

A new iterative solver for the time-harmonic wave equation

C. D. Riyanti¹, Y. A. Erlangga¹, R.-E. Plessix², W. A. Mulder², C. Vuik¹, and C. Oosterlee¹

ABSTRACT

The time-harmonic wave equation, also known as the Helmholtz equation, is obtained if the constant-density acoustic wave equation is transformed from the time domain to the frequency domain. Its discretization results in a large, sparse, linear system of equations. In two dimensions, this system can be solved efficiently by a direct method. In three dimensions, direct methods cannot be used for problems of practical sizes because the computational time and the amount of memory required become too large. Iterative methods are an alternative. These methods are often based on a conjugate gradient iterative scheme with a preconditioner that accelerates its convergence. The iterative solution of the time-harmonic wave equation has long been a notoriously difficult problem in numerical analysis. Recently, a new preconditioner based on a strongly damped wave equation has herald-

ed a breakthrough. The solution of the linear system associated with the preconditioner is approximated by another iterative method, the multigrid method. The multigrid method fails for the original wave equation but performs well on the damped version. The performance of the new iterative solver is investigated on a number of 2D test problems. The results suggest that the number of required iterations increases linearly with frequency, even for a strongly heterogeneous model where earlier iterative schemes fail to converge. Complexity analysis shows that the new iterative solver is still slower than a time-domain solver to generate a full time series. We compare the time-domain numeric results obtained using the new iterative solver with those using the direct solver and conclude that they agree very well quantitatively. The new iterative solver can be applied straightforwardly to 3D problems.

INTRODUCTION

For imaging seismic data, the oil and gas industry is gradually moving from ray-based techniques to finite-difference wave-equation migration. Ray-based methods are difficult to use or may even fail in complex earth models. Wave-equation migration can handle these situations better. To control computing costs, the wave equation is usually replaced by a one-way or paraxial approximation (Claerbout, 1985; Bamberger et al., 1988; Collino and Joly, 1995; Biondi and Palacharla, 1996; Jin et al., 1998). This approximation is in most cases valid for not-too-large velocity contrasts and not-too-wide angles of incidence. With increased computer power, it may become worthwhile to develop finite-difference two-way or full wave-equation migration techniques without making the approximations

needed for ray-based or one-way migration methods (Yoon et al., 2004).

In 2D space, two-way wave-equation migration can be carried out efficiently by working in the frequency domain. In that case, the LU decomposition of the matrix arising from the discretization of the two-way wave equation is computed once with a direct method for each frequency. The result can be used to compute all wavefields for all shots and also for back-propagated receiver wavefields. The latter correspond to the reverse-time wavefields in the time domain (Marfurt, 1984; Mulder and Plessix, 2004b; Plessix and Mulder, 2004). This makes the method an order of magnitude faster than its time-domain counterpart when many shots must be processed. In 3D space, this is impossible because the problem's size soon exceeds the capabilities of direct solvers. Obviously, this situation can be avoided by reverting to the time domain. Present-day hardware allows this only

Manuscript received by the Editor November 10, 2004; revised manuscript received January 14, 2006; published online August 30, 2006.

¹Delft University of Technology, Delft Institute of Applied Mathematics, Mekelweg 4, 2628 CD Delft, The Netherlands. E-mail: c.d.riyanti@ewi.tudelft.nl; yogiae@gmail.com; c.vuik@tudelft.nl.

²Shell International Exploration and Production, P.O. Box 60, 2280 AB Rijswijk, The Netherlands. E-mail: reneedouard.plessix@shell.com; wim.mulder@shell.com.

© 2006 Society of Exploration Geophysicists. All rights reserved.

on very coarse grids, meaning that only the low frequencies in the data can be used. Nevertheless, 3D two-way wave-equation migration is achievable in the time domain on large computer clusters (Yoon et al., 2003; Yoon et al., 2004).

Time-domain reverse-time migration requires storing the forward and time-reversed wavefields at time intervals that avoid aliasing. These wavefields are correlated to obtain a partially migrated image for each shot. Stacking over shots provides the desired result. In the frequency domain, only one forward- and one back-propagated wavefield need to be stored. These are multiplied to obtain a partial migration image. The summation over shots and frequencies produces the final migration image. In this way, storage requirements are reduced considerably.

Because direct solvers are computationally prohibitive, a suitable iterative method for the two-way wave equation is needed. A clear disadvantage of an iterative method is that it must be applied for each shot and each back-propagated wavefield computation. A direct method allows an LU decomposition of the linear system. Once this costly step has been carried out, all shot and receiver wavefields can be computed at a small cost (Marfurt, 1984). This attractive feature, which makes the frequency-domain approach so efficient in two dimensions, is lost with an iterative method. If we ignore storage requirements for a moment and just consider computational time, a frequency-domain formulation can compete with the time-domain approach only if the work involved in the iterations times the number of frequencies is significantly less than the work needed for performing all of the time steps in the time-domain method.

