GEOPHYSICS, VOL. 72, NO. 5 (SEPTEMBER-OCTOBER 2007); P. SM185–SM194, 12 FIGS.
10.1190/1.2738849

A Helmholtz iterative solver for 3D seismic-imaging problems

R.-E. Plessix

ABSTRACT

A preconditioned iterative solver for the 3D frequency-domain wave equation applied to seismic problems is evaluated. The preconditioner corresponds to an approximate inverse of a heavily damped wave equation deduced from the (undamped) wave equation. The approximate inverse is computed with one multigrid cycle. Numerical results show that the method is robust and that the number of iterations increases roughly linearly with frequency when the grid spacing is adapted to keep a constant number of discretization points per wavelength. To evaluate the relevance of this iterative solver, the number of floating-point operations required for two imaging problems are roughly evaluated. This rough estimate indicates that the time-domain migration approach is more than one order of magnitude faster. The full-waveform tomography, based on a least-squares formulation and a scale separation approach, has the same complexity in both domains.

INTRODUCTION

The common workflow to build a depth-impedance image from multishot, multireceiver data principally consists of two steps: (1) the velocity analysis, which aims to retrieve the low spatial frequencies, namely, a smooth velocity field; and (2) the depth migration, which aims to build a depth image of the impedance contrasts assuming a smooth velocity field. Generally, the velocity analysis is based on picked reflected events and is solved with a tomography algorithm. Seismic data are then migrated with the smoothed velocity found during the velocity analysis. For large 3D cases, Kirchhoff or Born migrations, based on high-frequency assumptions, were and are still commonly used. In the last 10 years, one-way, finite-difference solutions of the wave equation became popular for migration because of its finite-frequency aspect. Finite-frequency solutions of the wave equations are essential to correctly estimate the wavefront when rapid lateral variations exist, for instance, at salt interfaces. Unfortunately, one-way solutions, especially the amplitudes, become inaccurate at large incidence angles.

To overcome the high-frequency and one-way limits, two-way solutions of the wave equation at finite frequencies are needed. One can solve the two-way wave equation either in the time domain or in the frequency domain. Traditionally, time-domain solutions are obtained with a marching algorithm in time, which leads to an explicit scheme. The frequency-domain approach requires the solution of a large linear system. Marfurt (1984) discusses the advantages of each approach in 2D. In 2D, the linear system of the frequency-domain scheme can be solved with a direct solver, and the frequency-domain multishot migration algorithm is about one order of magnitude faster than its time-domain counterpart, because the LU decomposition of the matrix of the system has to be performed only once per frequency. Unfortunately, with realistic 3D cases, the linear system may be difficult to solve with a direct solver, because of the high memory requirement. This difficulty may limit the advantages of the frequency-domain formulation if no robust iterative solver of the frequency-domain wave equation is available. In fact, the existing 3D two-way, finite-difference migration algorithms are all based on a time-domain formulation (Yoon et al., 2003, 2004).

One approach to solve large 3D problems is based on domain decomposition; see, for instance, Heikkola et al. (1999) or Larson (1999). This approach may use a direct solver (Operto et al., 2007) or an iterative solver in each domain. Several authors worked on Krylov-based iterative solvers for the Helmholtz equation, the frequency-domain wave equation with constant density; see, for instance, Elman et al. (2001), Plessix and Mulder (2003), Erlangga et al. (2004, 2006), Riyanti et al. (2006), and the references mentioned in those papers. These methods differ mainly by the preconditioner they use to accelerate the convergence. The preconditioner is crucial because, without it, the Krylov-based iterative methods require more than 1000 iterations for a medium-sized 2D problem and more than 10,000 iterations for a medium-sized 3D problem. For geophysical applications, the preconditioner proposed by Erlangga et al. (2004, 2006), may be attractive. The method consists of a classic iterative solver for indefinite systems, for instance, the BICGSTAB method (Van der Vorst, 1992), and a preconditioner based on a heavi-

Manuscript received by the Editor October 7, 2006; revised manuscript received December 13, 2006; published online August 23, 2007.
Shell International E&P, Rijswijk, the Netherlands. E-mail: ReneEdouard.Plessix@shell.com.

Reprinted from Geophysics, 72, no. 5, SM185–SM194. © 2007 Society of Exploration Geophysicists

ly damped wave equation to accelerate the convergence. The preconditioner does not change the solution, and the preconditioned iterative method gives the solution to the undamped wave equation (Chen, 2005). The preconditioner based on a heavily damped wave equation is approximately solved with one multigrid cycle. The use of one multigrid cycle is efficient because its computational cost is equivalent to the computation of a few matrix-vector products (Briggs et al., 2000). Therefore, the preconditioner has an optimal complexity. Numerical results show that the method has a $O(n^3)$ complexity (i.e., is proportional to n^3) in 2D (Riyanti et al., 2006), with n the number of grid points per direction. In this paper, I present 3D results based on a slightly different implementation of the multigrid solver to facilitate high-order discretization schemes. The multigrid method is implemented with a symmetric Gauss-Seidel smoothing, a standard full-weighting coarsening, a trilinear interpolation, and a matrix-free implementation (Briggs et al., 2000). In this way, I can compute realistic 3D models containing up to 80 million grid points on a single computer unit with 16 GB of memory. I then discuss the relevance of this approach for two imaging problems based on a least-squares formulation (Tarantola, 1987). I consider the migration of reflection data and the full-waveform tomography of long-offset data. The migration is seen as the gradient of the least-squares misfit function (Lailly, 1983) and aims to give a high-resolution impedance image. This is achieved by processing a large frequency band. Because the wavenumber redundancy is limited as a result of the short aperture of reflection data, a large number of frequencies needs to be modeled, which tends to favor the time-domain formulation. One time-domain wave-equation solution contains all of the frequencies of the band, whereas many frequency-domain wave-equation solutions must be computed within the frequency band (Mulder and Plessix, 2004). With short-to-medium offset data, the least-squares formulation is not suited to retrieve the long wavelengths of the velocity because of local minima. However, the full-waveform tomography based on an iterative minimization of the least-squares misfit functional seems to give satisfactory results with long offset data and a reasonable starting model (Pratt, 1999; Operto et al., 2004). The minimization of the least-squares misfit is achieved per frequency and from low to high frequencies (Pratt, 1999). Because long-offset data contain a significant redundancy in the wavenumber domain, it is possible to invert only a limited number of frequencies, which tends to favor the frequency-domain formulation.

The outline of the paper is the following. In the first part, the iterative solver is described. In the second part, two 3D examples are shown. In the third part, by roughly counting the number of floating-point operations, the complexity of the frequency-domain iterative solver is compared to that of a time-domain solver for the least-squares migration and the full-waveform tomography.

METHOD

The computation of the pressure wavefield in the frequency domain requires the solution of a linear system after discretization of the wave equation. The accuracy of the solution depends on the order of the scheme and the number of discretization points per wavelength. In a smooth velocity field, a higher-order scheme generally requires fewer points per wavelength without losing accuracy, because the partial derivatives are more accurately approximated. This reduces the memory and computational cost. The higher-order schemes generally lead to a larger finite-difference stencil, which means that sharp contrasts will be smoothed. In practice, the scheme orders are 2, 4, 6, or 8.

