Feasibility of solving least squares prestack Kirchhoff migration using multigrid methods

Abdolnaser Yousefzadeh and John C. Bancroft

ABSTRACT

The multigrid technique is a powerful method for solving a linear matrix equation; for finding the low frequency components of the solution as rapidly as the high frequency components.

We investigate the feasibility of different approaches for using the multigrid method in solving the linear system of Kirchhoff Least-Squares Prestack Time Migration equation.

To date, this study shows that the standard multigrid method is not able to solve the Kirchhoff Least-Squares Prestack Time Migration equation for at least two reasons. First, the kernel of the main problem, $\boldsymbol{G'G}$, is not a diagonally dominant matrix, therefore, Jacobi or Gauss-Seidel iteration, the standard iterative algorithms used in multigrid methods, are not effective. Second, matrices are too large and dense to be loaded in the memory of today's computers.

However, the performance of the Conjugate Gradient and Kaczmarz methods as multigrid solvers is examined for some synthetic data sets, including Marmousi, and results are shown. Convergence rate of the Conjugate Gradient is independent of the frequency content of the solution. Therefore, it does not converge more quickly with high frequency contents of the solution. Since Conjugate Gradient does not smoothe, it should not be considered for the multigrid iterative solver.

Using the Conjugate Gradient as an iterative solver for the multigrid method may slightly reduce the number of iterations for the same rate of convergence in the Conjugate Gradient itself. However, it does not reduce the total computational cost.

1. INTRODUCTION

1.1 Kirchhoff Migration

Migration is an effective tool in seismic data processing in order to move dipping reflection events to their true geological locations. It also collapses diffracted energy back to the location of the scatterpoints.

Hagedoorn (1954) introduced the first computational method of migration, diffraction summation, using ruler and compass. Kirchhoff migration was later developed, based on this method. In the 70's Claerbout and Doherty (1972) showed how migration is an approximate solution to the wave equation. The integral formulation of the Kirchhoff migration was introduced by Schneider (1978). Gazdag (1978) and Stolt (1978) showed how to perform migration using properties of the Fourier Transform (Gary et. al., 2000).

Each point in the geological domain is considered a diffraction point. Kirchhoff wave equation migration (Schneider, 1978) collapses all diffracted energies back to the original

scatterpoint. In the resulted migration image, each dipping reflector is moved to the correct location with true dip and length.

With a simple geology structure and in the case of regularly and completely sampled data, Kirchhoff migration produces a good estimate of the underground reflectivity. However, Kirchhoff migration may not be able to reveal complex subsurface geology with strong lateral velocity variations.

In a real data acquisition, especially in 3D surveys, it is difficult to have a dense and spatially regular sampled seismic data set. Kirchhoff migration is able to handle incomplete or irregular seismic data. However, incomplete data produce migration artifacts and may give a blurred image of the earth subsurface reflectivity.

In order to overcome the mentioned problem, Kirchhoff migration can be augmented by a generalized inverse as an approximation to the exact inverse as proposed by Tarantola (1984). This approach is called Least-Squares migration or inversion (Nemeth et. al., 1999, Duquet et. al., 2000, Kuehl and Sacchi, 2001). This study focuses on the Prestack Kirchhoff Time Migration, therefore, Least-Squares Migration (LSM) is referred to as Kirchhoff Least-Squares Prestack Time Migration (LSPSTM). LSPSTM solves a system of large linear equations.

A modified version of the Conjugate Gradient (CG) method, called Least Squares Conjugate Gradient (LSCG) (Scales, 1987) has been widely used as a solver for the LSM equation (Nemeth et. al., 1999, Duquet et. al., 2000, Kuehl and Sacchi, 2001, and Yousefzadeh, 2008).

In the LSCG algorithm, migration and modelling are operators. Therefore, the matrix form of the migration and de-migration operators are avoided. In this paper, the feasibility and disadvantages of using the matrix form of the Kirchhoff PSTM and also different numerical solvers for the reflectivity inversion problem are investigated.

The same study could be done for Kirchhoff prestack depth migration. However, in order to compare different solvers of prestack Kirchhoff migration the simpler and faster algorithm of time migration is chosen. The same results can be achieved using Kirchhoff depth migration.

1.2. Kirchhoff Least-Squares Prestack Time Migration (LSPSTM) Equation

Convolution and Kirchhoff seismic modelling with diffraction are examples in exploration seismology which can be formulated as algebraic linear problems.

These problems can be written in the general form of

$$\boldsymbol{d} = \boldsymbol{G}\boldsymbol{m}. \tag{1}$$

where, in case of seismic modelling, $\boldsymbol{d}$ is the observed seismic data, $\boldsymbol{m}$ is the earth reflectivity model, and $\boldsymbol{G}$ is an operator acting on $\boldsymbol{m}$ in order to produce $\boldsymbol{d}$.

The inversion process,

$$\boldsymbol{m} = \boldsymbol{G}^{-1}\boldsymbol{d}, \tag{2}$$

recover the earth model or reflectivity from the seismic data. Inverting the $\boldsymbol{G}$ matrix may be extremely difficult. Thus, approximations to the inversion are used. The first approximation uses the transpose of $\boldsymbol{G}$:

$$\widehat{\boldsymbol{m}} = \boldsymbol{G}'\boldsymbol{d}. \tag{3}$$

which is the same as Kirchhoff migration that sums energy over diffraction. In equation 3 $\widehat{\boldsymbol{m}}$ is the migrated image and $\boldsymbol{G}'$ is the migration operator.

By defining Kirchhoff modelling as the forward process and Kirchhoff migration as its adjoint (transpose) operator, seismic imaging becomes an inversion problem. Substitution of $\boldsymbol{d}$ from equation 1 into equation 3 gives:

$$\widehat{\boldsymbol{m}} = \boldsymbol{G}'\boldsymbol{G}\boldsymbol{m}. \tag{4}$$

If the Hessian matrix ($\boldsymbol{G}'\boldsymbol{G}$) was equal to the unity matrix ($\boldsymbol{G}'\boldsymbol{G} = \boldsymbol{I}$) then Kirchhoff migration would be able to reconstruct the true model of earth subsurface reflectivity. However, due to geometrical spreading loss during modelling or irregularity, incompleteness of the sampled seismic data, and existence of noise, Hessian matrix, $\boldsymbol{G}'\boldsymbol{G}$ is different from identity matrix, $\boldsymbol{I}$ (Nemeth et. al., 1999). In such cases Kirchhoff prestack migration produces some artifacts in the migrated image. These migration artifacts can be attenuated by minimizing the difference between the observed data, $\boldsymbol{d}$, and modeled data, $\boldsymbol{G}\boldsymbol{m}$, expressed by $|\boldsymbol{G}\boldsymbol{m} - \boldsymbol{d}|$. Since data include some errors, trying to find a model to fit the data perfectly is not recommended. Therefore, the exact fitting will be replaced by

$$\boldsymbol{e} = \boldsymbol{G}\boldsymbol{m} - \boldsymbol{d}, \tag{5}$$

where $\boldsymbol{e}$ is the error vector (Sacchi, 2005). Minimum norm solution includes finding a model, $\boldsymbol{m}$, that minimizes the following cost function:

$$J(\boldsymbol{m}) = \|\boldsymbol{m}'\boldsymbol{m}\|^2, \tag{6}$$

subject to data constraint:

$$\|\boldsymbol{G}\boldsymbol{m} - \boldsymbol{d}\|^2 = \epsilon. \tag{7}$$

These two together implies the minimization of a cost function in the form of:

$$J(\boldsymbol{m}) = \|\boldsymbol{G}\boldsymbol{m} - \boldsymbol{d}\|^2 + \mu^2\|\boldsymbol{m}\|^2. \tag{8}$$

A general cost (objective) function to reduce migration artifacts can be written in the form of (Nemeth, 1999):

$$J(\boldsymbol{m}) = \|\boldsymbol{G}\boldsymbol{m} - \boldsymbol{d}\|^2 + \mu^2\mathcal{R}(\boldsymbol{m}). \tag{9}$$

where $\boldsymbol{d}$ is the observed data, which may be spatially incomplete or irregularly recorded. The first term in the right hand side of equation 9 is called data misfit. In the minimization of the cost function, $J(\boldsymbol{m})$, this term recovers a model to fit the data. The second term on the right hand-side of equation 9 is a regularization term, and μ is a

regularization weight. $\mathcal{R}(\boldsymbol{m})$ is a linear operator acting on $\boldsymbol{m}$ and is different for each purpose and uses some a priori information about model.

After ignoring the regularization term, $\mu = 0$, by taking first derivative of the remaining cost function and setting it equal to zero, the following normal equation is achieved:

$$\boldsymbol{G'Gm_{LS} - G'd = 0}.$$

This gives the least squares solution, $\boldsymbol{m_{LS}}$,

$$\boldsymbol{m_{LS} = (G'\,G)^{-1}G'd}. \tag{10}$$

The minimum norm or Euclidian norm is the simplest form of the regularization function, $\mathcal{R}(\boldsymbol{m})$ ($\mathcal{R}(\boldsymbol{m}) = \|\boldsymbol{m}\|_2^2$), which leads to the Damped Least-Squares solution, $\boldsymbol{m_{DLS}}$, to the problem:

$$\boldsymbol{m_{DLS} = (G'G + \mu^2 I)^{-1}G'd}. \tag{11}$$

Smoothing in the offset direction (smoothing in the direction of the hyperbola trajectories in a shot gather) is another constraint which can be performed on the LSPSTM by considering $\mathcal{R}(\boldsymbol{m}) = \|\boldsymbol{D}_h\boldsymbol{m}\|_2^2$, where $\boldsymbol{D}_h$ is the first derivative in the offset direction and mathematically is expressed by multiplication of the following matrix with a vector:

$$\boldsymbol{D}_h = \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & -1 \end{bmatrix}. \tag{12}$$

The first derivative operator acts as a high pass filter. Therefore, mimimization of a cost function with this regularization term is equal to penalizing high frequency contents. Solution to a LSPSTM problem with smoothing as the regularization, $\mathbf{m_{SLS}}$, is:

$$\boldsymbol{m_{SLS} = (G'G + \mu^2 D_h' D_h)^{-1}G'd}. \tag{13}$$

Generally, images resulted from LSM have higher resolution than from migration. These high resolution images can then be used in a forward problem in order to reproduce or interpolate the missing traces (Nemeth, 1999).