The iterative solution of the Helmholtz equation has been notoriously difficult in numerical analysis for a long time. A number of iterative solvers have been proposed, but generally they perform poorly on geophysics applications where the wavenumbers are relatively large and the velocities are heterogeneous (Elman and O'Leary, 1998; Heikkola et al., 1999; Larsson, 1999; Plessix and Mulder, 2003). Iterative solvers are usually based on Krylov subspace methods, such as the conjugate-gradient method for nonsymmetric matrices (Van der Vorst, 1992; Saad, 2003). They often converge rather slowly (sometimes not at all) and require a preconditioner to accelerate the convergence. This preconditioner should closely resemble the linear system of the original problem yet be relatively easy to invert, preferably in $O(n^d)$ operations with a number of operations O proportion to n^d , where n is the number of discretization points and d the spatial dimension (two or three).

The iterative multigrid method (Briggs et al., 2000; Trottenberg et al., 2000) achieves this $O(n^d)$ complexity for a wide class of elliptic problems — for instance, Poisson or diffusive problems that have only positive eigenvalues. The method also works for the Helmholtz equation with a complex wavenumber (Kim and Kim, 2002), which represents a wave equation with damping. Unfortunately, it fails when applied directly to the undamped wave equation because its discrete linear system has both positive and negative eigenvalues. However, Erlangga et al. (2004, 2006) propose using the multigrid solution of the damped wave equation as a preconditioner for the undamped one.

The choice of the amount of damping is a trade-off between a preconditioner that leads to fast convergence of the multigrid method and a preconditioner that causes the conjugate-gradient iterations to converge quickly. The latter requires a small amount of damping. The multigrid method will converge rapidly if the damping is large. Analytical and numerical results suggest that the strong damping of

$(1 - i0.5)k^2$ with k real and i imaginary units, is a good choice to obtain fast convergence of the multigrid method.

In this paper, we briefly review the iterative scheme based on the new preconditioner (Erlangga et al., 2004, 2006). We investigate the complexity of this new iterative solver and compare it with the complexity of the time-domain approach. Before concluding, we present 2D numeric results obtained with the new iterative solver. To validate our results, we compare them with the results obtained using the direct solver developed by George and Liu (1981), Marfurt and Shin (1989), and Plessix and Mulder (2004).

THEORY

The time-harmonic constant-density acoustic wave equation is represented by the Helmholtz equation:

$$\mathcal{L}p = f, \quad \mathcal{L} = -k^2 - \Delta, \quad (1)$$

where \mathcal{L} is the wave operator, p is the pressure field, and $k = \omega/c$ is the wavenumber, in which ω is the angular frequency and c is the velocity, which varies with position. The source term is denoted by f , and $-\Delta$ is minus the Laplace operator, which is positive, i.e., it is an operator with positive eigenvalues. If k^2 is larger than the smallest eigenvalue of $-\Delta$, which is the case in seismic applications, the Helmholtz operator is indefinite, meaning it has both positive and negative eigenvalues.

Equation 1 is discretized using the finite-difference scheme presented in Harari and Turkel (1995). This scheme is based on the fourth-order finite difference to improve accuracy, leading to the linear system

$$\mathbf{A}\mathbf{x} = \mathbf{b}. \quad (2)$$

Here, \mathbf{A} is a large but sparse matrix with complex values because of the absorbing boundary conditions. The vectors \mathbf{x} and \mathbf{b} are obtained from finite-difference discretization of p and f , respectively. If n is the number of discretization points in each coordinate of the computational grid and d is the spatial dimension, then \mathbf{A} has a size $n^d \times n^d$. With the natural ordering of the grid, \mathbf{A} has a bandwidth of n^{d-1} but only $O(n^d)$ nonzero elements. In 2D space a direct solver can be applied after reordering the grid with the nested dissection (George and Liu, 1981), and the LU decomposition has a complexity of $O(n^3)$ (Marfurt and Shin, 1989; Mulder and Plessix, 2004b). In 3D space, this approach is not feasible because the complexity of the direct solver is $O(n^6)$ and the storage required for \mathbf{L} and \mathbf{U} is $O(n^5)$. The alternative is an iterative solver.

Common iterative methods for indefinite systems are based on the Krylov subspace method (Saad, 2003). These methods are generalizations of the conjugate-gradient (CG) method. One of these generalizations is the biconjugated gradient stabilized (BI-CGSTAB) algorithm (Van der Vorst, 1992). In a Krylov method, such as the BI-CGSTAB algorithm, the method updates \mathbf{x}_i , starting from an initial guess \mathbf{x}_0 until the residual error $\|\mathbf{b} - \mathbf{A}\mathbf{x}_i\|$ is small enough. Without the preconditioner, this method converges slowly or not at all (Plessix and Mulder, 2003). By preconditioning, we solve the equivalent linear system as follows:

$$(\mathbf{A}\mathbf{M}^{-1})\mathbf{v} = \mathbf{b}, \quad \mathbf{v} = \mathbf{M}\mathbf{x}, \quad (3)$$

where \mathbf{M} is a preconditioning matrix. The solution of system 2 is $\mathbf{x} = \mathbf{M}^{-1}\mathbf{v}$. The algorithm for preconditioned BI-CGSTAB is found in Appendix A. The BI-CGSTAB algorithm needs two matrix-vector products per iteration. In our case, $\mathbf{A}\mathbf{M}^{-1}\mathbf{v}$, with \mathbf{v} being an intermediate vector, is required. In practice, \mathbf{M}^{-1} is not explicitly computed because it is too expensive. The matrix-vector product is evaluated in two steps. First, the system $\mathbf{M}\tilde{\mathbf{v}} = \mathbf{v}$ is solved; then, the matrix-vector product $\mathbf{A}\tilde{\mathbf{v}}$ is computed. Whereas the second step is trivial, the first step requires an efficient method to solve the linear system associated with the preconditioner. Therefore, the efficiency of the iterative method depends strongly on the choice of \mathbf{M} .