Preconditioned iterative solver

The acoustic constant density frequency-domain wave equation corresponds to the Helmholtz equation:

$$-k^2(x)u(x,\omega) - \Delta u(x,\omega) = s(x,\omega), \tag{1}$$

with u the pressure field, s the source function, ω the angular frequency, k the wavenumber, $k = \omega/v$ with v the velocity, and x the spatial coordinates. Δ is the Laplacian.

A centered finite-difference discretization with a $2J$-order scheme in space gives

$$\sum_{j=1}^{J}\left[\frac{a_j^J}{h^2}(u_{l-j,m,p} + u_{l+j,m,p}) + \frac{b_j^J}{h^2}(u_{l,m-j,p} + u_{l,m+j,p}) + \frac{c_j^J}{h^2}(u_{l,m,p-j} + u_{l,m,p+j})\right] + \left(\frac{d^J}{h^2} - k_{l,m,p}^2\right)u_{l,m,p} = s_{l,m,p}, \tag{2}$$

where h is the spacing of the cubic grid; a_j^J, b_j^J, c_j^J, and d^J are the coefficients of the scheme; and $u_{l,m,p}$, $s_{l,m,p}$, and $k_{l,m,p}$ are the discrete values of the pressure field, the source function, and the wavenumber, respectively. Equation 2 is valid for $i,j,k \in [J,n_x - J] \times [J,n_y - J] \times [J,n_z - J]$, with n_x, n_y, and n_z being the number of discretization points in each direction. Outside, the field values are set to zero. In practice, absorbing boundary layers are added to the domain to mimic an infinite domain. In the boundary layers, a complex part to the wavenumber is added; this means that the wavefield is damped. Therefore, the pressure field can be assumed to be zero close to the boundaries.

In a matrix form, equation 2 reads

$$\mathbf{A}(\mathbf{k})\mathbf{u} = \mathbf{s}, \tag{3}$$

where $\mathbf{u}$ and $\mathbf{s}$ are now the pressure and source vectors, respectively. $\mathbf{A}$ is generally indefinite because the left-hand side of equation 1 is the sum of a positive operator $(-\Delta)$ and a negative operator $(-k^2)$. With n being the number of discretization points in one spatial direction, $\mathbf{A}$ is an $n^3 \times n^3$ sparse matrix. For realistic geophysical examples, n ranges between 200 and 1000.

The traditional direct method to solve equation 3 is a Gaussian elimination technique, namely, an $\mathbf{LU}$ decomposition: $\mathbf{A} = \mathbf{LU}$, with $\mathbf{L}$ a lower triangular matrix and $\mathbf{U}$ an upper triangular matrix. $\mathbf{A}$ is a very sparse matrix with approximately $(2J + 1)n^3$ nonzero elements. With the natural ordering of the grid, the nonzero elements are situated along some diagonals of the matrix. The largest distance between the main (central) diagonal and the off-diagonals is Jn^2; this is the bandwidth of $\mathbf{A}$. Here, I do not consider compact fourth-order schemes that reduce the bandwidth of the matrix (Jo et al., 1996; Operto et al., 2007). An $\mathbf{LU}$ decomposition will generally introduce fill-in within the bandwidth, which means that $\mathbf{L}$ will have approximately Jn^2n^3 nonzero elements. With a 3D medium-size problem, this is not computationally affordable. We should notice that the memory requirement is theoretically reduced to $O(n^2log(n))$ in 2D and $O(n^4)$ in 3D with a nested dissection reordering and a second-order scheme or a compact fourth-order scheme (George and Liu, 1981; Ashcraft and Liu, 1998). Because of the high memory and computational requirements of the direct solver in 3D, to solve the linear system 3, a

Krylov-based iterative method, for instance, the BICGSTAB method (Van der Vorst, 1992), is natural to try. (The BICGSTAB is an extension of the classic linear conjugate-gradient method for indefinite matrices). The efficiency of the solver relies on the preconditioner. The right preconditioned system, with the preconditioner **M**, is

$$\mathbf{A}(\mathbf{k})\mathbf{M}^{-1}\mathbf{v} = \mathbf{s}, \quad \text{with } \mathbf{M}\mathbf{u} = \mathbf{v}. \tag{4}$$

Within the iterations, the matrix-vector product $\mathbf{A}(\mathbf{k})\mathbf{M}^{-1}\mathbf{v}$ needs to be evaluated. This is carried out in two steps: first, system $\mathbf{M}\mathbf{u} = \mathbf{v}$ is solved, then the matrix-vector product $\mathbf{A}(\mathbf{k})u$ is computed. In order to efficiently solve system $\mathbf{M}\mathbf{u} = \mathbf{v}$, Erlangga et al. (2004, 2006) proposed to use one multigrid cycle (Briggs et al., 2000; Trottenberg et al., 2000) and to consider $\mathbf{M} = \mathbf{A}(\mathbf{k}(\beta_r + i\beta_i))$, with $i = \sqrt{-1}$ and β_r and β_i two (positive) real numbers. One multigrid cycle has the same complexity as a matrix-vector product. Therefore, solving $\mathbf{M}\mathbf{u} = \mathbf{v}$ with one multigrid cycle leads to a preconditioner with optimal complexity.

The choice of β_r and β_i should be such that **M** is close enough to **A** and that the multigrid method is applicable to **M**. The first condition means that β_r should be close to 1, and β_i should be as small as possible. The second condition means that β_i should be large enough. It appears that β_r close to 1 and β_i between 0.25 and 1 generally lead to a correct choice. M corresponds to the discretization of the wave equation with a wavenumber equal to $\mathbf{k}(\beta_r + i\beta_i)$, namely, the discretization of a heavily damped wave equation. Indeed, assuming k constant, the solution of the damped wave equation is $ae^{ik(\beta_r+i\beta_i)\|x\|}$, with a the amplitude (geometrical divergence) term and $\|.\|$ the L2-norm. The term β_i induces an extra damping equal to $e^{-k\beta_i\|x\|}$. Over a wavelength, i.e., for $\|x\| = 2\pi/k$, the extra damping is 0.04 with $\beta_i = 0.5$; 96% of the energy is lost over one wavelength because of this extra damping. The heavily damped wave equation almost behaves like a diffusive equation, since the energy hardly propagates after a wavelength. This explains why the multigrid method can be used on **M**. A 1D numerical example is given in Appendix A.

The iterative solver to compute **u**, solution of equation 3, is implemented as follows:

1) Solve $\mathbf{A}\mathbf{M}^{-1}\mathbf{v} = \mathbf{s}$ with the BICGSTAB iterative method, s being a given source term:

   a) Initiate the iterations with $\mathbf{v} = 0$.

   b) Update **v** with one BICGSTAB iteration. During this step, one needs to evaluate twice a matrix-vector product of the form $\mathbf{A}\mathbf{M}^{-1}\mathbf{y}$ with **y** a vector determined by the BICGSTAB method. This requires the solution of the system $\mathbf{M}\mathbf{z} = \mathbf{y}$ and the evaluation of $\mathbf{A}\mathbf{z}$.

   c) If $\|\mathbf{A}\mathbf{M}^{-1}\mathbf{v} - \mathbf{s}\| < \varepsilon\|\mathbf{s}\|$ (in the examples, $\varepsilon = 10^{-5}$), go to 2; otherwise, repeat step (b).

2) Compute $\mathbf{u} = \mathbf{M}^{-1}\mathbf{v}$. The requires the solution of the system $\mathbf{M}\mathbf{u} = \mathbf{v}$.