Figure 1 compares the resolution of the Kirchhoff migration and Kirchhoff LSPSTM for a simple synthetic model. Figure 1a is the synthetic reflectivity model including some horizontal and dipping folded and faulted layers. This model is 1 km long and 1.2 sec in depth (time). Background velocity starts at 1300 m/s at the top and increases linearly to 2300m/s at the bottom. Using the forward Kirchhoff prestack time modelling operator with five sources (250m interval in between) and 60 receivers per source (with 16.67m interval), synthetic data are generated. Then, 5% random noise is added to the data. The synthetic data is prestack migrated (Figure 1b) and LSPSTM-ed with 20 iterations in the CG (Figure 1c). The LSPSTM produced higher resolution image than the migration itself.

There are two issues associated with replacing migration with LSM. The main problem is that the convergence of the method to the correct solution strongly depends on

the accuracy of the background velocity information. LSM is more sensitive to the accuracy of the velocity information than migration itself (Yousefzadeh, 2008).

In addition to the dependency of the method on the accurate velocity information, LSPSTM requires more computer time and memory than migration. As an example, in solving the equation with LSCG method, each iteration in the CG requires two migration/modelling passes. Migration is one of the most time consuming and expensive procedures in the seismic data processing.

Each Kirchhoff prestack time migration includes calculation and/or application of the Double Square Root (DSR) equation, proper weight function, rho filter, antialiasing filter, cross-correlation, and interpolation. In order to have a modelling operator which is exactly the adjoint of the migration operator, adjoint of all these steps must be considered in the modelling, as well. This procedure causes the modelling operator to be even more expensive than the migration operator.

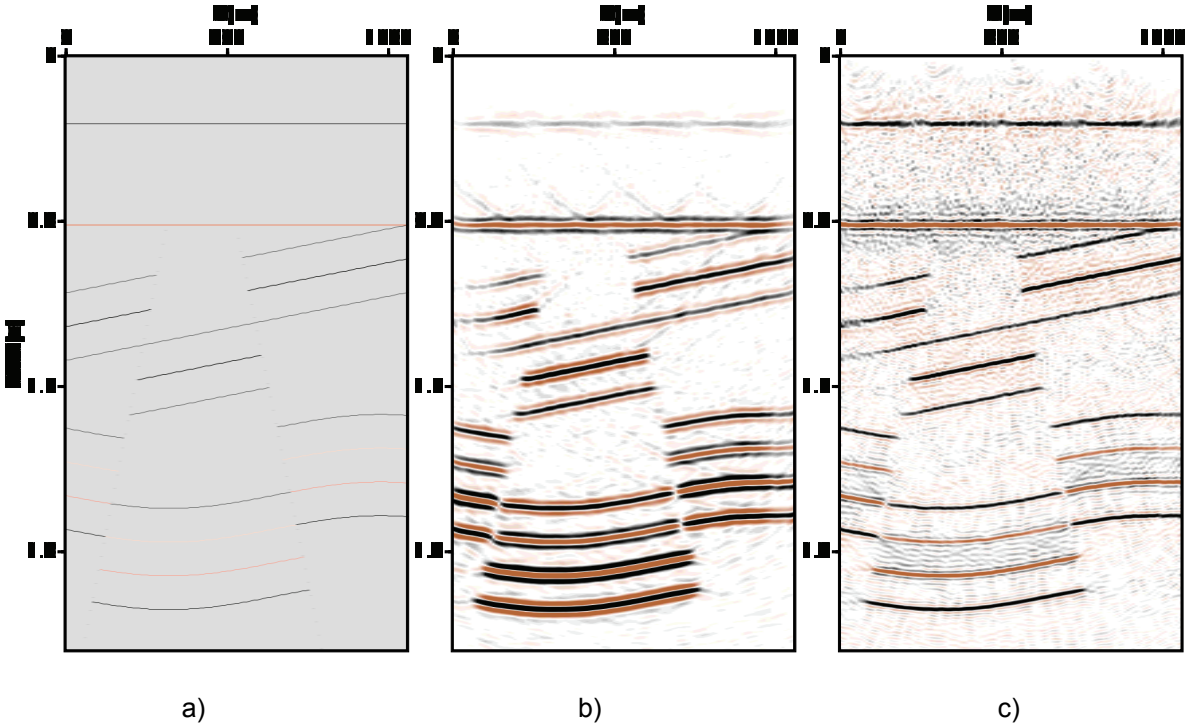


FIG. 1. a) Synthetic reflectivity model, 1.2 km length and 2 seconds depth. Background velocity linearly increases from 1300m/s at the top to 2300m/s at the bottom. b) Kirchhoff PSTM of the synthetic data from model in a). c) Kirchhoff LSPSTM after 20 iterations in the LSCG. Image of LSPSTM has higher resolution than standard migration.

1.3. Solving the LSPSTM Equation

The CG method is the classical solver in seismic inversion problems. In this study, the feasibility of using multigrid properties in solving LSPSTM in order to reduce the computational cost or enhance the resolution of the resulting image is investigated.

To understand the multigrid methods, the method of Jacobi iterations and its properties are explained in detail in the next section. Then the feasibility of using standard multigrid

methods for solving LSPSTM is investigated. It is shown why standard multigrid methods are not applicable to the mentioned problem.

In the last next section, the results of using the multigrid CG method on the mentioned problem is shown.

2. MULTIGRID METHODS

The performance of least squares seismic inversion usually requires solving a large system of linear equations in the general form of:

$$\boldsymbol{A}\boldsymbol{u} = \boldsymbol{b}, \tag{14}$$

where $\boldsymbol{A}$ is a general $M \times N$ matrix, for example Hessian, $\boldsymbol{G}'\boldsymbol{G}$, in equation 10 or $\boldsymbol{G}'\boldsymbol{G} + \mu^2\boldsymbol{I}$ in equation 11, $\boldsymbol{b}$ is a vector with M elements and $\boldsymbol{u}$ is the unknown, exact solution (model) vector.

In linear algebra, methods to solve equation 14 are placed in the one of these two main groups (Strang, 1986): direct methods such as Gaussian elimination, which find the exact solution. Iterative methods, on the other hand, try to improve an initial estimation of $\boldsymbol{u}$ to obtain a reasonably good approximation to the exact solution. In the stationary iteration methods, for example Jacobi, convergence steps are applied after each iteration while in the gradient methods, for example CG and Krylove, changes are applied within the iteration to find the solution faster (Strang, 1986).

In the LSPSTM equation Hessian matrix, $\boldsymbol{G}'\boldsymbol{G}$, is a large matrix which may never be solved using direct methods. The size of $\boldsymbol{G}$ equals the number of grids in the model multiplied by the number of observations (data). In such cases direct methods are not efficient.

2.1 Multigrid Solvers

Multigrid methods use one special property of the Jacobi and Gauss-Seidel methods in order to converge to the solution more rapidly, with better recovery of the low frequency components of the solution.

The ability to use multigrid methods to solve many types of PDEs more rapidly than other iterative methods is proven (Briggs et. al., 2000). In this study, we attempted to use the multigrid method to solve Kirchhoff LSPSTM in order to reduce the computational costs or enhance the resolution of the resulted image.

Using multigrid methods for solving seismic problems is not a new idea. Bunks et. al. (1995) used the multigrid method to perform seismic waveform velocity inversion on the Marmousi data set. Their idea was eliminating the local minima of the objective function by solving the problem on a coarser grid in order to guarantee convergence to the global minimum and avoid local minima, which are closer to the starting point.

The multigrid method is used to enhance the resolution of the seismic data during deconvolution (Millar and Bancroft, 2004). They showed better recovery of reflectivity and low frequency damping than the Gauss-Seidel method. However, they claimed that

successfulness of the method depends on the good estimation of wavelet and also the frequency content of the data (Millar and Bancroft, 2004).

Plessix (2007) studied the effects of using multigrid cycles for the 3D frequency domain wave equation migration. He considered the result of using the multigrid method on the undamped wave equation at seismic frequencies.

In order to understand how the method of multigrid works, it is necessary that some properties of the Jacobi (and Gauss-Seidel) method be explained in detail.