A good preconditioner should closely resemble \mathbf{A} and be easy to invert, preferably with an $O(n^d)$ computational cost because \mathbf{A} has $O(n^d)$ nonzero elements. Preconditioners based on the separation-of-variables method (Heikkola et al., 1999; Plessix and Mulder, 2003) or on an incomplete LU decomposition of \mathbf{A} (Erlangga et al., 2004) do not provide fast convergence and may fail in some cases. In typical seismic examples on a 500×500 grid with a spacing of about 10 m, these iterative methods generally do not converge after 2000 iterations for frequencies above 30 Hz.

To build the preconditioner, Erlangga et al. (2004, 2006) propose an operator of the preconditioner of the following form:

$$\mathcal{L}_p = - (1 - i0.5)k^2 - \Delta, \quad (4)$$

where i is the imaginary unit. The operator \mathcal{L}_p is a complex-shifted Laplace operator and corresponds to a strongly damped wave equation. The preconditioning matrix \mathbf{M} is obtained from finite-difference discretization of \mathcal{L}_p using, for example, a five-point finite-difference stencil. To compute the approximate inverse of \mathbf{M} appearing in the BI-CGSTAB algorithm, we use a multigrid method (Briggs et al., 2000; Trottenberg et al., 2000). The multigrid method performs well on this class of problems.

The multigrid method is an iterative scheme, useful for solving large-scale linear and nonlinear problems. It is widely used to obtain numeric solutions of Poisson's equation, and it has been generalized to parabolic (diffusion) and hyperbolic (flow) problems, although it does not always perform optimally for those cases.

The multigrid technique solves large-scale linear systems by using several grids simultaneously. The rationale is that the error between numeric approximation and exact solution consists of spatially rapidly varying or oscillatory components and slowly varying or smooth components. The oscillatory components can often be removed easily by some local relaxation scheme, called a smoother. If the problem is projected to a coarser grid, which usually has twice the grid spacing of the finer grid, the smooth components become more oscillatory. The projection of the residual $\mathbf{v} - \mathbf{M}\tilde{\mathbf{v}}$ to the coarser grid is called restriction. The matrix \mathbf{M} must be restricted to the coarse grid as well. Alternatively, a matrix-free implementation can be obtained if \mathbf{M} is a discretization of the differential operator and the same discretization is used on all grids.

If we assume for a moment that the solution of the coarse-grid problem is computed by a direct solver, the result must be interpolated back to the finer grid and added to the fine-grid solution. This interpolation is called prolongation. The interpolated coarse-grid solution is called the coarse-grid correction. Instead of a direct solver, the multigrid method can be used recursively on increasingly coarser grids until a level is reached where a direct solver is very cheap. By

recursive restriction to coarser and coarser grids, any error component will become oscillatory on one of these grids and can be removed easily by a smoother.

One must choose the order in which the coarser grids are visited and where the smoothing is applied. Usually, a number of pre-smoothing steps by a suitable smoother are performed before restricting the residual to the coarser grid. This can be repeated on the coarser grid until the coarsest is reached. There, a direct solver is applied and the solution is prolonged back to the finer grid. This can be followed by a number of postsmoothing steps before prolongation to the next finer grid. Once the finest grid has been reached and some postsmoothing steps have been carried out, a single multigrid iteration ends. Grids visited in this order are called a *V*-cycle (Figure 1a and b). A variant is a *W*-cycle, in which the number of coarse-grid corrections doubles on subsequently coarser grids. A less costly variant is an *F*-cycle (Figure 1c), in which the number of coarse-grid corrections increases by one on subsequently coarser grids. Details can be found in Briggs et al. (2000).

The iterative solver we propose to solve the undamped wave equation is a combination of one inner iteration with the multigrid method for solving the (damped) preconditioner system 4 and outer iterations with BI-CGSTAB.