The solution **u** satisfies $\|\mathbf{A}(\mathbf{k})\mathbf{u} - \mathbf{s}\| < \varepsilon\|\mathbf{s}\|$ and is then the solution of the undamped equation 3. The preconditioner does not change the solution of the initial system (Chen, 2005). Thus, the damping of the preconditioner does not change the solution of equation 3.

Multigrid cycle

One obtains an efficient iterative method when system $\mathbf{M}\mathbf{z} = \mathbf{y}$ is quickly evaluated and when **M** is close to **A**. **M** contains $O(n^3)$ elements because it corresponds to the discretization of a damped wave equation. One multigrid cycle requires $O(n^3)$ operations (Briggs et al., 2000). Therefore, the method proposed by Erlangga et al. (2004, 2006) has an optimal complexity for the evaluation of the preconditioner. This is why determining the inverse of the preconditioner with one multigrid cycle is attractive.

I implemented a matrix-free algorithm in order to handle large-scale problems and any discretization order. This is slightly different from the implementation presented in Erlangga et al. (2004, 2006) and Riyanti et al. (2006). A matrix-free algorithm requires less memory, because the coefficients of the matrix are not stored. It also allows me to easily change the discretization of the matrix without modifying the algorithm itself; only a few subroutines have to be rewritten. In my implementation, the boundary conditions are Dirichlet boundary conditions, and absorbing layers are added to mimic an infinite space.

The idea of multigrid is to estimate the oscillatory part of the solution on the fine grid and to estimate the smooth part on a coarser grid. In the preconditioned iterative method, one needs to solve

$$\mathbf{M}\mathbf{z} = \mathbf{y}, \tag{5}$$

with **y** a given vector. The discretization of the damped wave equation, equation 2, is denoted $\mathbf{M}^h$ after replacing **k** by $\mathbf{k}(\beta_r + i\beta_i)$. I then rewrite system 5 as

$$\mathbf{M}^h\mathbf{z}^h = \mathbf{y}^h, \tag{6}$$

with $\mathbf{M} = \mathbf{M}^h$, $\mathbf{y}^h = \mathbf{y}$, and $\mathbf{z} = \mathbf{z}^h$. The superscript h means that the quantities are related to the grid with the spacing h.

An approximation, $\mathbf{w}^h$, of $\mathbf{z}^h$, with the smoother operator, $S^h(\mathbf{M}^h, \mathbf{y}^h, \mathbf{z}_0^h)$, is computed as

$$\mathbf{w}^h = S^h(\mathbf{M}^h, \mathbf{y}^h, \mathbf{z}_0^h). \tag{7}$$

$\mathbf{z}_0^h$ is an initial guess for $\mathbf{z}^h$, generally $\mathbf{z}_0^h = 0$. In the implementation, the smoother is one iteration of the symmetric Gauss-Seidel method (Briggs et al., 2000). Here, $\mathbf{w}^h$ contains an approximation of the oscillatory part of the solution, but the smooth part of the solution is difficult to obtain from the Gauss-Seidel method. The residual, $\mathbf{r}^h = \mathbf{y}^h - \mathbf{M}^h\mathbf{w}^h$, satisfies

$$\mathbf{M}^h\mathbf{e}^h = \mathbf{r}^h, \quad \text{with } \mathbf{e}^h = \mathbf{z}^h - \mathbf{w}^h. \tag{8}$$

When $\mathbf{w}^h$ approximates the oscillatory part of $\mathbf{z}^h$, $\mathbf{r}^h$ and $\mathbf{e}^h$ have mainly smooth variations; $\mathbf{r}^h$ and $\mathbf{e}^h$ should then be correctly approximated on a coarser grid with a $2h$ spacing. With $\mathbf{C}_h^{2h}$ the coarsening operator from the fine grid to the coarse grid and $\mathbf{P}_{2h}^h$ the prolongation operator from the coarse grid to the fine grid, $\mathbf{M}^h\mathbf{e}^h = \mathbf{r}^h$ becomes

$$\mathbf{C}_h^{2h}\mathbf{M}^h\mathbf{P}_{2h}^h\mathbf{e}^{2h} = \mathbf{r}^{2h} \quad \text{with } \mathbf{r}^{2h} = \mathbf{C}_h^{2h}\mathbf{r}^h \quad \text{and} \quad \mathbf{e}^h = \mathbf{P}_{2h}^h\mathbf{e}^{2h}. \tag{9}$$

In the implementation, the coarsening operator is the full-weighting operator and the prolongation operator is the trilinear interpolation (Briggs et al., 2000).

The matrix-free implementation consists of approximating $\mathbf{C}_h^{2h}\mathbf{M}^h\mathbf{P}_{2h}^h$ by $\mathbf{M}^{2h}$, the discretization of the damped wave equation on the grid with a $2h$ spacing. So, one must solve on the coarse grid,

$$\mathbf{M}^{2h}\mathbf{e}^{2h} = \mathbf{r}^{2h}. \tag{10}$$

One can solve this equation with the same approach as the one described for system 6, namely, by applying recursively the two-grid approach. The interpolation of $\mathbf{e}^{2h}$ can add artificial oscillations in $\mathbf{e}^h$. One removes those oscillations by applying the smoother on the approximate solution $\mathbf{w}^h + \mathbf{e}^h$.

A multigrid cycle defines the way to go from the fine grid to the coarsest grid (in the implementation, the coarsest grid is a 3 × 3 × 3 grid) and to return to the fine grid. This gives an approximation of z^h. One can iteratively apply the cycle to solve system 6. In the implementation, $\mathbf{M}^h\mathbf{z}^h = \mathbf{y}^h$ is approximately solved with one cycle. There are many possibilities to go back and forth from the fine grid to the coarsest grid. I chose the full multigrid V-cycle (Briggs et al., 2000).

NUMERICAL EXAMPLES

In these numerical examples, I use an eighth-order scheme. For a given frequency, the grid spacing is determined such that there are five points per minimum wavelength. To mimic an infinite medium, I added simple sponge layers. The sponge layers have 35 points in each direction, in which the ratio between the imaginary and real parts of the velocity varies from 0 to 0.5. Perfectly matched layers (PML) could have been used to reduce the number of points in the boundary layers (Berenger, 1994). The stopping criterion of the iterative method is the classic normalized-residual norm, $\|s - Au\|/\|s\|$. For the examples, I assume that convergence is reached when the criterion is smaller than 10^{-5}. The computations are carried out on an AMD Opteron 2.2 GHz processor with 16 GB of memory.

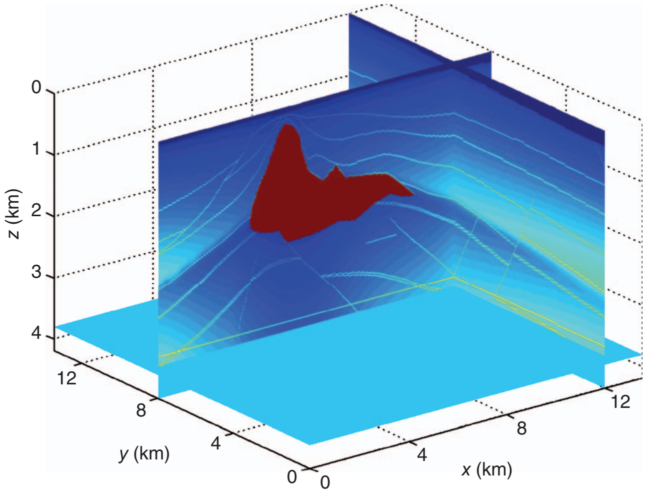

Figure 1. A section of the SEG/EAGE salt model.