2.2. Methods of Jacobi and Gauss-Seidel

In equation (14) ($\boldsymbol{Au = b}$), $\boldsymbol{u}$ is the unknown desired exact solution. If $\boldsymbol{v}$ is considered as an approximation to the exact solution, $\boldsymbol{u}$, then the (algebraic) error, $\boldsymbol{e}$, is the difference between these two solutions:

$$\boldsymbol{e = u - v}. \tag{15}$$

The norm of vector $\boldsymbol{e}$ is usually expressed as the error (Briggs et. al., 2000). The maximum norm, $\|\boldsymbol{e}\|_\infty$, is the largest absolute value of the vector elements:

$$\|\boldsymbol{e}\|_\infty = max|e_j|. \tag{16}$$

The euclidean norm or L2-norm, $\|\boldsymbol{e}\|_2$, of a vector $\mathbf{e}$ with n elements is expressed by (Briggs et. al., 2000):

$$\|\boldsymbol{e}\|_2 = \sqrt{\sum_{j=1}^{n} e_j^2}\ . \tag{17}$$

Because $\boldsymbol{u}$ is unknown, $\boldsymbol{e}$ is not directly computable. Therefore, residual, a measureable version of the error, is defined by:

$$\boldsymbol{r = b - Av}. \tag{18}$$

Residuals can be written as:

$$\boldsymbol{r = Au - Av = A(u - v) = Ae}. \tag{19}$$

Equation (19) ($\boldsymbol{r = Ae}$) is called the "residual equation" (Briggs et. al., 2000). With $\boldsymbol{v}$ as an approximation to $\boldsymbol{u}$, $\boldsymbol{r}$ is computable from equation (18). Again solving the residual equation for $\boldsymbol{e}$, gives a new approximate solution using equation (15) in the form of $\boldsymbol{u = v + e}$. Substitution of the residual equation into equation (15) gives:

$$\boldsymbol{u = v + P^{-1}r}, \tag{20}$$

where $\boldsymbol{P \cong A}$, is a preconditioner. This suggests iterations in the form of (Briggs et. al., 2000):

$$\boldsymbol{v^{k+1} = v^k + P^{-1}r} \tag{21}$$

In the Jacobi method, preconditioner $\boldsymbol{P}$ is the diagonal matrix of $\boldsymbol{A}$: $\boldsymbol{P} = \boldsymbol{D}$. With this in mind, and splitting matrix $\boldsymbol{A}$ to $\boldsymbol{D}$, a diagonal matrix, and $-(\boldsymbol{L}+\boldsymbol{U})$, summation of the lower and upper triangle matrices, (or $\boldsymbol{A} = \boldsymbol{D} - \boldsymbol{L} - \boldsymbol{U}$), equation (14) can be written as:

$$\boldsymbol{D}\boldsymbol{v} - (\boldsymbol{L}+\boldsymbol{U})\boldsymbol{v} = \boldsymbol{b} \text{ or } \boldsymbol{D}\boldsymbol{v} = (\boldsymbol{L}+\boldsymbol{U})\boldsymbol{v} + \boldsymbol{b}. \tag{22}$$

This leads to the following equation:

$$\boldsymbol{v} = \boldsymbol{D}^{-1}(\boldsymbol{L}+\boldsymbol{U})\boldsymbol{v} + \boldsymbol{D}^{-1}\boldsymbol{b}, \tag{23}$$

which suggests the Jacobi iterations in the form of (Briggs et. al., 2000):

$$\boldsymbol{v}^{k+1} = \boldsymbol{D}^{-1}(\boldsymbol{L}+\boldsymbol{U})\,\boldsymbol{v}^{k} + \boldsymbol{D}^{-1}\boldsymbol{b}. \tag{24}$$

The advantage of choosing $\boldsymbol{D}$, the diagonal matrix of $\boldsymbol{A}$, as the preconditioner is that it is easily invertible. The inverse of a diagonal matrix can be simply found by inversing the nonzero (diagonal) elements of that matrix (Strang, 1986).

By definition of the Jacobi iteration matrix, $\boldsymbol{R}_J$, as $\boldsymbol{R}_J = \boldsymbol{D}^{-1}(\boldsymbol{L}+\boldsymbol{U})$, the Jacobi method can also be expressed by (Briggs et. al., 2000):

$$\boldsymbol{v}^{k+1} = \boldsymbol{R}_J\boldsymbol{v}^{k} + \boldsymbol{D}^{-1}\boldsymbol{b}. \tag{25}$$

The weighted Jacobi method is a modification to the Jacobi method in the form of (Briggs et. al., 2000):

$$\boldsymbol{v}^{k+1} = ((1-w)\boldsymbol{I} + w\boldsymbol{R}_J)\,\boldsymbol{v}^{k} + w\boldsymbol{D}^{-1}\boldsymbol{b},\ 0 < w < 2, \tag{26}$$

where $w \in \mathcal{R}$ is weighting factor. By defining weighted Jacobi iteration, $\boldsymbol{R}_w = (1-w)\boldsymbol{I} + wR_J$, the Jacobi method has the following matrix shapes (Briggs et. al., 2000):

$$\boldsymbol{v}^{k+1} = \boldsymbol{R}_w v^{k} + w\boldsymbol{D}^{-1}\boldsymbol{b}, \tag{27}$$

and

$$\boldsymbol{v}^{k+1} = \boldsymbol{v}^{k} + w\boldsymbol{D}^{-1}\boldsymbol{r}^{k}, \tag{28}$$

or component forms (Saad, 2000):

$$v_i^{k+1} = \frac{1}{a_{ii}}\left(b_i - \sum_{\substack{j=1 \\ j\neq i}}^{N} a_{ij}\,v_j^{k}\right), i = 1,2,\dots,N, \tag{29}$$

where k is the iteration number and i is the component number of the vectors $\boldsymbol{b}$ and $\boldsymbol{v}$. Starting from an initial value for $\boldsymbol{v}^0$, in each iteration all components of $\boldsymbol{v}^{k+1}$are calculated, then $\boldsymbol{v}^{k}$ is replaced by $\boldsymbol{v}^{k+1}$. This procedure repeats until the desired convergence is achieved.

In the Gauss-Seidel method, each component is replaced as soon as it is updated, which leads to (Saad, 2000):

$$v_i^{k+1} = \frac{1}{a_{ii}}\left(-\sum_{j=1}^{i-1} a_{ij}\, v_j^{k+1} - \sum_{j=i+1}^{N} a_{ij}\, v_j^{k} - b_i\right), i = 1,2,\dots,N, \qquad (30)$$

This reduces the necessary memory to keep all components of v^{k+1}before updating and decreases the number of iterations for the same rate of convergence. By defining the Gauss-Seidel iteration by $\boldsymbol{R}_G = (\boldsymbol{D} - \boldsymbol{L})^{-1}\boldsymbol{U}$, in the matrix form, the Gauss-Seidel method may be expressed by:

$$\boldsymbol{V} \leftarrow \boldsymbol{R}_G\, \boldsymbol{V} + (\boldsymbol{D} - \boldsymbol{L})^{-1}\boldsymbol{b}, \qquad (31)$$

where "←" shows the displacement of elements.

2.3. Analysis of Jacobi Convergence and the multigrid method

The convergence of the Jacobi (and the Gauss-Seidel) iterations is guaranteed if and only if the magnitude of all eigenvalues of $\boldsymbol{R}_J$ be less than 1 (Strang, 1986):

$$|\lambda(\boldsymbol{R}_G)| < 1 \qquad (32)$$

The spectral Radius of a matrix, ρ, is the maximum amount of its eigenvalues: $\rho(\boldsymbol{R}_G) = max|\lambda(\boldsymbol{R}_G)|$. The Jacobi and the Gauss-Seidel methods converge if and only if $\rho(\boldsymbol{R}_G) < 1$. The speed of convergence depends on the amount of $\rho(\boldsymbol{R}_G)$. Smaller $\rho(\boldsymbol{R}_G)$ causes faster convergence to the solution (Strang, 1986).

In order to explain the role of the Jacobi iterations in the multigrid method, let us consider a system of equation (14) where $\boldsymbol{A}$ is the second difference matrix:

$$\boldsymbol{A} = \begin{bmatrix} 2 & -1 & 0 & 0 & & \\ -1 & 2 & -1 & 0 & \cdots & \\ 0 & -1 & 2 & -1 & & \\ & & \vdots & & \ddots & \vdots \\ & & & & \cdots & 2 \end{bmatrix}. \qquad (33)$$

Therefore, the Jacobi iteration matrix is:

$$\boldsymbol{R}_J = \begin{bmatrix} 0 & 1 & 0 & 0 & & \\ 1 & 0 & 1 & 0 & \cdots & \\ 0 & 1 & 0 & 1 & & \\ & & \vdots & & \ddots & \vdots \\ & & & & \cdots & 0 \end{bmatrix}. \qquad (34)$$

The eigenvalues of $\boldsymbol{A}$ are $\lambda_j(\boldsymbol{A}) = 2 - 2\cos j\theta$, where $\theta = \frac{\pi}{N+1}$. Thus, $\lambda_j(\boldsymbol{R}_J) = \lambda\left(I - {}^{1}/_{2}\ \boldsymbol{A}\right) = \cos j\theta < 1$ and convergence is guaranteed. For example if $N = 4$, then $\boldsymbol{R}_J$ has four eigenvalues as (Briggs et. al., 2000):

$$\lambda_j = \cos\frac{\pi}{5}, \cos\frac{2\pi}{5}, \cos\frac{3\pi}{5}\ and\ \cos\frac{4\pi}{5}\ \left(= -\cos\frac{\pi}{5}\right). \qquad (35)$$

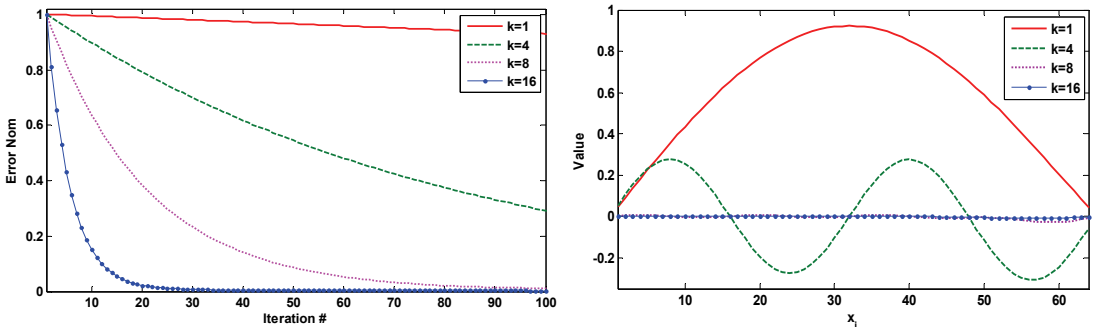
As observed, λs are larger for smaller angles. It means that the convergence is slower for lower frequencies of the solution. This is a general property of the Jacobi iterations.