COMPLEXITY ANALYSIS

This section presents a complexity analysis of the new iterative solver. The performance prediction is compared to the time-domain complexity (Mulder and Plessix, 2004b; Plessix and Mulder, 2004). With n_s being the number of shots and n_t being the number of time steps, the time-domain complexity is $n_s n_t O(n^d)$ ($d = 2$ or 3). With n_f being the number of frequencies, the overall complexity of the new iterative solver is $n_s n_f n_{it} O(n^d)$, where n_{it} is the number of iterations with BI-CGSTAB. The cost of the matrix-vector product is $O(n^d)$. The cost of approximately solving the preconditioner system is also $O(n^d)$. If we assume $n_s = n_t = O(n)$, $n_f = O(n)$, and $n_{it} = O(1)$, we see that the iterative frequency-domain and time-domain solvers have an $O(n^{d+2})$ complexity. The 2D frequency-domain direct solver has $n_f O(n^3)$ complexity if we use nested dissection reordering (George and Liu, 1981). This is independent of the number of shots. Once the LU decomposition of the reordered linear system is performed, shot computation requires $n_s n_f O(n^2 \log(n))$ operations. If $n_f = n_s = O(n)$ and $\log(n)$ is ignored, then the overall complexity of

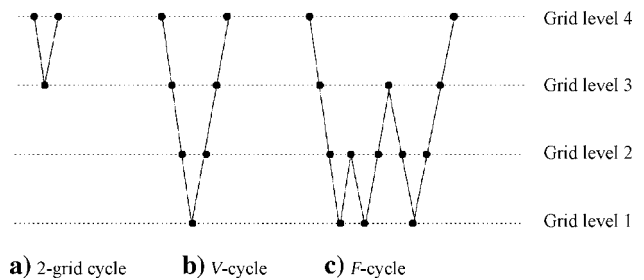


Figure 1. Three different cycles used in multigrid algorithms. The dashed lines denote grid levels from the finest (top level of each diagram) to the coarsest (level 1 grid). The filled circles represent smoothing, with the circle on level 1 representing the direct solution on the coarsest grid. Each descending line (\backslash) denotes restriction, while each ascending line ($/$) denotes prolongation.

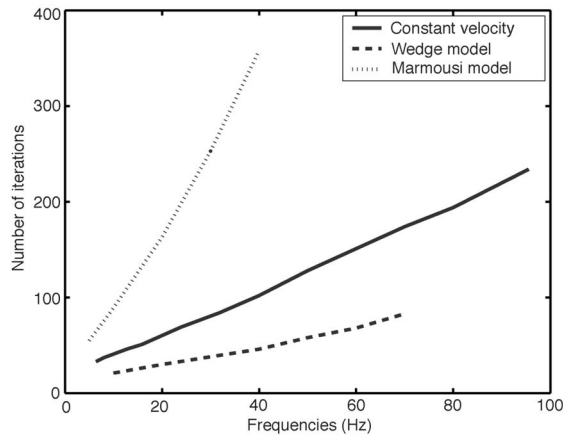


Figure 2. The number of iterations of the iterative solver versus frequency. The constant-velocity model has a velocity of 1500 m/s. The three-layer wedge model varies from 1500–3000 m/s. The Marmousi model is plotted in Figure 4.

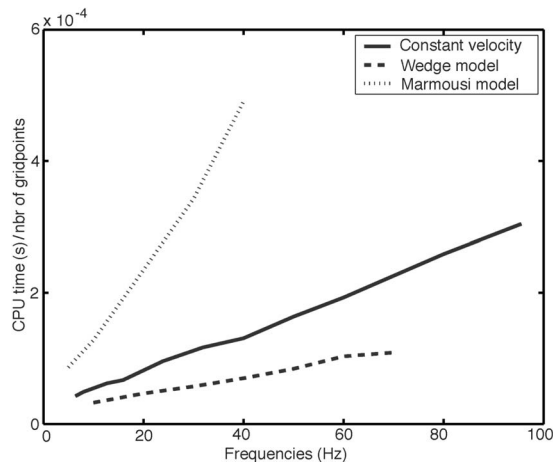


Figure 3. The computation time normalized by the number of gridpoints versus frequency. The constant-velocity model has a velocity of 1500 m/s. The three-layer wedge model varies from 1500–3000 m/s. The Marmousi model is plotted in Figure 4.

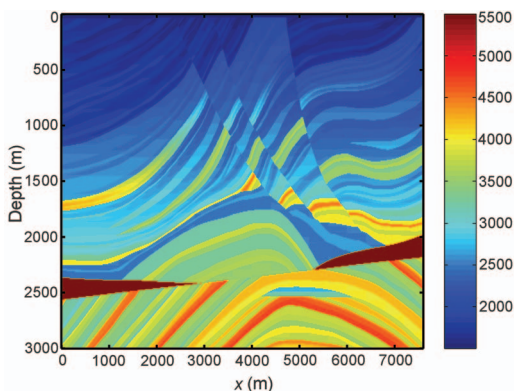


Figure 4. The Marmousi velocity model used in the numeric experiments.

the 2D frequency-domain direct solver and the 2D time-domain solver is $O(n^4)$.

For migration problems with n around 1000 and n_s around 200, the frequency-domain method is about one order of magnitude faster than the time-domain approach in two dimensions. The multiplicative constant for the frequency-domain direct solver is apparently far smaller than the one for the time-domain solver. The reason is that the number of frequencies needed is smaller than the number of time steps imposed by the stability condition of the time-domain solver (Mulder and Plessix, 2004a). Also, coarser grids can be used for lower frequencies. Furthermore, n_s is close to n in these applications. This motivates further investigation of the frequency-domain approach in three dimensions.