3D SEG/EAGE salt-dome model

The 3D SEG/EAGE salt-dome model (Aminzadeh et al., 1997) is a model of 12 by 12 by 4.5 km. A section is plotted in Figure 1. The minimum velocity is 1500 m/s, and the maximum velocity is 4482 m/s. The 5-Hz pressure response to a source located in the top middle of the model is displayed in Figure 2. At 5 Hz , the core model has 225 × 225 × 70 points and a spacing of 60 m. The computational model has 295 × 295 × 140 points, namely 12.1 million grid points. The run required 2.2 GB and took 154 min. The 10-Hz pressure response is displayed in Figure 3. At 10 Hz, the core model has 450 × 450 × 140 points, and the computational model has 520 × 520 × 210 points, meaning 56.8 million grid points. The run required 10.6 GB and took 1111 min. The BICGSTAB convergence history is plotted for four different frequencies in Figure 4. The number of BICGSTAB iterations depends on frequency and roughly doubles when frequency doubles.

3D SEG/EAGE overthrust model

The second set of computations is carried out on the SEG/EAGE overthrust model (Aminzadeh et al., 1997). The model is 20 km long, 20 km wide, and 4 km deep; see Figure 5. The minimum velocity is 2179 m/s, and the maximum velocity is 6000 m/s. The 5-Hz pressure response to a source located in the middle top of the model is displayed in Figure 6. At 5 Hz, the core model has 236 × 236 × 56 points, which means a computational grid of 11.8 million grid points after having added the sponge layers. The run required 2.2 GB and took 126 min. The 10-Hz pressure response is displayed in Figure 7. At 10 Hz, the core model has 471 × 471 × 111

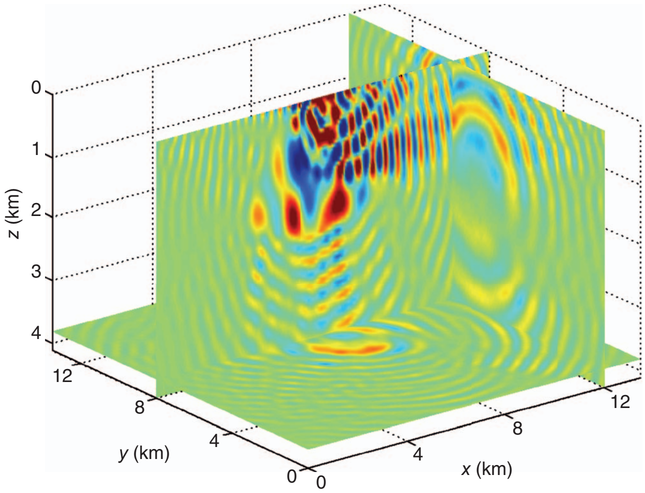

Figure 2. The 5-Hz pressure response on the SEG/EAGE salt model.

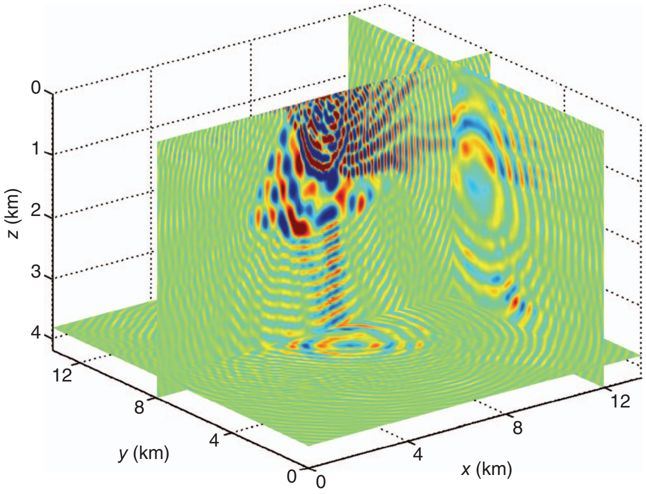

Figure 3. The 10-Hz pressure response on the SEG/EAGE salt model.

points, leading to a computational model of about 53.0 million grid points. The run required 9.9 GB and took 1380 min. The BICGSTAB convergence history is displayed in Figure 8 for four different frequencies. The number of BICGSTAB iterations depends again on frequency and roughly doubles when frequency doubles.

Convergence results

To analyze this iterative solver, the number of BICGSTAB iterations is plotted versus the average number of points per direction, n. Since the number of points per minimum wavelength is kept constant during the computation, this means that n is proportional to frequency f. The results are given for the 2D Marmousi example (Riyanti et al., 2006), the 3D SEG/EAGE salt model, and the 3D SEG/EAGE overthrust model. The largest 3D model that fits on 16 GB of RAM corresponds to the 12-Hz case and a model of about 85 million unknowns. Figure 9 numerically shows that the number of iterations increases roughly linearly with n. Because one multigrid cycle is in $O(n^3)$ in 3D, one BICGSTAB iteration requires $O(n^3)$ iterations. This numerical result indicates that the complexity of the iterative solver for the 3D Helmholtz equation is $O(n^4)$. This complexity is suboptimal because the optimal one is $O(n^3)$. The optimal complexity implies that the number of iterations is independent of the number of grid points. The possibility to improve this approach to reach optimal complexity is still an open question.

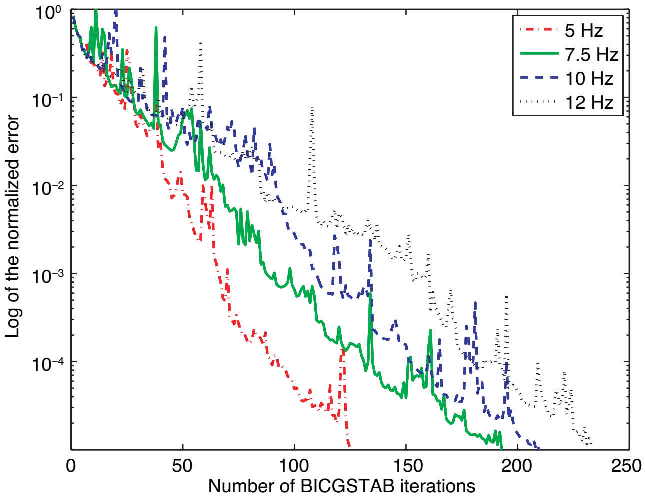

Figure 4. Convergence of the BICGSTAB for the 5-, 7.5-, 10-, and 12-Hz cases with the SEG/EAGE salt model.

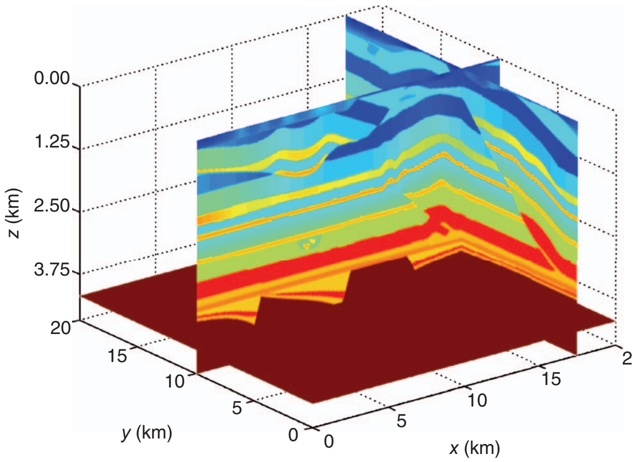

Figure 5. A section of the SEG/EAGE overthrust model.