Removing high frequency contents from residuals in the Jacobi (and the Gauss-Seidel) first few iterations produces a smooth (includes mostly low frequency contents) error vector. This "smoothing" property is shown in an example; consider the system of equation

$$\boldsymbol{Ax} = \mathbf{0}, \tag{36}$$

where $\boldsymbol{A}$ is the second difference matrix in equation (33), with $n = 64$. The trivial exact solution is $\boldsymbol{x} = \mathbf{0}$, therefore, $\boldsymbol{e} = -\boldsymbol{v}$. Let us apply the weighted Jacobi method (with $w = 2/3$) to solve this equation with $\boldsymbol{x}0_j = sin\left(\frac{jk\pi}{n}\right), 0 \leq j \leq n, 1 \leq k \leq n-1$, the Fourier modes with the frequency k, as the initial guess.

Figure 2a shows the convergence rate of the weighted Jacobi method for different initial values with $k = 1, 4, 8$ and 16. There is a faster convergence to the solution by choosing an initial guess with higher frequency contents (larger ks). The solutions after 100 iterations are shown in figure 2b. Again, with higher frequency content in the initial value, the solution is closer to the exact solution, $\boldsymbol{x} = \mathbf{0}$.

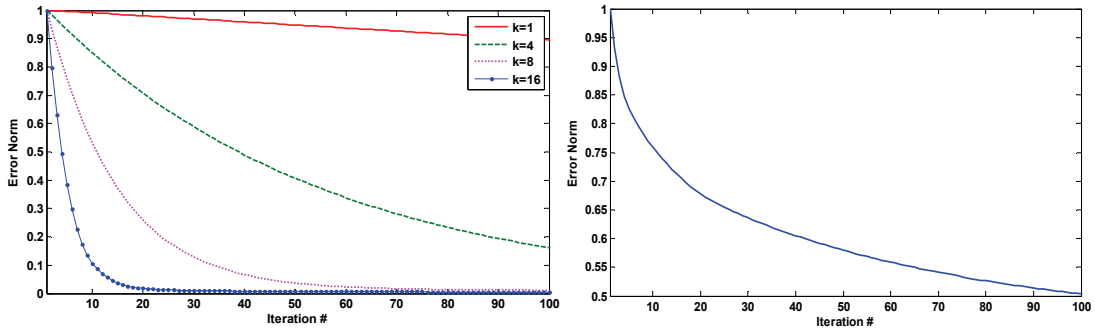


a) b)

FIG. 2. a) The maximum norm error versus number of iterations is plotted for the weighted Jacobi iterations for different initial values to solve equation 36. b) Solution after 100 iterations.

The weighted Gauss-Seidel method behaves similarly for this equation, as shown in figure 3a. Figure 3b shows the convergence when initial values are a superposition of the four Fourier modes with both low and high frequencies. The initial fast decrease corresponds to the presence of high frequencies in the initial value.

Now let consider the initial guess to be a superposition of the four previous Fourier modes, $x0_j = \frac{1}{4}\left[sin\left(\frac{j\pi}{n}\right) + sin\left(\frac{4j\pi}{n}\right) + sin\left(\frac{8j\pi}{n}\right) + sin\left(\frac{16j\pi}{n}\right)\right]$. The convergence for the weighted Jacobi iterations starting with this initial guess is shown in figure 3b. In first five iterations the error decreases rapidly. Then, the convergence becomes slower. The faster decrease corresponds to the presence of the high frequency components in the initial value and less rapid decrease is due to the lower frequency components of the initial value (Strang, 1986).



a) b)

FIG. 3. a) The maximum norm error versus number of iterations for the weighted Gauss-Seidel method with different initial values. b) Convergence when initial values are superposition of the four Fourier modes with both low and high frequencies.

This property, leads to the multigrid idea. In the multigrid method , an iterative solver (Jacobi or Gauss-Seidel, generally), produces low frequency contents in the residual after a few iterations of equation (14). By restriction, the kernel of the main problem and its residual are transferred to a coarser grid (scale), where the low frequency components act as the high frequency components. Solving the original equation with this initial guess, gives a solution which also contains more low frequency components than the solving equation with a vector of zeros as the initial guess (Strang, 1986).

A v-cycle multigrid method starts with a few (three for instance) iterations on the fine grid, then the error is transferred to a coarser grid by a restriction process. Iterations are performed on the coarse grid, and the interpolated results are used as the starting point in the Jacobi method on the finer (main) grid.

There are two processes in the multigrid method for transferring the problem to the coarser or finer grid (Strang, 1986): multiplying by the restriction matrix, $\boldsymbol{R}$, which transfers the problem from the finer grid to the coarse grid; and multiplying by the Interpolation matrix, $\boldsymbol{I}$, which returns the problem back to the finer grid.

A v-cycle multigrid method includes only two grids: fine grid (with the size of the main problem) and coarse grid (Figure 4). Iterations (weighted Jacobi or Gauss-Seidel) start with zero as the initial model to solve equation $\boldsymbol{A}^h\boldsymbol{u} = \boldsymbol{b}^h$ on the fine grid where h corresponds to the size of the main grid which is the finest grid. After a few iterations, the residual to equation $\boldsymbol{A}^h\boldsymbol{u} = \boldsymbol{b}^h$ is calculated as $\boldsymbol{r}^h = \boldsymbol{b}^h - \boldsymbol{A}^h\boldsymbol{u}^h$. Then multiplication of the Restriction matrix converts $\boldsymbol{r}^h$ to the coarser grid $\boldsymbol{r}^{2h}$, where $2h$ corresponds to the first coarser grid with size equal to half of the original grid size. Solving $\boldsymbol{A}^{2h}\boldsymbol{e}^{2h} = \boldsymbol{r}^{2h}$ for $\boldsymbol{e}^{2h}$ on the coarse grid requires a few more iteration, and then the solution, $\boldsymbol{e}^{2h}$, must be interpolated to the fine grid as $\boldsymbol{e}^h$. Finally, iterations to solve $\boldsymbol{A}^h\boldsymbol{u} = \boldsymbol{b}^h$ start with the improved initial value $\boldsymbol{u}^h + \boldsymbol{e}^h$. A few iterations on this problem size/grid returns a solution which include both low and high frequency contents (Strang, 1986; Briggs et. al., 2000).

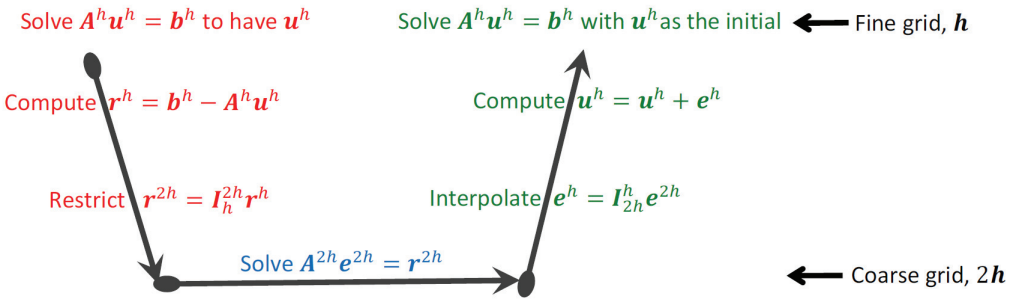


FIG. 4. Schematic v-cycle multigrid. Main problem is solved on the fine grid and residuals which are restricted to the coarse grid with two grids are used to solve equation in the coarse grid. Solutions are interpolated to the fine grid and used as the initial value for iterations on the main grid.

The simplest algorithm in the multigrid method is a v-cycle. It is possible to calculate the residuals in the coarse grid, $\boldsymbol{r}^{2h} = \boldsymbol{r}^{2h} - \boldsymbol{A}^{2h}\boldsymbol{e}^{2h}$, and restrict it to a coarser grid $4h$ and repeat the procedure to a very coarse grid, nh!. This algorithm is known as V-cycle. W-cycle algorithm performs more iteration on the coarser grids (Strang, 1986; Briggs et. al., 2000).

In the full multigrid method (FMG), iteration starts on the coarsest grid, the solution is interpolated and used as the initial value for one step finer grid. A v-cycle improves the result. Then, result will be used for the finer grid, a V-cycle improves this result and the process continues to arrive to the finest grid which is the size of the initial problem (Strang, 1986; Briggs et. al., 2000).

To compare the advantages of using the multigrid method to the Jacobi method, consider equation

$\boldsymbol{Ax} = \boldsymbol{b}$, where $\boldsymbol{A}$ is the mentioned second difference matrix, and $b_j = \frac{1}{4}\left[sin\left(\frac{j\pi}{n}\right) + sin\left(\frac{4j\pi}{n}\right) + sin\left(\frac{8j\pi}{n}\right) + sin\left(\frac{16j\pi}{n}\right)\right]$, where $n = 64$. Figure 5 shows the convergence of the Jacobi method after 10 iterations and the full multigrid method after 5 iterations on each grid size. After 10 iterations, the residual norm in the Jacobi method reaches 75% where in the full multigrid method it reaches 13% after only 5 iterations on each grid. The time spent in the coarse grids is small in comparison to the time spent in the main problem (finest grid). Therefore, the multigrid method is cheaper than applying many Jacobi iterations to arrive to the same solution.

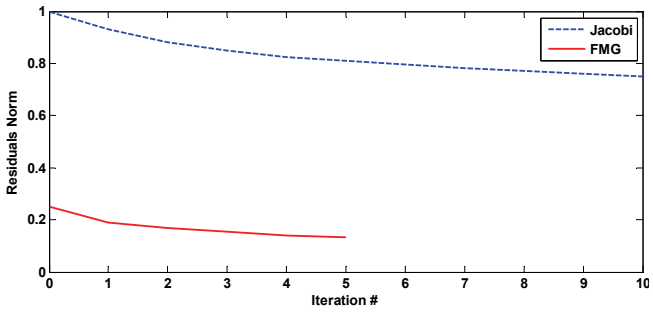


FIG. 5. The residuals norm error versus number of iterations is plotted for the weighted Jacobi iterations and full multigrid iteration in the last cycle. The FMG converges to %13 of the starting point in the residuals norm where Jacobi reaches %75 after 10 iterations.