To investigate the behavior of the new frequency-domain iterative solver, we conducted a series of 2D simulations on three different models: a constant model, a three-layer wedge model with velocities varying from 1500 m/s in the first layer to 3000 m/s in the third layer, and the Marmousi model. The number of iterations versus frequency and the computational time normalized by the number of gridpoints versus frequency are plotted in Figures 2 and 3, respectively. In these experiments, the number of gridpoints per wavelength is kept constant, so the number of gridpoints n in each coordinate is proportional to the frequency used. Figure 2 shows the number of iterations varies linearly with the frequency. This behavior is suboptimal. Nevertheless, the numeric results suggest that the preconditioned iterative method proposed by Erlangga et al. (2004, 2006) is predictable and robust. In Figure 3, the computational time divided by the number of gridpoints increases linearly with frequency. From a complexity point of view, this result shows that the frequency-domain iterative solver is less efficient asymptotically than the frequency-domain direct solver and the time-domain approach.

In three dimensions, the use of a direct solver is not an option for problems of realistic size. The time-domain method has a complexity $n_s O(n^4)$, and an extrapolation of the above 2D results suggests that the iterative solver for the frequency-domain problem has a complexity of $n_s O(n^5)$ when $n_{ii} = O(n)$. Still, only the precise values of the constants involved in the cost estimates reveal the efficiency of this new iterative solver. This will require a 3D implementation of the iterative method.

NUMERICAL EXPERIMENTS

To illustrate this method, we compare the wavefield computed with the iterative solver to the one obtained by a direct frequency-domain solver. The example is based on the Marmousi model (Bourgeois et al., 1990). The velocity model used in the numerical experiments is plotted in Figure 4. The source is located at the surface ($x = 6000$ m) just below the absorbing boundary. The model is discretized on a regular grid with $dx = dz = 4$ m. The pressure wavefield is recorded by 1900 receivers spaced 4 m apart. The real part of the wavefield at 30 Hz, obtained with the new iterative solver, is displayed in Figure 5; the solution from the direct solver is depicted in Figure 6. In Figure 7, we compare the solutions of the new iterative solver and the direct solver at x positions of 5.4 and 4 km, respectively.

We can conclude that the frequency-domain results obtained by the new iterative solver and the direct solver are the same. The iterative solver actually converged at 30 Hz in this example, whereas earlier preconditioners based on incomplete LU decomposition or separation of variables did not. Figure 8 shows the time-domain wave-

field obtained by the iterative solver; the result from the direct solver is plotted in Figure 9. The frequency of the source varies from 0–60 Hz. In Figure 10 we compare the results obtained by the iterative solver with the results obtained by the direct solver for traces at 5.8, 5, 4.5, 4, 3.5, and 3 km, respectively. From these figures, we can also conclude that the time-domain numerical results obtained by the iterative solver and by the direct solver are in very good quantitative agreement.

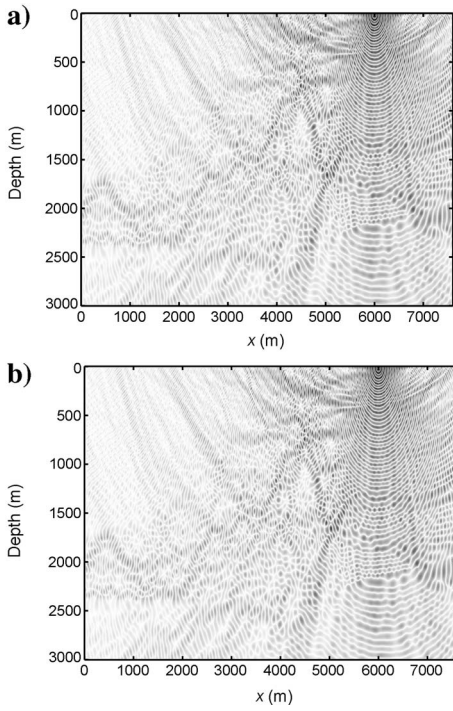


Figure 5. The real part of the wavefield at 30 Hz obtained by (a) the new iterative solver for the Marmousi model and (b) the direct solver for the Marmousi model.

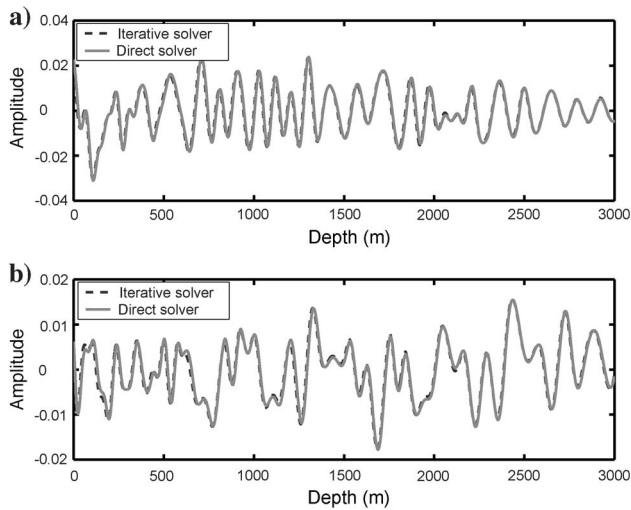


Figure 6. Comparison between the frequency-domain results from Figure 5 at x is (a) 5400 m and (b) 4000 m. The solid line is the numeric result obtained by the direct solver; the dashed line is the result obtained by the iterative solver.