APPLICATIONS TO SEISMIC IMAGING

The numerical results seem to indicate that this iterative solver is more efficient and robust than the ones based on the incomplete lower-upper (ILU) decomposition or separation of variables (Plessix and Mulder, 2003; Erlangga et al., 2004). However, is it good enough for geophysical applications?

The imaging problem can be formulated as a least-squares inverse problem (Tarantola, 1987). The computation of the gradient requires the solution of the adjoint of system 3. The adjoint system can be solved with the same modeling method. I consider two different imaging problems:

1) The migration of reflection seismic data aims to reposition in depth the events recorded in time assuming a smooth velocity field. It corresponds to the first step of the optimization, i.e., the gradient computation (Lailly, 1983). It requires processing a

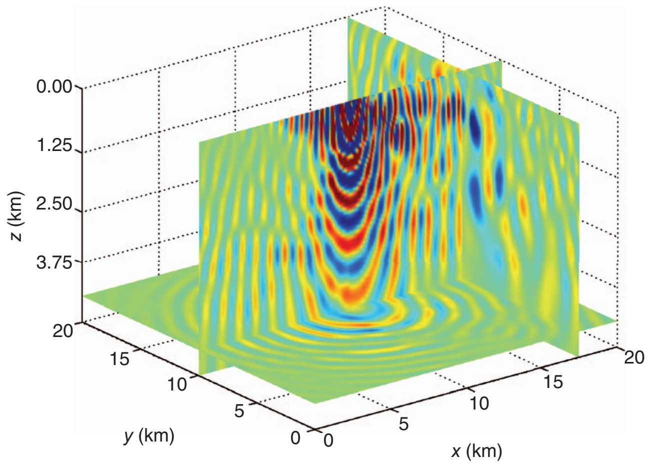

Figure 6. The 5-Hz pressure response on the SEG/EAGE overthrust model.

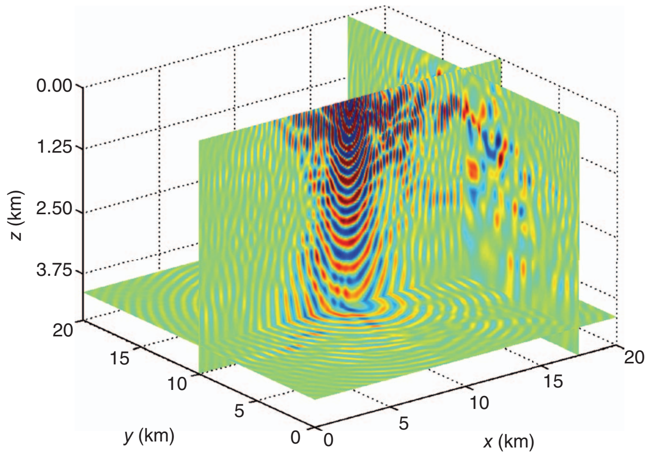

Figure 7. The 10-Hz pressure response on the SEG/EAGE overthrust model.

sufficiently large frequency band to obtain an impedance-contrast image with a good resolution. A large number of frequencies must be computed because the wavenumber redundancy is limited with reflection data.

2) The full-waveform tomography aims to recover the velocity field from long-offset data (Pratt, 1999; Operto et al., 2004). It does require an iterative approach and a reasonable initial guess. The minimization of the least-squares misfit functional with respect to the velocity may be done per frequency, or per small overlapping frequency band, from low to high frequencies. Thanks to the wavenumber redundancy with long-offset data, inverting a limited number of frequencies is generally sufficient. This scale separation is natural with a frequency-domain formulation. It is also feasible in the time domain after data filtering (Bunks et al., 1995).

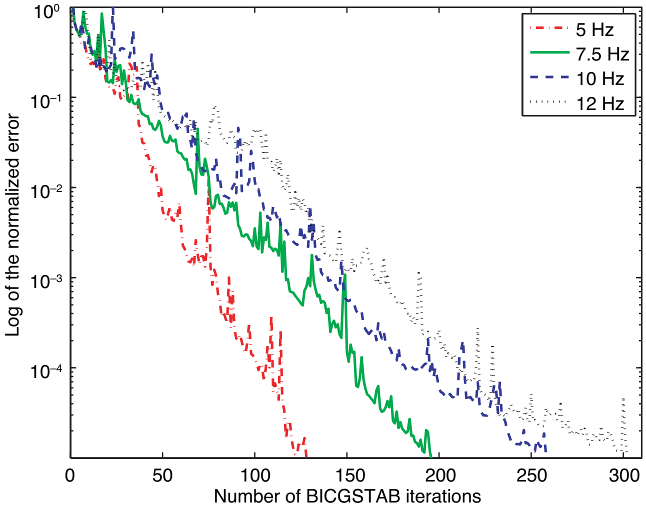

Figure 8. Convergence of the BICGSTAB for the 5-, 7.5-, 10-, and 12-Hz cases with the SEG/EAGE overthrust model.

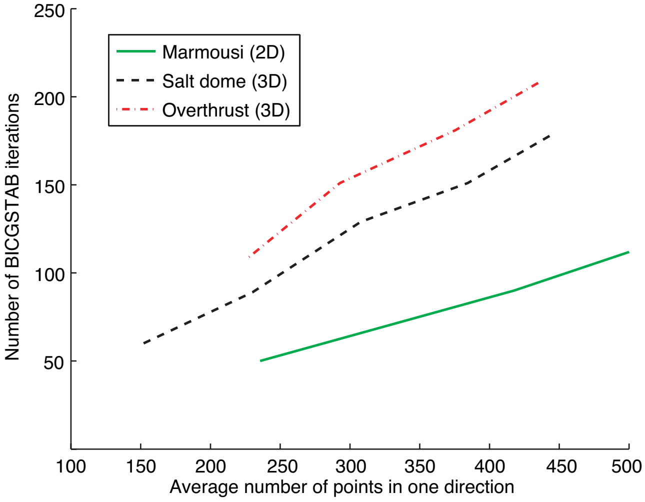

Figure 9. Convergence results: the number of BICGSTAB iterations versus the average number of discretization points in one direction. The solid green line corresponds to the 2D Marmousi model, the dashed black line to the 3D SEG/EAGE salt model, and the dotted-dashed red line to the 3D SEG/EAGE overthrust model.

One should estimate the approximate number of floating-point operations to determine in which domain the migration and the full-waveform tomography should be done.

