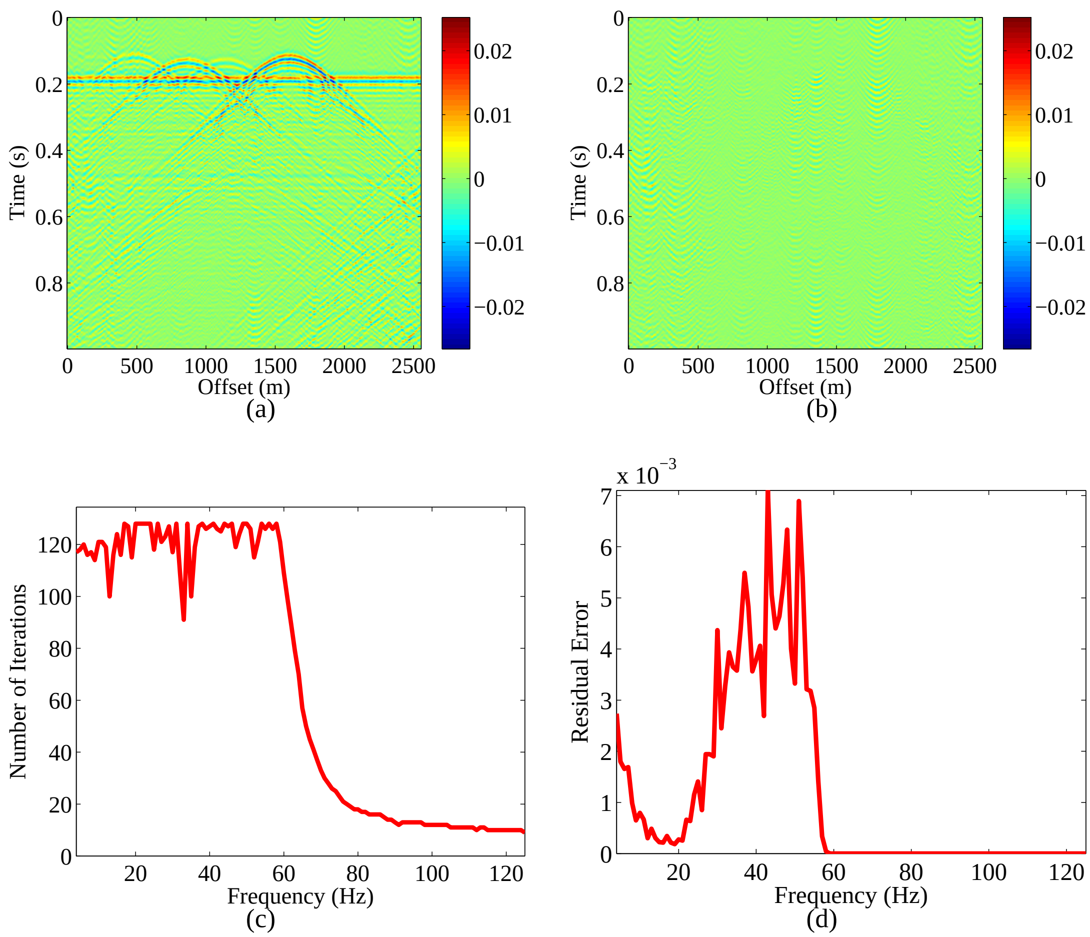

FIG. 4. Output for the explicit preconditioning scheme. (a) The recovered wavefield. (b) The difference between the recovered wavefield and the source. (c) The number of CG iterations required at each frequency. (d) The residual error of the output at each frequency.