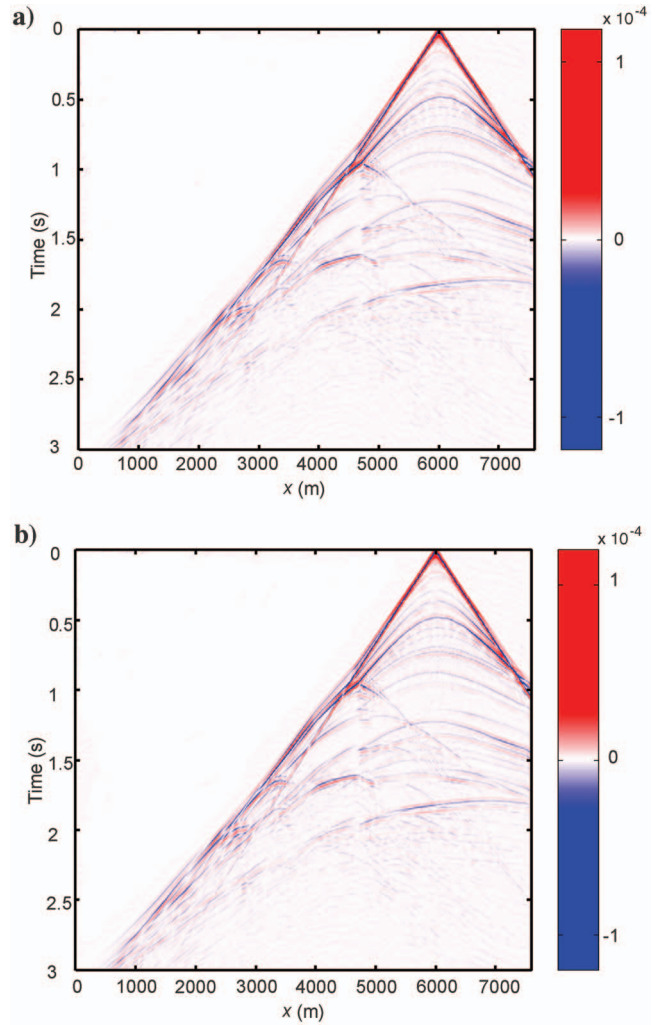


Figure 7. The time-domain results using (a) the new iterative solver and (b) the direct solver.

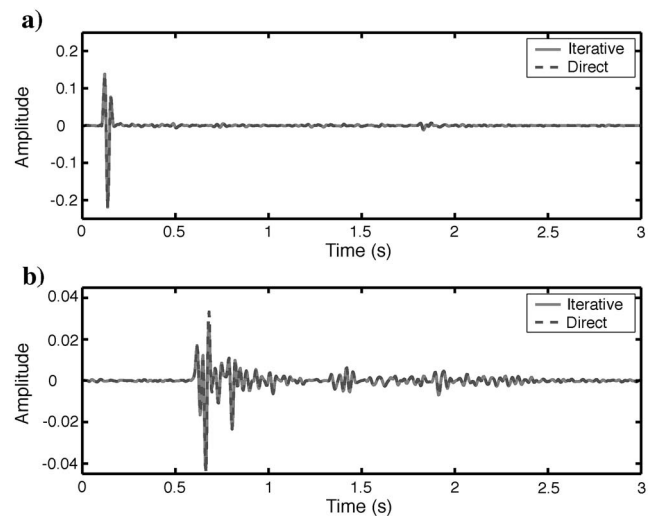


Figure 8. Comparison in time of the response obtained by the direct solver and the iterative solver at x is (a) 5800 m and (b) 5000 m. The solid line is the numeric result obtained by the iterative solver; the dashed line is the result obtained by the direct solver.

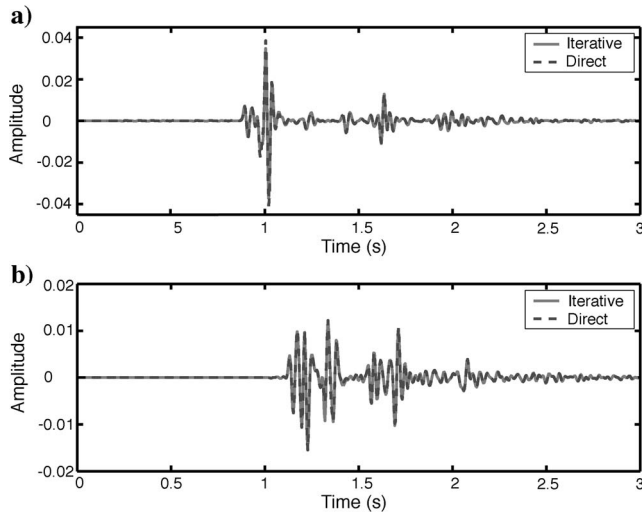


Figure 9. Same as in Figure 8, but at x is (a) 4500 m and (b) 4000 m.