3 SOLVING LSPSTM USING STANDARD MULTIGRID METHODS

3.1 Applying Standard Multigrid Solvers to the LSPSTM Equation

In using the explicit form of the matrix $\boldsymbol{G}'$ (migration), instead of the migration operator, once $\boldsymbol{G}'$ is calculated inside the migration operator, there is no requirement for the recalculation of the DSR equation, weight function, rho filter, and interpolation coefficients on each iteration. All these calculations are hidden in the matrix $\boldsymbol{G}$ and a proper matrix-vector multiplication is equal to the performing of all necessary procedures of migration. Therefore, multiplication of $\boldsymbol{G}$ (or $\boldsymbol{G}'$) with vectors may be cheaper than migration or modelling. This seems to be one advantage of using the explicit form of $\boldsymbol{G}$ instead of the operator form, where in a LSCG scheme these processes are repeated several times.

Mathematically, if a problem is solvable by a multigrid method, it will be solved faster and with better recovery of low frequency contents than many other methods such as Successive Over Relaxation (SOR) and CG (Stuben, 2002). The multigrid method uses Jacobi or Gauss-Seidel algorithms for smoothing. In both methods it is necessary to extract the diagonal elements of matrix $\boldsymbol{A}$ (in equation 14) and invert it. Therefore, it is necessary to have $\boldsymbol{G}$ in the explicit matrix form. From this point of view, the multigrid method may increase the speed of convergence, but it increases the required memory to load matrix $\boldsymbol{G}$.

In order for convergence of the Jacobi and Gauss-Seidel methods to be guaranteed, it is necessary that the magnitude of every eigenvalue of $\boldsymbol{R}_J$ be less than 1 ($|\lambda(\boldsymbol{R}_G)| < 1$). Equivalently, matrix $\boldsymbol{A}$ (in equation 14) must be diagonally dominant. Otherwise, these methods do not converge to the solution. A matrix $\boldsymbol{A}$ is strictly diagonally dominant if for all elements a_{ij} of the matrix:

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}| \tag{37}$$

where i and j are the number of rows and columns, respectively.

However, the experience with the explicit form of matrices $\boldsymbol{G}$ and $\boldsymbol{G'G}$, for different geometries of seismic data acquisition patterns, shows that they are relatively dense matrices. The presence of many non-zero and relatively large elements in each row

prevents the diagonal element from being dominant (larger than sum of absolute values of non-zero elements).

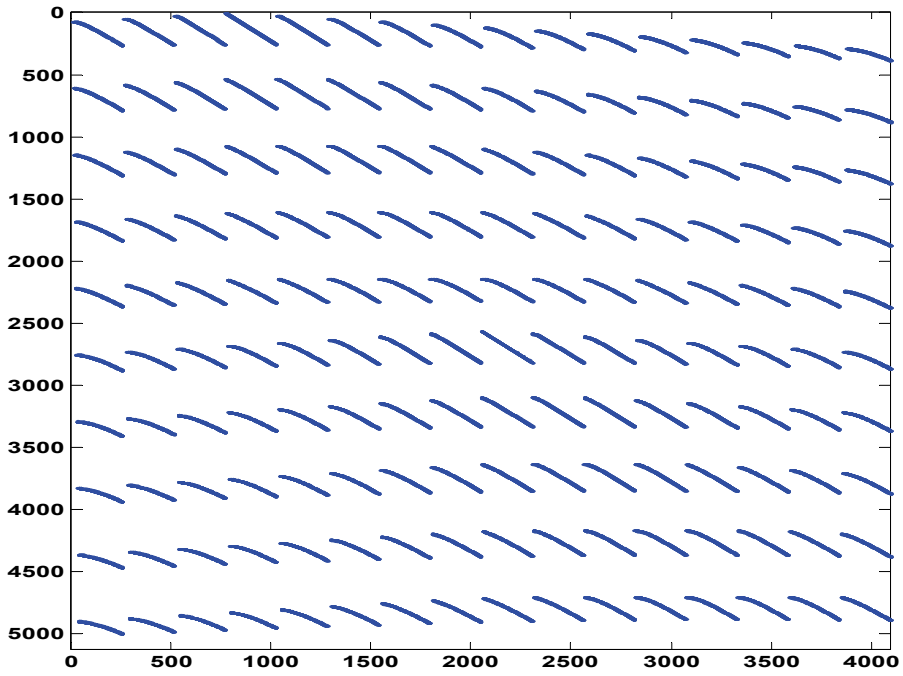
For instance, for a modeling operator with two sources with 180 m interval spacing and five receivers per source with 72 m interval spacing and with a model 632 m long distance and 0.512 seconds depth, matrix $\boldsymbol{G'G}$ has 10% nonzero elements as shown in figure 6b. Figure 6a shows the non-zero elements of the matrix $\boldsymbol{G}$, the modeling operator. Figure 6c shows the ratio of absolute values of diagonal elements to the sum of absolute values of nondiagonal elements for each row of the matrix $\boldsymbol{G'G}$. This example and many other examples showed that the Hessian matrix is a dense and diagonally non-dominant matrix. Numerical examples showed that adding some white noise or a reasonably large constant scalar to the diagonal elements of the Hessian matrix does not change it to a (strictly) diagonally dominant matrix.

The experiment applies the restriction operator several times on the Hessian matrix, $\boldsymbol{G'G}$, and converts it to a very coarse matrix. It does not change the diagonal non-dominancy of the matrix. In fact, the restricted matrix is denser than the original matrix. Therefore, the idea of moving the main problem from the main grid to a coarser grid, performing the standard multigrid method on the coarse grid and using the interpolated result as the initial value for another iterative method such as CG, is not applicable.

Another possibility would be reducing the size of $\boldsymbol{G}$ (and $\boldsymbol{G'G}$) by solving for each column of the model at each time ($ix = 1, \dots, Nx$). This procedure reduces the size of $\boldsymbol{G'G}$ from $(Nx \times Nz)^2$ to only $(1 \times Nz)^2$ where Nx and Nz are numbers of model grid in the horizontal (usually equal to the number of CMPs) and vertical directions, respectively.

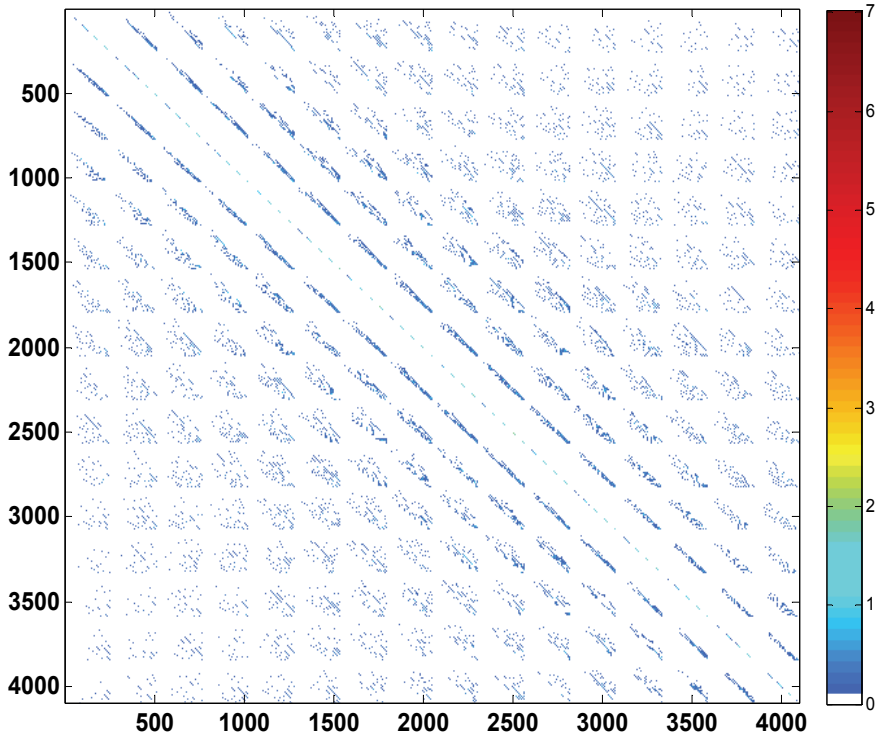
By inverting for each column separately and putting the resulted inverted columns next to each other, the inversion image is achieved. This separation is valid only when horizontal velocity variations of the subsurface are small. However, matrix $\boldsymbol{G_{ix}'G_{ix}}$, is not a diagonally dominant matrix and solvable by Jacobi or gauss Seidel methods.

Therefore, the main precondition for the Kirchhoff LSPSTM problem to be solvable by standard solvers of the multigrid methods is violated. Therefore, at least in its standard form, the multigrid method is not an effective solver for the LSPSTM problem.

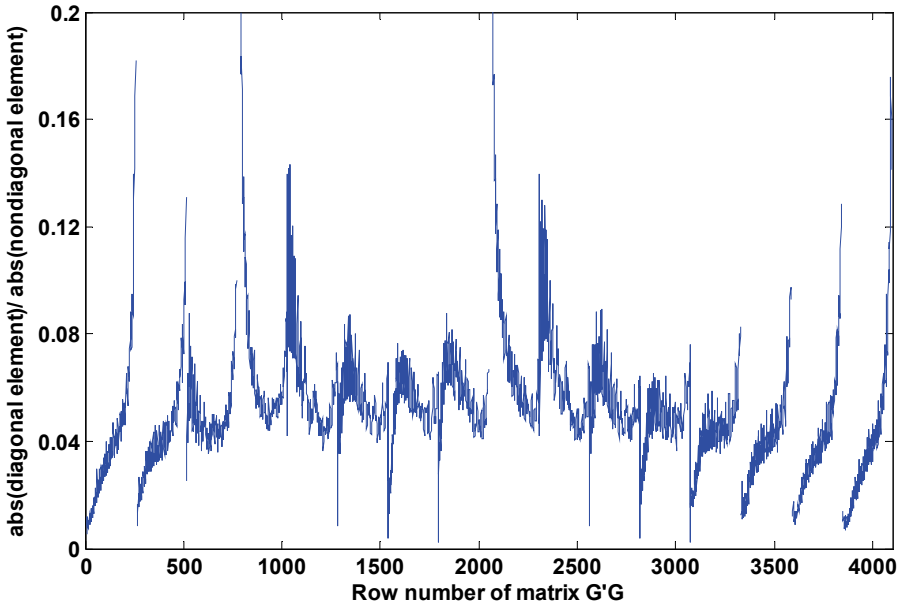


a)

FIG. 6. a) Non-zero elements of the matrix $\boldsymbol{G}$ for the mentioned geometry. Matrix is dense and there are many non-diagonal elements.



b)



c)

FIG. 6. b) Non-zero elements of matrix $\boldsymbol{G}'\boldsymbol{G}$, matrix is more dense than matrix $\boldsymbol{G}$. c) The ratio of absolute values of diagonal elements to the sum of absolute values of nondiagonal elements for each row of $\boldsymbol{G}'\boldsymbol{G}$.