Rough estimation of the number of operations of the time-domain approach

I consider an explicit time-domain scheme with an eighth-order scheme in space and a second-order scheme in time. The evaluation of a time step requires approximately $35n^3$ operations (the factor 35 corresponds mainly to the number of operations needed to evaluate the Laplacian operator; see equation 2). The total number of operations is roughly $35 n_t n^3$, where n is the number of grid points in one direction and n_t is the number of time steps. For stability reasons, the time sampling Δt and the spacing Δx satisfy $\Delta t = \alpha_1/\Delta x/v_{\max}$, with $v_{\max}$ the maximum velocity; α_1 is generally smaller than 1, depending on the particular implementation; here, we assume $\alpha_1 = 0.2$. By definition, the length of the model $L = n\Delta x$ and the recording time $T = n_t \Delta t$. In practice, the recording time is also linked to the length of the model, for instance, $T = \alpha_2/L/v_{\min}$, with $v_{\min}$ the minimum velocity and α_2 around two. This gives $n_t = (\alpha_2/\alpha_1)\ (v_{\max}/v_{\min})\ n$, with $v_{\max}/v_{\min} \simeq 3, n_t = 30n$. This crude calculation gives the total number of operations for the time-domain solver N_t^{op}:

$$N_t^{op} = \alpha n^4, \quad \text{with } \alpha \simeq 1000. \tag{11}$$

Rough estimation of the number of operations of the frequency-domain approach

Whereas, in the time-domain approach, the number of grid points is fixed for a given frequency band (and related to the highest frequency modeled during the computation), the number of grid points in the frequency domain can be adapted to the current frequency to save computation time. Let us call n the (average) number of discretization points in one direction associated with a given frequency. The cost of the frequency-domain approach equals the cost of one BICGSTAB iteration multiplied by the number of iterations. One BICGSTAB iteration mainly consists of 12 vector-vector products and two matrix-vector products. Since the calculus is carried out in the complex space, the 12 vector-vector operations represent approximately $75 n^3$ (scalar) operations. A matrix-vector product is constituted by one multigrid cycle to solve $\mathbf{Mz} = \mathbf{y}$ and one wave-operator evaluation $\mathbf{Az}$. The cost of the multigrid cycle is the cost on the fine grid multiplied by $\Sigma_{q=0}^{Q} 1/8^q \simeq \frac{8}{7} \simeq 1.15$, with Q the number of coarsening levels. On the fine grid and with the current implementation, the main operations are two Gauss-Seidel evaluations (corresponding to one post-smoother with a symmetric Gauss-Seidel), one wave-operator evaluation (related to the computation of the residual), two restrictions (one for the pressure field and one for the velocity), one prolongation (for the pressure field), and one vector-vector operation. A Gauss-Seidel evaluation is more or less similar to the wave-operator evaluation. With a an eigth-order scheme, this means approximately $70n^3$ (because one works with complex numbers). A restriction or a prolongation takes approximately $50n^3$ operations. After adding all of these operations, the number of floating-point operations of one BICGSTAB iteration is $\gamma_1 n^3$, with $\gamma_1 \simeq 1200$. From the numerical results and the convergence plot, Figure 9, $n_{it} = \gamma_2 n$, with $\gamma_2 \simeq 0.5$ and n_{it} the number of BICGSTAB iterations. This calculation gives the total number of operations for the iterative solver, N_f^{op}:

$$N_f^{op} = \gamma n^4, \quad \text{with } \gamma \simeq 600. \tag{12}$$

Gradient computation

The number of discretization points depends on the current frequency in the frequency domain and on the maximum frequency in the time domain. We can write a formal system satisfied by the synthetics $c(n(f),v)$:

$$B(n(f),v)c(n(f),v) = s, \tag{13}$$

where B is the wave operator and f is the current frequency in the frequency domain or the maximum frequency in the time domain. The least-squares misfit function is simply, with d the data,

$$J(v) = \frac{1}{2}\|c(n(f),v) - d\|^2, \tag{14}$$

and the gradient (Plessix, 2006) is

$$\frac{\partial J(v)}{\partial v} = \left\langle \lambda(n(f),v), \frac{\partial B(n(f),v)}{\partial v} c(n(f),v) \right\rangle, \tag{15}$$

with

$$B(n(f),v)^* \lambda(n(f),v) = (c - d), \tag{16}$$

where $\langle , \rangle$ is the scalar product (in the time domain, this would correspond to the crosscorrelation); $B(n(f),v)^*$ is the reverse time operator in the time domain and the adjoint operator in the frequency domain. System 16 is solved like system 13. If one ignores the memory or input/output (I/O) requirements for the moment, the computational cost of the gradient is roughly equal to twice the computational cost of the wave operator.

Migration

As mentioned before, the migration requires the processing of a large enough frequency band. Let us call $f_{\max}$ the maximum frequency. In the time domain, the least-squares misfit functional is

$$\frac{1}{2}\|c(n(f_{\max}),v) - d\|^2. \tag{17}$$

Therefore, the number of operations for the time-domain migration is roughly

$$N_{t,\text{mig}}^{op} \simeq 2N_t^{op} = 2\alpha n^4(f_{\max}). \tag{18}$$

In the frequency-domain, the least-squares misfit functional is

$$\sum_{j=1}^{n_f} \frac{1}{2}\|c(n(f_j),v) - d\|^2, \tag{19}$$

with n_f the number of frequencies and $f_j \in [0, f_{\max}]$. When the number of grid points per wavelength is kept constant, $n(f) = f_j/f_{\max} n(f_{\max})$. For migration, the full frequency band is needed to obtain a good resolution; $f_j = j\Delta f$, with Δf the frequency spacing and $n_f = f_{\max}/\Delta_f$. Using equation 12 and the fact that $\Sigma_{j=1}^{n_f} j^4 \simeq n^5/5$, the number of operations for the frequency-domain approach is $\frac{\gamma}{5} n_f n^4(f_{\max})$. The Nyquist theorem tells us that $\Delta f = 1/T$, with T the recording time (this criterium can be slightly relaxed if we consider the maximum offset [Mulder and Plessix, (2004)]. As mentioned previously, $T = \alpha_2 n(f_{\max})\Delta x/v_{\min}$. A discretization of five points per minimum wavelength gives $5\Delta x = v_{\min}/n_f \Delta f$. Therefore, $n_f = \alpha_2/5n(f_{\max})$. The total number of operations for the migration with the iterative Helmholtz solver is then roughly

$$N_{f,\text{mig}}^{op} = 2\beta n^5(f_{\max}) \quad \text{with } \beta = \frac{\gamma\alpha_2}{25} \simeq 48. \tag{20}$$

This shows that the number of operations for the frequency-domain approach divided by the number of operations for the time-domain approach is roughly $0.05n$. This estimation indicates that the time-domain migration is one order of magnitude faster for medium-sized problems and will be even more advantageous for larger problems.

Full-waveform tomography

The full-waveform tomography requires the minimization of a least-squares misfit functional. In the frequency domain, the minimization is carried out per frequency (or per narrow frequency band). In the time domain, the minimization is carried out per low-pass filtered data. Given a frequency set (f_j), the full-waveform tomography consists of a sequence of minimization problems:

$$\tilde{v}(f_j) = \text{argmin}_v \frac{1}{2}\|c(n(f_j),v) - d\|^2 \tag{21}$$

with $\tilde{v}(f_{j-1})$ as the starting model.

The number of operations to compute $c(n(f_j),v)$ is given by equation 11 for the time domain and by equation 12 for the frequency domain. The two approaches have the same complexity. The number of operations of the frequency-domain approach is roughly one half that of the time-domain approach. Since the count of the floating-point operations is crude, this result indicates that the time-domain and the frequency-domain approaches require approximately the same number of floating-point operations.