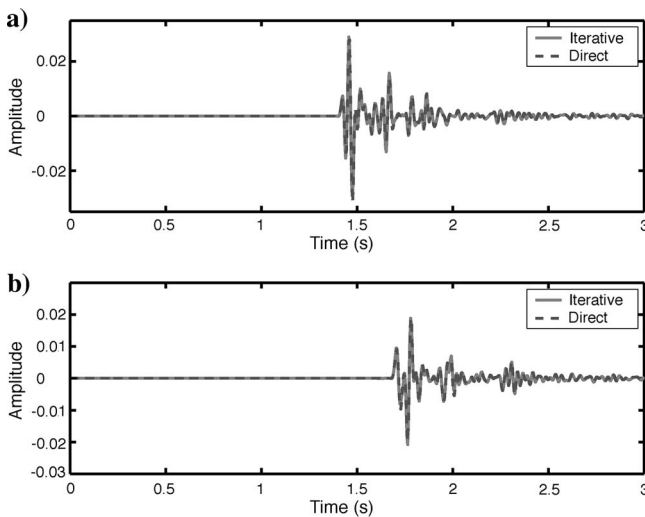


Figure 10. Same as in Figure 8, but at x is (a) 3500 m and (b) 3000 m.

CONCLUSIONS

The complexity of the new frequency-domain iterative solver of the undamped wave equation developed in Erlangga et al. (2004, 2006) has been investigated and compared to its time-domain counterpart. The iterative solver is based on a preconditioned BI-CGSTAB method. The preconditioner is based on a strongly damped wave operator. The iterative solver obtains its efficiency by using the multigrid method for the approximate inversion of the preconditioner. To obtain reasonably fast convergence of the multigrid method, a fairly large amount of damping corresponding to a quality factor of two is applied in the preconditioner. The resulting iterative solver is a combination of an inner iteration with the multigrid method and outer iterations with BI-CGSTAB.

Complexity estimates, based on 2D simulations, show that the new solver is suboptimal because the number of iterations is proportional to frequency. Nevertheless, the iterative solver appears to be robust, and convergence was reached in complex models for relatively high frequencies. This result is an improvement over other

iterative solvers that generally fail at high frequencies in complex earth models. Asymptotically, the time domain is faster by an order n , with n being the number of gridpoints in each coordinate. In practice, the actual constants in the complexity play an important role. The usefulness of the iterative method for 3D migration still needs assessment.

We have applied the new iterative solver to the Marmousi model and have validated our numeric results by comparing them to the results obtained using a direct solver. We conclude that the time-domain numeric results obtained by the new iterative solver and the direct solver are in very good quantitative agreement.

ACKNOWLEDGMENTS

This work was part of a joint project between Philips Electronics Nederland B.V. and Shell International Exploration and Production B.V. on rigorous modeling of 3D wave propagation in inhomogeneous media for geophysical and optical problems. This project was supported financially by the Dutch Ministry of Economic Affairs, project BTS01044.

APPENDIX A

THE BI-CGSTAB ALGORITHM

The preconditioned BI-CGSTAB algorithm (Van der Vorst, 1992) for solving the linear system $\mathbf{Ax} = \mathbf{b}$ with preconditioning matrix \mathbf{M} is given as follows:

Choose an initial guess \mathbf{x}_0 ; $\mathbf{r}_0 = \mathbf{b} - \mathbf{Ax}_0$.

Choose $\tilde{\mathbf{r}}_0$ such that $(\tilde{\mathbf{r}}_0 \mathbf{r}_0) \neq 0$, e.g., $\tilde{\mathbf{r}}_0 = \mathbf{r}_0$.

$$\rho_{-1} = \alpha_{-1} = \omega_{-1} = 1$$

$$\mathbf{v}_{-1} = \tilde{\mathbf{v}}_{-1} = \mathbf{0}$$

for $i = 0, 1, 2, \dots$

$$\rho_i = (\tilde{\mathbf{r}}_0, \mathbf{r}_i); \beta_{i-1} = (\rho_i / \rho_{i-1})(\alpha_{i-1} / \omega_{i-1})$$

$$\mathbf{v}_i = \mathbf{r}_i + \beta_{i-1}(\mathbf{v}_{i-1} - \omega_{i-1}\tilde{\mathbf{v}}_{i-1})$$

$$\mathbf{y} = \mathbf{M}^{-1}\mathbf{v}_i$$

$$\tilde{\mathbf{v}}_i = \mathbf{Ay}$$

$$\alpha_i = \rho_i / (\tilde{\mathbf{r}}_0, \tilde{\mathbf{v}}_i)$$

$$\mathbf{s} = \mathbf{r}_i - \alpha_i \tilde{\mathbf{v}}_i$$

$$\mathbf{z} = \mathbf{M}^{-1}\mathbf{s}$$

$$\mathbf{t} = \mathbf{Az}$$

$$\omega_i = (\mathbf{t}, \mathbf{s}) / (\mathbf{t}, \mathbf{t})$$

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{y} + \omega_i \mathbf{z}$$

$$\mathbf{r}_{i+1} = \mathbf{s} - \omega_i \mathbf{t}$$

end.