3.2. The problem with the Large Size of $\boldsymbol{G}$ matrix

In order to solve equation 10 by multigrid methods, matrices $\boldsymbol{G}$ (and $\boldsymbol{G}'\boldsymbol{G}$) need to be calculated. The size of matrix $\boldsymbol{G}$ equals the number of data values (number of traces multiplied by the number of samples per trace) multiplied by the number of values in the migration image (usually the number of CMPs, Nx, multiplied by the number of depth samples, Nz). Therefore, especially in 3D seismic surveys, $\boldsymbol{G}$ can be large enough to be impossible to be loaded in the memory of today's computers.

For example, a 20 km long 2D seismic line in the Gulf of Mexico includes almost 150,000 traces, with 1800 CMPs, each has 1750 samples. The size of matrix $\boldsymbol{G}$ for a time migration for this 2D line is $150{,}000 \times 1751 \times 1751 \times 1800 = 8.3 \times 10^{14}$ samples which require more than 6000 TB memory to be loaded. The most efficient multigrid algorithms need at least twice this memory. In the case of a marine 3D survey, this number can be hundreds of times larger.

Many examples show that multiplication of $\boldsymbol{G}$ or $\boldsymbol{G}'$to the vectors are more costly than applying migration and modelling operators and this is due to the large size and non-sparseness of $\boldsymbol{G}'\boldsymbol{G}$ matrix.

4. MULTIGRID LSPSTM WITH OTHER ITERATIVE SOLVERS

In order to be effective, multigrid techniques require an iterative method to be a smoother when used as the solver. The smoother must be able to find the high frequency contents of the solution and leave the low frequency contents in the residuals after a few iterations. When the multigrid method is not effective using its standard iterative smoothers, Jacobi and Gauss-Seidel for example, other iterative methods may be examined.

4.1. Kaczmarz Method

The kaczmarz method is an iterative method for solving systems of linear equation (14). The method of Kaczmarz initially invented to solve tomography problems. It is not as fast as many other iterative solvers such as CG. However, it does not require the matrix $\boldsymbol{A}$ (in equation 14) to be diagonally dominant or positive definite (Kaczmarz method, 2009).

In this method, system of linear equation 14 (where $\boldsymbol{A}$ is a general $M \times N$ matrix) is being considered as the composition of M systems $\boldsymbol{A}_{i,:}\boldsymbol{u} = \boldsymbol{b}_i$, where $\boldsymbol{A}_{i,:}$ is ith row of $\boldsymbol{A}$. Each of M $\boldsymbol{A}_{i,:}\boldsymbol{u} = \boldsymbol{b}_i$ is a n-dimensional hyperplane in the R^m space (Aster, 2005).

The algorithm starts with an initial guess $\boldsymbol{u}^0$. Projecting the initial guess onto the hyperplane defined by $\boldsymbol{A}_{i,:}\boldsymbol{u} = \boldsymbol{b}_i$, $i = 1$, returns $\boldsymbol{u}^1$. The same procedure for the second row of $\boldsymbol{A}$, $i = 2$, using $\boldsymbol{u}^1$, returns $\boldsymbol{u}^2$. The procedure is repeated for all M rows. If the desired convergence has not been achieved, the cycle repeats with newly calculated $\boldsymbol{u}$, as many times as preferred (Aster, 2005).

The Kaczmarz algorithm is an iterative method with smoothing properties (McCormick, 1987). However, it is more effective on sparse and rank efficient matrices. Numerical experiments showed that it is not a good solver for the LSPSTM problem.

In addition to the convergence problem, the limitation of computer memory is still a problem. We are looking for a method to avoid loading big matrixes of $\boldsymbol{G'G}$ leads to the CG method.

4. 2. CG Method

When $\boldsymbol{A}$ is a positive definite matrix, Steepest Descent is a powerful method to solve equation (14). When or $\boldsymbol{A}$ (or $\boldsymbol{G'G}$ in equation 10) is positive definite, the quadratic form of $f(\boldsymbol{u}) = \frac{1}{2}\boldsymbol{u}^T\boldsymbol{A}\boldsymbol{u} - \boldsymbol{b}^T\boldsymbol{u} + c$ has a paraboloid shape with a minimum which is equivalent to the solution to equation (14) (Shewchuk, 1994). In the method of Steepest Descent the algorithm starts at an arbitrary point and follows the opposite direction of the maximum gradient until the gradient changes. Then algorithm changes the direction to find a new minimum. The procedure continues until it arrives at the minimum which is the exact solution in case of linear problem.

In the method of Conjugate Gradient (CG) (Hestenes and Steifel, 1952) which is an improvement to the Steepest Descent, the new direction is orthogonal to all previous directions. Therefore, CG converges to the solution in fewer iterations than Steepest Descent (Shewchuk, 1994).

CG requires that $\boldsymbol{G'G}$ be symmetric and positive definite matrix. Least-Squares Conjugate Gradient (LSCG), a modified version of CG method, does not require this condition and directly works with $\boldsymbol{G}$ and $\boldsymbol{G'}$ matrices (Scales, 1987). If equation $\boldsymbol{Gm} = \boldsymbol{d}$ is an overdetrermined problem, then $\boldsymbol{G}'\boldsymbol{G}$ is nonsingular and LSCG converges to solve equation $\boldsymbol{G}'\boldsymbol{G}\boldsymbol{m}_{\boldsymbol{LS}} = \boldsymbol{G}'\boldsymbol{d}$. However, since the condition number of matrix $\boldsymbol{G}'\boldsymbol{G}$ is the square of the condition number of matrix $\boldsymbol{G}$, convergence of the LSCG method is slower than convergence of the CG method.

However, by replacing the method of CG with LSCG the multiplication of the matrices $\boldsymbol{G}$ or $\boldsymbol{G}'$ with vectors is replaced by applying forward (seismic modelling or de-migration) or adjoint (seismic migration) operators to the model or data, respectively. This procedure reduces memory required to load big matrixes $\boldsymbol{G}$ and $\boldsymbol{G}'$ into the computer and also avoids big matrix-vector multiplications.

A LSCG's algorithm with Fletcher-Reeves (Shewchuk, 1994) line search method to solve $\boldsymbol{G}'\boldsymbol{G}\boldsymbol{m}_{\boldsymbol{LS}} = \boldsymbol{G'd}$, is:

$\boldsymbol{m}_0 = \boldsymbol{0}$

$\boldsymbol{s}_0 = \boldsymbol{d} - \boldsymbol{G}\boldsymbol{m}_0$

$\boldsymbol{r}_0 = \boldsymbol{G's}_0$

$\boldsymbol{p}_0 = \boldsymbol{r}_0$

$\boldsymbol{q}_0 = \boldsymbol{G}\boldsymbol{p}_0$

$i = 0$

$for\ i = 1: \#\ of\ iterations$

$\alpha_{i+1} = \dfrac{\boldsymbol{r}_i \,.\, \boldsymbol{r}_i}{\boldsymbol{q}_i \,.\, \boldsymbol{q}_i}$

$\boldsymbol{m}_{i+1} = \boldsymbol{m}_i + \alpha_{i+1}\boldsymbol{p}_i$

$\boldsymbol{s}_{i+1} = \boldsymbol{s}_i - \alpha_{i+1}\boldsymbol{q}_i$

$\boldsymbol{r}_{i+1} = \boldsymbol{G}'\boldsymbol{s}_{i+1}$ $\quad *$ migration

$\beta_{i+1} = \frac{\boldsymbol{r}_{i+1} \,.\, \boldsymbol{r}_{i+1}}{\boldsymbol{r} \,.\, \boldsymbol{r}}$

$\boldsymbol{p}_{i+1} = \boldsymbol{r} + \beta\boldsymbol{p}_i$ $\quad \dagger$

$\boldsymbol{q}_{i+1} = \boldsymbol{G}\boldsymbol{p}_{i+1}$ $\quad **$ modeling

$i = i + 1$

$endfor$

There are two multiplications of $\boldsymbol{G}$ or $\boldsymbol{G}'$ matrices with vectors (lines marked by $*$ and $**$, respectively) in each iteration. Since there is not any decomposition of $\boldsymbol{G}$ or $\boldsymbol{G}'$ in the CG algorithm, it is possible to use operators instead of multiplication of explicit forms of $\boldsymbol{G}$ or $\boldsymbol{G}'$ matrices with vectors. However, each iteration in the LSCG has twice the computational cost of one migration (or modelling). For example, time cost of 10 iterations in the LSCG is equal to time cost of 20 migrations.

For a linear problem with any starting point, Steepest Descent and CG methods always converge to the solution after n iterations (Shewchuk, 1994). In the case of a nonlinear problem, replacing the Fletcher-Reeves line search (in line marked by †) with the non-negative Polak-Ribiere-Polyak line search,

$$\beta_{i+1} = max\left\{0, \frac{\boldsymbol{r}_{i+1} \,.\, (\boldsymbol{r}_{i+1} - \boldsymbol{r}_i)}{\boldsymbol{r} \,.\, \boldsymbol{r}}\right\}, \tag{38}$$

is a better choice.