If one considers a survey of n_s shots, the gradient computation of the least-squares misfit function requires $C^i n_s n^4$ operations for the frequency-domain iterative method and $C^t n_s n^4$ for the time-domain approach ($C^t \simeq \mathrm{C}^i$). The computation of the gradient with a direct solver requires $C_1^d n^6 + C_2^d n_s n^4$ (Ashcraft and Liu, 1998) (n^6 corresponds to the **LU** factorization of the matrix and n^4 to the matrix-vector products once the matrix is factorized, assuming a compact scheme on the 27-point stencil (Operto et al., 2007). If $n_s \ll n^2$, this shows that the time-domain or the iterative approaches are the methods of choice. If the survey is fully 3D, $n_s = C_s n^2$ and all of the approaches have the same complexity. For large problems, one should use a parallelized implementation. The iterative method can a priori be parallelized using domain-decomposition techniques. The time-domain approach is also parallelized using domain-decomposition techniques, for instance. The direct solver has been parallelized (Amestoy et al., 2000). The method of choice now depends on the constants C^t,C^f, C_1^d, C_2^d, and C_s. Except C_s, these constants depend on the efficiency of the parallelization and on the computer-cluster architecture. An estimation of these constants is required to decide on the method of choice from a computation-time point of view.

The memory requirement of these approaches is quite different. The frequency-domain iterative method needs approximately 12 arrays of n^3 elements. Its memory requirement is then $D^i n^3$ bytes. The frequency-domain direct solver needs $D^d n^4$ bytes. A classic time-domain implementation needs $D^t n^4$ bytes for the gradient computation. However, with optimal checkpointing, the memory requirement decreases to $O(n^3)log(n)$, while the computation time is only multiplied by $O(log(n))$ (Griewank, 1992). Symes (2007) showed the relevance of this approach for reverse time migration. With n between 200 and 1000, $log(n)$ is approximately 6 and can be included in the constants. Thus, the memory requirement for the time domain becomes $D_2^t n^3$ bytes. D_2^t is a priori larger than D^i, but of the same order. From a memory point of view, the time-domain or the frequency-domain iterative approaches are much cheaper compared to the frequency-domain direct solver.

CONCLUSIONS

A 3D iterative solver for the wave equation with a preconditioned BICGSTAB method has been studied in a geophysical context. The preconditioner consists of a multigrid cycle applied to a heavily damped wave equation. This leads to a robust iterative solver for the undamped wave equation at seismic frequencies. A numerical study based on 3D examples shows that the number of BICGSTAB iterations increases roughly linearly with frequency. This confirms the results obtained on 2D examples.

Based on the numerical results, the relevance for two imaging problems has been studied. The migration is more efficiently solved in time domain than in frequency domain. A rough count of the number of floating-point operations indicates that the time-domain approach is at least one order of magnitude faster for realistic size problems. The situation is different for the full-waveform tomography based on a least-squares formulation and a scale separation. For this problem, the two approaches have the same complexity and roughly the same number of floating-point operations. With a full 3D problem, when the number of sources is proportional to the square of the number of grid points in one direction, the time-domain and iterative frequency-domain methods have the same complexity as the frequency-domain method based on a direct solver. However, their memory requirement is much smaller compared to that of the direct solver.

The imaging problem requires the processing of multishot data. It may still be possible to speed up the frequency-domain iterative approach. For instance, solutions already computed may serve to define a good initial guess. This remains an open question. A priori, one can parallelize the iterative solver with a domain-decomposition technique. The efficiency of this parallelization is also an open question.

ACKNOWLEDGMENT

I would like to thank the reviewers for their comments and suggestions.

APPENDIX A

MULTIGRID ON A 1D WAVE EQUATION

In this appendix, I directly apply the multigrid method on the wave equation. With a simple 1D homogeneous model, I illustrate that the multigrid solver for the wave equation only works with a large viscous term, but does not work in a pure acoustic case. This is why one cannot directly apply the multigrid method to solve equation 3 and a preconditioned Krylov-based iterative method is proposed. This behavior has been discussed in the literature, for instance, in Elman et al. (2001).

Let us consider the 1D wave equation

$$-\frac{\partial^2 u(x)}{\partial x^2} - k^2(\beta_r + i\beta_i)^2 u(x) = s(x), \qquad \text{(A-1)}$$

with Dirichlet boundary conditions, namely, $u = 0$ at the boundary of the numerical domain. A second order scheme gives

$$-\frac{u_{j+1} - 2u_i + u_{j-1}}{h^2} - k_j^2(\beta_r + i\beta_i)^2 u_s = s_j, \qquad \text{(A-2)}$$

with $j \in [1, n-1]$, $u(0) = 0$ and $u(n) = 0$, and $n + 1$ is the number of discretization points. This leads to a linear system with a triangular matrix that can be solved with an **LU** decomposition. The solution obtained with this direct solver is called “exact.” For the numerical examples, I consider a velocity of 2000 m/s, a frequency of 10 Hz, i.e., a wavenumber equal to $\pi/100$, and a discretization of 5 m and 257 points. I apply one multigrid V-cycle and a direct solver on the coarsest grid. When the discretization spacing on the coarse grid is 2^q, this corresponds to q coarsening levels. I look at the cases $q = 1,2,3,4,$ and 5. The results after one multigrid V-cycle are shown in Figure A-1 for the pure acoustic case ($\beta_r = 1, \beta_i = 0$), in Figure A-2 for a viscoacoustic case with a mild damping ($\beta_r = 1$, $\beta_i = 0.1$), and in Figure A-3 for a heavily damped viscoacoustic case ($\beta_r = 1$, $\beta_i = 1$). With the pure acoustic case or the mildly damped viscoacoustic case, the results obtained with one multigrid V-cycle do not approximate the true solution when more than three coarsening levels are applied. In fact, as noted by many authors, for example, Elman et al. (2001), because of the indefiniteness of the linear system, the solution on the coarse grid does not approximate the smooth part of the true solution. The situation is different with the heavily damped viscoacoustic case, because now the equation behaves like a diffusive equation, namely, the response is attenuated over approximately a wavelength. One multigrid V-cycle does ap-

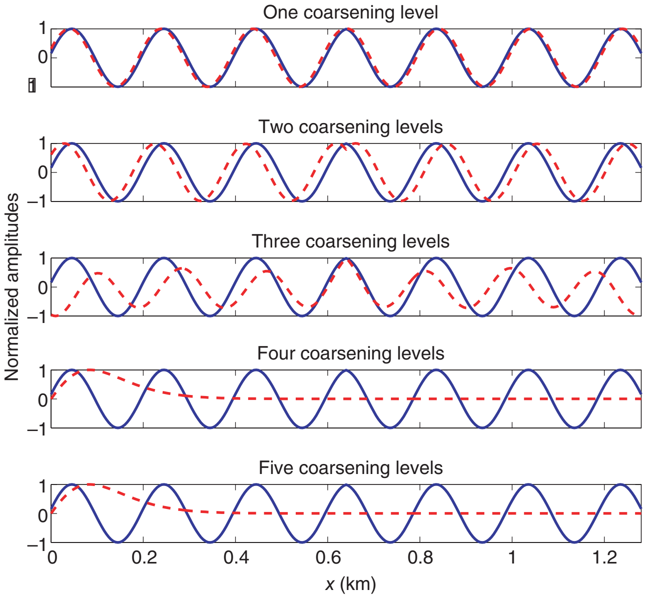

Figure A-1. Solutions in the pure acoustic case ($\beta_r = 1, \beta_i = 0$). The blue line corresponds to the exact solution (obtained with a direct solver), and the dotted red lines correspond to the solutions obtained after one multigrid V-cycle when one to five coarsening levels are used.

proximate the true solution. The multigrid method is not the only method that would successfully solve the damped system. One uses it because it has the optimal complexity.