Here, \mathbf{M} is the preconditioning matrix defined in equation 4, and (\cdot, \cdot) denotes the inner product. In this algorithm, we need two matrix-vector products. The algorithm carries out the BI-CGSTAB procedure for the explicitly preconditioned linear system $\mathbf{AM}^{-1}\mathbf{v} = \mathbf{b}$. The residual \mathbf{r}_i corresponds to the original system $\mathbf{Ax}_i = \mathbf{b}$.

REFERENCES

- Bamberger, A., B. Enquist, L. Halpern, and P. Joly, 1988, Parabolic wave equation approximation in heterogeneous media: Society for Industrial and Applied Mathematics (SIAM) Journal of Applied Mathematics, **48**, 99–128.
- Biondi, B., and G. Palacharla, 1996, 3-D prestack migration of common-azimuth data: Geophysics, **61**, 1822–1832.
- Bourgeois, A., M. Bourget, P. Lailly, M. Poulet, P. Ricarte, and R. Versteeg, 1990, Marmousi, model and data: 52nd Annual International Meeting, EAEG, Extended Abstracts, 5–16.
- Briggs, W. L., V. E. Henson, and S. F. McCormick, 2000, A multigrid tutorial: Society for Industrial and Applied Mathematics.
- Claerbout, J. F., 1985, Imaging the earth's interior: Blackwell Scientific Publications, Inc.
- Collino, F., and P. Joly, 1995, Splitting of operators, alternate directions, and paraxial approximations for the three-dimensional wave equation: Society of Industrial and Applied Mathematics (SIAM) Journal on Scientific Computing, **16**, 1019–1048.
- Elman, H. C., and D. P. O'Leary, 1998, Efficient iterative solution of the three-dimensional Helmholtz equation: Journal of Computational Physics, **142**, 163–181.
- Erlangga, Y. A., C. Vuik, and C. Oosterlee, 2004, On a class of preconditioners for the Helmholtz equation: Applied Numerical Mathematics, **50**, 409–425.
- , 2006, A novel multigrid based preconditioner for heterogeneous Helmholtz problems: Society of Industrial and Applied Mathematics (SIAM) Journal on Scientific Computing, **27**, 1471–1492.
- George, A., and J. W. Liu, 1981, Computer solution of large sparse positive definite systems: Prentice-Hall, Inc.
- Harari, I., and E. Turkel, 1995, Accurate finite difference methods for time-harmonic wave propagation: Journal of Computational Physics, **119**, 252–270.
- Heikkola, E., Y. A. Kuznetsov, and K. N. Lipnikov, 1999, Fictitious domain methods for the numerical solution of three-dimensional acoustic scattering problems: Journal of Computational Acoustics, **7**, 161–183.
- Jin, S., R. S. Wu, and G. Peng, 1998, Prestack depth migration using a hybrid pseudoscreen propagator: 68th Annual International Meeting, SEG, Expanded Abstracts, 1819–1822.
- Kim, S., and S. Kim, 2002, Multigrid simulation for high-frequency solutions of the Helmholtz problem in heterogeneous media: Society of Industrial and Applied Mathematics (SIAM) Journal on Scientific Computing, **24**, 684–701.
- Larsson, E., 1999, A domain decomposition method for the Helmholtz equation in a multilayer domain: Society of Industrial and Applied Mathematics (SIAM) Journal on Scientific Computing, **20**, 1713–1731.
- Marfurt, K. J., 1984, Accuracy of finite-difference and finite-element modeling of the scalar and elastic wave equations: Geophysics, **49**, 533–549.
- Marfurt, K. J., and C. S. Shin, 1989, The future of iterative modeling in geophysical exploration, in E. Eister, ed., Supercomputers in seismic exploration: Pergamon Press Inc., 203–228.
- Mulder, W. A., and R.-E. Plessix, 2004a, How to choose a subset of frequencies in frequency-domain finite-difference migration: Geophysical Journal International, **158**, 801–812.
- , 2004b, One-way and two-way wave-equation migration: Geophysics, **69**, 1491–1504.
- Plessix, R.-E., and W. A. Mulder, 2003, Separation of variables as a preconditioner for an iterative Helmholtz solver: Applied Numerical Mathematics, **44**, 385–400.
- , 2004, Frequency-domain finite-difference amplitude-preserving migration: Geophysical Journal International, **157**, 975–987.
- Saad, Y., 2003, Iterative methods for sparse linear system, 2nd ed.: Society for Industrial and Applied Mathematics.
- Trottenberg, U., C. W. Oosterlee, and A. Schüller, 2000, Multigrid: Academic Press Inc.
- Van der Vorst, H. A., 1992, BI-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems: Society for Industrial and Applied Mathematics (SIAM) Journal on Scientific and Statistical Computing, **13**, 631–644.
- Yoon, Y., K. J. Marfurt, and W. Starr, 2004, Challenges in reverse-time migration: 74th Annual International Meeting, SEG, Expanded Abstracts, 1057–1060.
- Yoon, K., C. Shin, S. Suh, L. R. Lines, and S. Hong, 2003, 3D reverse-time migration using the acoustic wave equation: An experience with the SEG/EAGE data set: The Leading Edge, **22**, 38–41.