However, in the case of nonlinear problems, the best result is achievable when the starting point is considered to be close to the global minimum of the quadratic function. In a linear problem, any "rough estimate" of $\boldsymbol{x}$ is suitable as the initial guess (Shewchuk, 1994). If there is not any other information about $\boldsymbol{x}$, $\boldsymbol{x} = \boldsymbol{0}$ is chosen.

Convergence of the CG method depends on the ill-conditioning of the matrix. Convergence is slower for problems with larger condition numbers (more ill-conditioned problems).

4.3. Multigrid CG Versus CG

CG is a powerful method for solving linear equation (14). LSPSTM after a few iterations in CG retrieves a high resolution image of the earth subsurface reflectivity. However, CG does not have smoothing propertieswhich is an essential property for a solver to be used in the multigrid method. In fact, the Steepest Descent and CG methods are "roughers" and not smoothers (Shewchuk, 1994).

To show that CG is not a smoother, let consider solving equation (36) with $\boldsymbol{x}0_j = sin\left(\frac{jk\pi}{n}\right), 0 \leq j \leq n, 1 \leq k \leq n-1$, the Fourier modes with the frequency k, as the initial value. Figure 7 shows the convergence for different frequency contents, $k = 1,4,8,16$, in the initial value. There is not any relationship between the rate of convergence and the frequency content of the initial value. Starting with any initial value, CG converges to the solution in less than ten iterations. It is important to mention that the convergence of CG is faster than the convergence of the weighted Jacobi or Gauss Seidelmethods even if the initial value includes mostly high frequency contents as shown in figure 2.

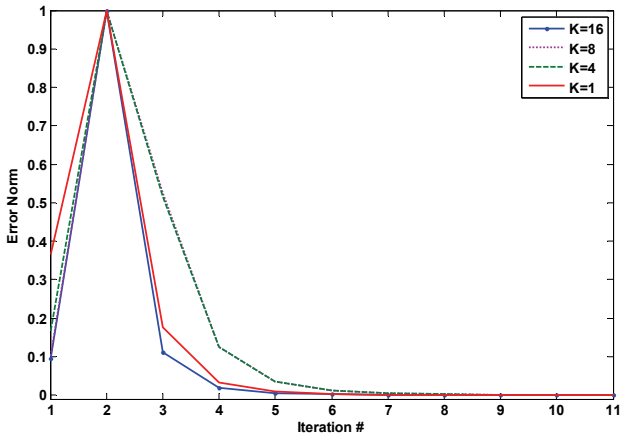


FIG. 7. The maximum norm error versus number of iterations is plotted for the CG iterations with different initial values. No relationship between rate of convergence and frequency content of initial value is visible.

This property is investigated for solving LSPSTM problem. Figure 8a shows the convergence rate of LSCG for the synthetic model of figure 1 with wavelets having different dominant frequencies. This figure shows no better convergence when the wavelet (and consequently the data) has higher frequency content. Therefore, the convergence of the CG method for any linear problem such as LSPSTM does not depend on the frequency contents of the data. The same conclusion is achieved by solving LSPSTM with smoothing in the offset direction as the regularization (Figure 8b). The convergence of LSCG strongly depends on the condition number of the kernel matrix and not on the starting point.

This property is shown on a complex model of the Marmousi data set. It is necessary to mention that the Kirchhoff time migration is not a suitable method of migration for the Marmousi data set which is a geologically complex model. The Marmousi data set needs a depth migration instead of a simple Kirchhoff time migration. However, imaging of the Marmousi data set is not the main goal of this study. The advantages of using multigridmethods over other methods of solving LSPSTM is the main idea for using the Marmousi data set.

Figure 9 shows the convergence of LSPSTM of the Marmousi data filtered by different band-pass filters. In figure 9, L-Pass, B-Pass I, B-Pass II, and H-Pass curves correspond to convergence of LSCG of the Marmousi data passed through a low pass ($f < 15Hz$), band pass ($15\text{Hz} < f < 30Hz$), band pass ($30\text{Hz} < f < 45Hz$), and high pass ($45\text{Hz} < f$) filters, respectively. There is not any clear relationship between the frequency contents of the data and the rate of convergence. However, it seems that there is a faster convergence for the very low frequency (less than *15 Hz*) data which may be related to the roughness property of the CG method.

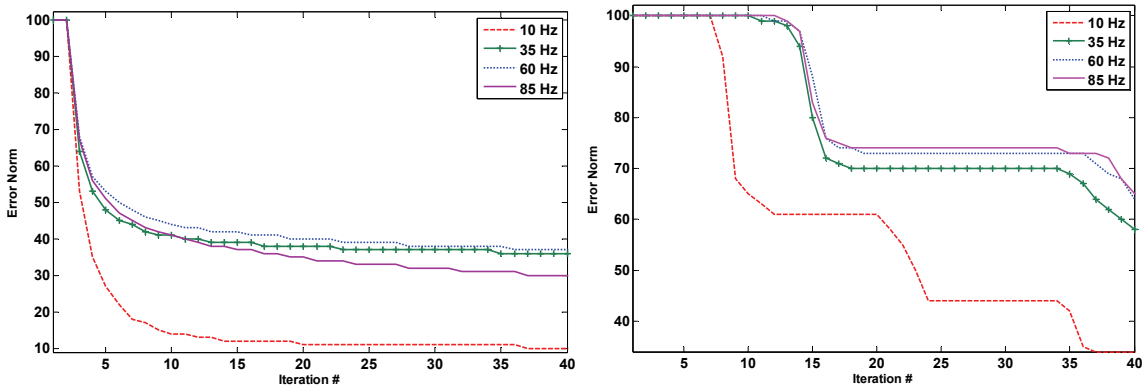


FIG. 8. a) Convergence of LSCG to solve damped LSPSTM for synthetic data of figure 1a with wavelets with different dominant frequencies: 10, 35, 60 and 85 Hz. b) Same result with regularized LSPSTM (smoothing in the offset direction).

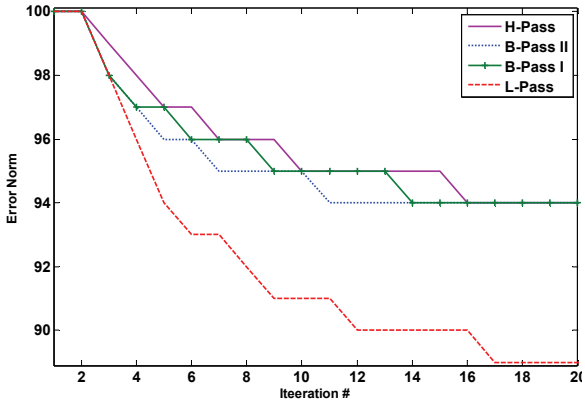


FIG. 9. Convergence of LSCG to solve LSPSTM for the filtered version of the Marmousi data set. L-Pass: $f < 15Hz$, B-Pass I: $15Hz < f < 30Hz$, B-Pass II: $30Hz < f < 45Hz$, H-Pass: $45Hz < f$.

When an iterative method does not converge faster for data with higher frequency content than for data with low frequency content, it does not leave low frequency contents in the residuals to act as a smoother. Therefore, CG (and LSCG) should not be effective solvers for the multigrid method. However, different approaches of applying multigrid LSCG to the LSPSTM equation may be examined.

There are three main approaches for applying multigrid methods to a LSPSTM equation. Restriction and interpolation of a LSPSTM problem can be applied in each direction, horizontal (distance), vertical (time), or both directions of the model in order to transfer the problem to a higher or lower grid size. As seen in the previous section, applying multigrid methods in the vertical (time) direction should not improve the performance of LSM since there is no faster convergence for the lower frequency components of the data.

Using multigrid LSCG with restriction and interpolation in the distance direction is investigated by comparison between multigrid LSCG and LSCG. Restriction to a coarser grid is performed by deleting half of the traces (leaving one trace and removing next one) from the migration image in order to move to a coarser grid.

Figure 10 shows the performance of the multigrid LSCG versus LSCG for the synthetic example of figure 1. Figure 10a shows the true model, figure 10b shows the image from Kirchhoff LSPSTM with five iterations on the LSCG, and figure 10c shows the result of full multigrid LSCG with five iterations on each grid. Comparison between figures 10b and 10c shows that the multigrid method has not improved the result.

Figure 11 shows the result of solving LSPSTM for the Marmousi data set with LSCG (Figure 11a) and multigrid LSCG (Figure 11b). Again there is not much improvement when using multigrid LSCG instead of LSCG.

The same conclusion is reached when solving regularized (smoothing in the offset direction) LSPSTM as shown in figure 12.

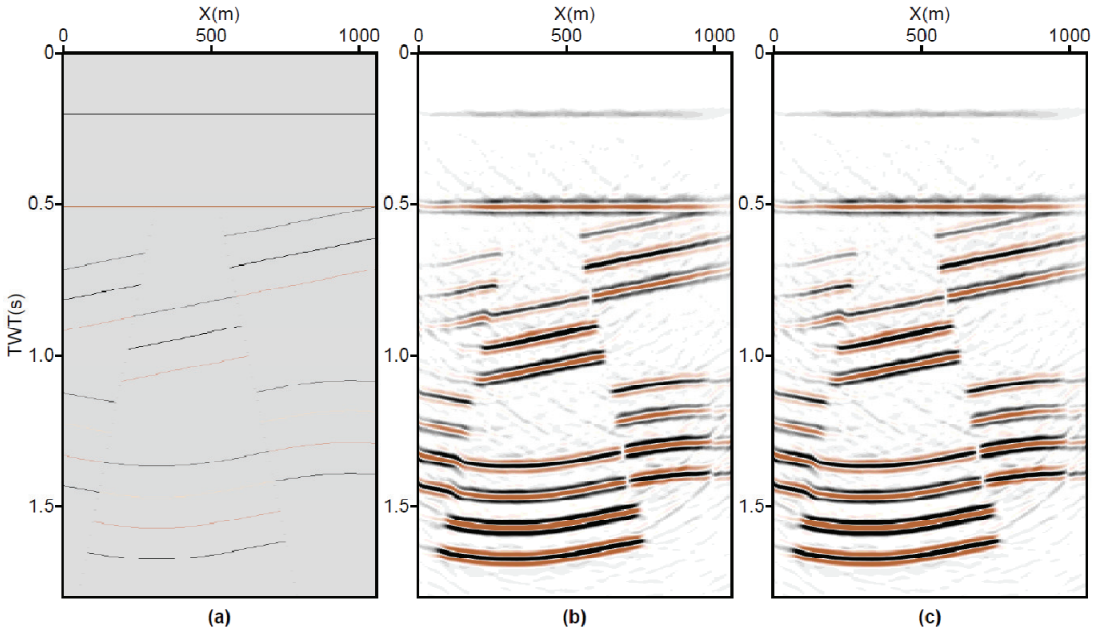


FIG. 10. Comparison between LSCG and multigrid LSCG for the synthetic model of figure 1, a) model, b) LSCG, c) Multigrid CG.