I now use the heavily damped viscoacoustic approximated solution as a preconditioner to solve iteratively the undamped acoustic case as proposed in Erlangga et al. (2004, 2006). The BICGSTAB iterative solver without preconditioner converges in 440 iterations to the undamped 1D acoustic solution. If the BICGSTAB method is preconditioned by a mildly damped viscoacoustic wave equation, the solver converges in seven iterations to the undamped 1D acoustic solution. Unfortunately, the mildly damped viscoacoustic equation cannot be easily solved. If the BICGSTAB method is preconditioned by a heavily damped viscoacoustic wave equation, the solver converges in 19 iterations to the undamped 1D acoustic solution, and one can approximately solve the heavily damped viscoacoustic equation with one multigrid cycle.

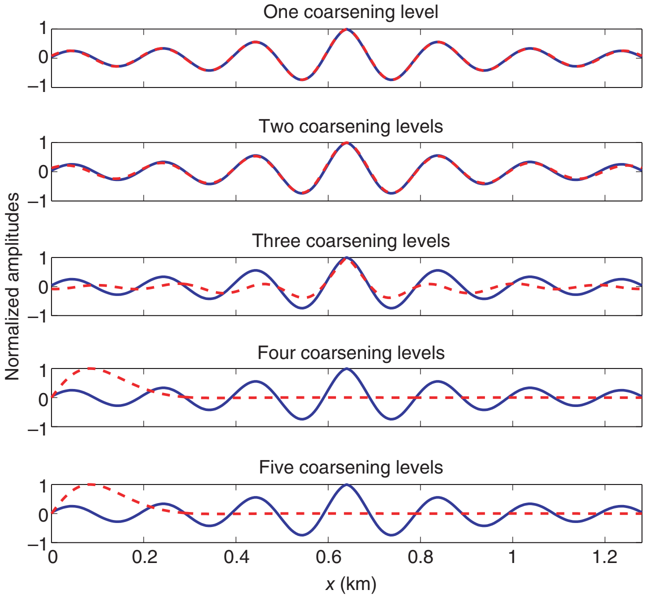

Figure A-2. Solutions in the mildly damped viscoacoustic case ($\beta_r = 1, \beta_i = 0.1$). The blue line corresponds to the exact solution (obtained with a direct solver), and the dotted red lines correspond to the solutions obtained after one multigrid V-cycle when one to five coarsening levels are used.

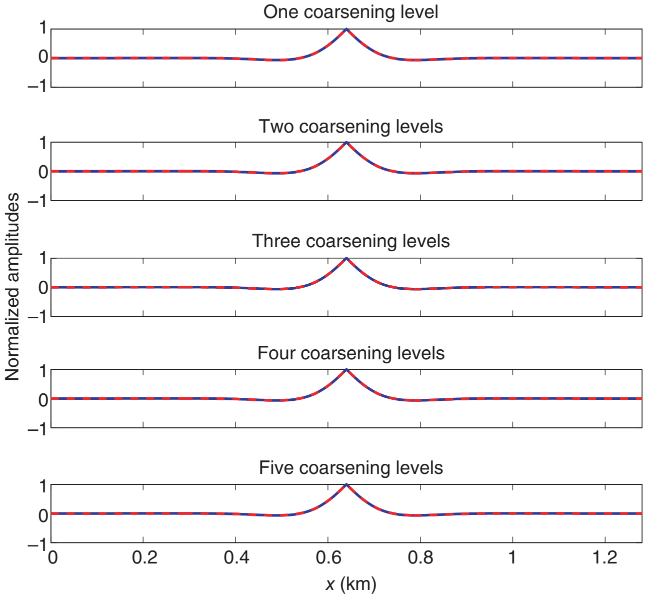

Figure A-3. Solutions in the heavily damped viscoacoustic case ($\beta_r = 1, \beta_i = 1$). The blue line corresponds to the exact solution (obtained with a direct solver), and the dotted red lines correspond to the solutions obtained after one multigrid V-cycle when one to five coarsening levels are used.

REFERENCES

Amestoy, P. R., I. S. Duff, and J.-Y. L'Excellent, 2000, Multifrontal parallel distributed symmetric and unsymmetric solvers: Computer Methods in Applied Mechanics and Engineering, **184**, 501–520.

Aminzadeh, F., J. Brac, and T. Kunz, 1997, 3-D salt and overthrust models: SEG/EAGE 3-D Modeling Series no.1, SEG.

Ashcraft, C., and J. W. H. Liu, 1998, Robust ordering of sparse matrices using multisection: SIAM Journal on Matrix Analysis and Applications, **19**, 816–832.

Berenger, J.-P., 1994, A perfectly matched layer for absorption of electromagnetic wave: Journal of Computational Physics, **114**, 185–200.

Briggs, W. L., V. E. Henson, and S. F. McCormick, 2000, A multigrid tutorial, 2nd ed.: SIAM.

Bunks, C., F. M. Saleck, S. Zaleski, and G. Chavent, 1995, Multiscale seismic waveform inversion: Geophysics, **60**, 1457–1473.

Chen, K., 2005, Matrix preconditioning techniques and applications: Cambridge University Press.

Elman, H., O. Ernst, and D. O'Leary, 2001, A multigrid method enhanced by Krylov subspace iteration for discrete Helmholtz equation: SIAM Journal on Scientific Computing, **23**, 1291–1315.

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: 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.

Griewank, A., 1992, Achieving logarithmic growth of temporal and spatial complexity in reverse automatic differentiation: Optimization Methods and Software, **1**, 35–54.

Jo, C.-H., C. Shin, and J. H. Suh, 1996, An optimal 9-point, finite-difference, frequency-space, 2-D scalar wave extrapolator: Geophysics, **61**, 529–537.

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.

Larson, E., 1999, A domain decomposition method for the Helmholtz equation in a multilayer domain: SIAM Journal on Scientific Computing, **20**, 1713–1731.

Lailly, P, 1983, The seismic inverse problem as a sequence of before stack migrations: SIAM Conference on Inverse scattering: Theory and Applications, 206–220.

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

Mulder, W. A., and R.-E. Plessix, 2004, How to choose a subset of frequencies in frequency-domain finite-difference migration: Geophysical Journal International, **158**, 801–812.

Operto, S., C. Ravaut, L. Improta, J. Virieux, A. Herrero, and P. Dell'Aversana, 2004, Quantitative imaging of complex structures from dense wide-aperture seismic data by multiscale traveltime and waveform tomography: A case study: Geophysical Prospecting, **52**, 625–651.

Operto, S., J. Virieux, P. Amestoy, J. Y. L'Excellent, and L. Giraud, 2007, 3D finite-difference frequency-domain modeling of viscoacoustic wave propagation using a massively parallel direct solver: A feasibility study: Geophysics, this issue.

Plessix, R.-E., 2006, A review of the adjoint-state method for computing the gradient of a functional with geophysical applications: Geophysical Journal International, **167**, 495–503.

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.

Pratt, R. G., 1999, Seismic waveform inversion in the frequency domain, Part I: Theory and verification in a physical scale model: Geophysics, **64**, 888–901.

Riyanti, C. D., Y. A. Erlangga, R.-E. Plessix, W. A. Mulder, C. Vuik, and C. W. Oosterlee, 2006, A new iterative solver for the time-harmonic wave equation: Geophysics, **71**, no. 5, E57–E63.

Symes, W. W., 2007, Reverse time migration with optimal checkpointing: Geophysics, this issue.

Tarantola, A., 1987, Inverse problem theory: Elsevier Science Publ. Co, Inc.

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: 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.