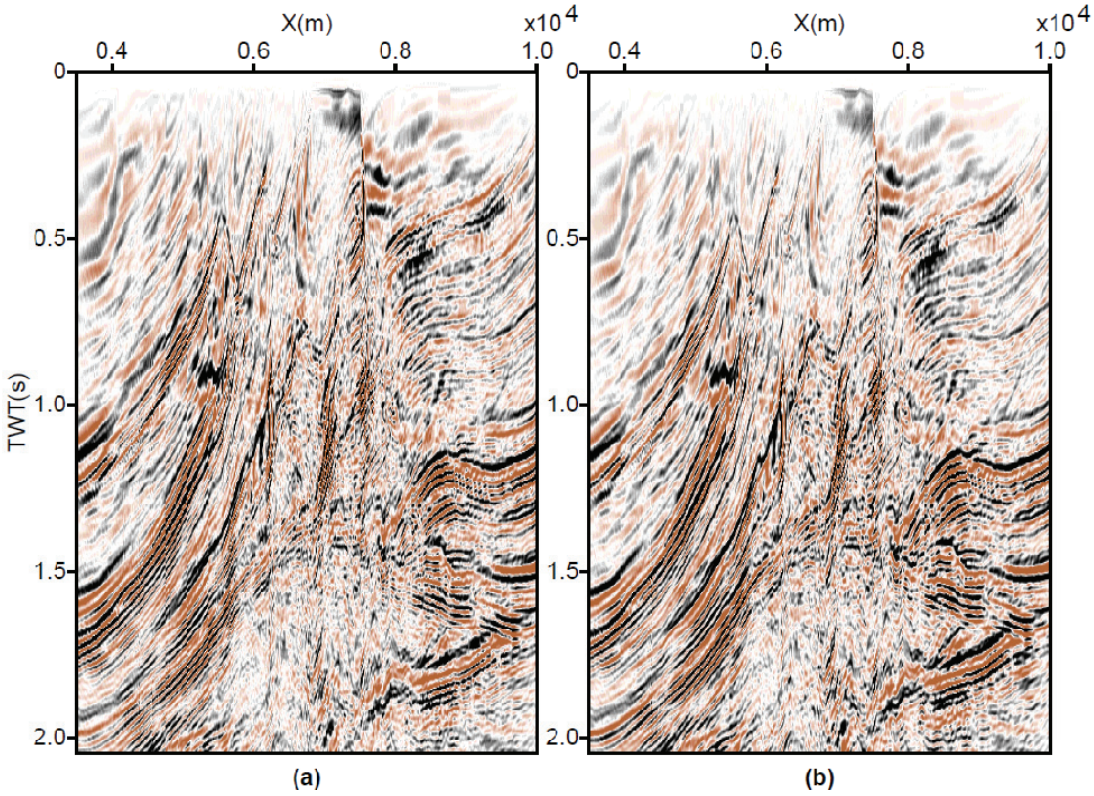


FIG. 11. Comparison between LSCG and multigrid LSCG to solve LSPSTM of the Marmousi data set with 2 iterations on each grid. a) LSCG, b) Multigrid CG.

4.4. Feasibly of Using Reverse-v Cycle Multigrid CG

When CG is a rougher, and not a smoother (Shewchuk, 1994), a new multigrid method , reverse-v (Λ) cycle, may be considered.

In this method, the problem may be solved in the main grid size and then, results which include mostly high frequency contents, interpolated to a finer grid where high frequency contents act as low frequency contents. In this new grid size, the problem is solved and the restricted solution added to the solution in the main grid size to be used as the new initial value in the main problem.

However, this method is not effective in LSPSTM for two reasons: first, solving LSPSTM in a grid finer than the main problem is very costly; second, roughness property of CG is not as obvious as the smoothness property of the Jacobi method (shown in figures 8 and 9). It seems that the convergence of CG is faster with an initial value with very low frequency contents in the initial model. For higher frequencies, there is not any clear relationship between rate of convergence and the frequency contents of the initial value, at least for the mentioned problems.

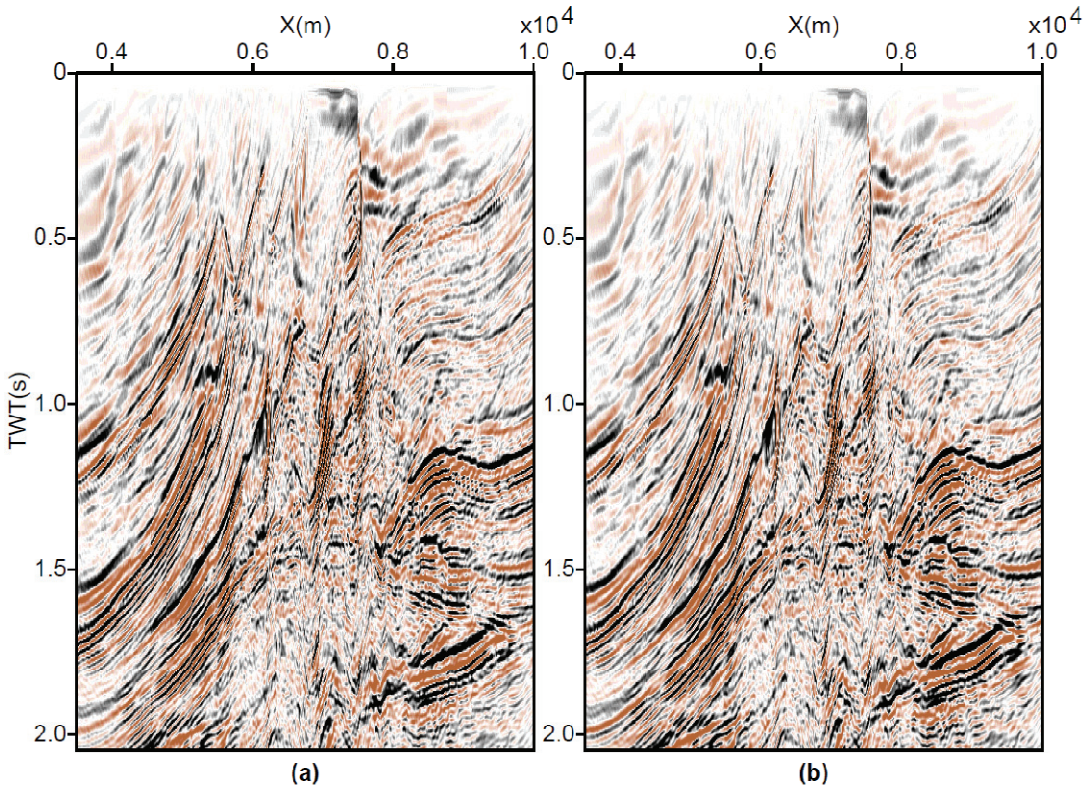


FIG. 12. Comparison between LSCG and multigrid LSCG to solve regularized LSPSTM of the Marmousi data set with 5 iterations on each grid. a) LSCG, b) Multigrid CG.

5. SUMMARY

Solving inverse problems requires solving a large linear equation in the form of equation (14). Numerical examples showed that the LSPSTM problem is not solvable by the Jacobi or Gauss-Seidel iteration methods. Consequently, the standard multigrid

approach which uses Jacobi or Gauss-Seidel algorithms as iterative solvers is not applicable to the described problem. The requirement of large memory size is another problem associated with this method.

The CG algorithm is an effective solver. The LSCG method has the advantage of using operators instead of matrices. However, the CG method does not have smoothing properties. Therefore, using the CG method as the multigrid solver does not increase the speed of convergence. This is shown on synthetic examples.

Using the multigrid method with CG as the iterative solver may slightly reduce the number of iterations for the same rate of convergence, in comparison to the LSCG method, by introducing an initial value. However, it does not reduce the total computational cost.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the sponsors of CREWES project for their continuing support. They also appreciate the faculty staff and computer technicians in the CREWES project.

REFERENCES

Aster, R. C., Borchers, B., and Thurber C. H., 2005, Parameter Estimation and Inverse Problems: Elsevier Academic Press.

Briggs, W. L., Henson, V. E., and McCormick, S. F., 2000, A multigrid tutorial, second edition, SIAM.

Bunk, C., Saleck, F. M., Zaleski, S., and Chavents, G., 1995, Multiscale seismic waveform inversion. Geophysics, 60, No. 5, 1457-1473.

Claerbout, J. F., and Doherty, S. M., 1972, Downward continuation of moveout corrected seismograms: Geophysics, 37, 741-768.

Duquet, B., Marfurt, K. J., and Dellinger, J. A., 2000, Kirchhoff modeling, inversion for reflectivity, and subsurface illumination: Geophysics, 65, 1195-1209.

Gazdag, J., 1978, Wave equation migration with the phase-shift method: Geophysics, 43, 1342-1351.

Gray, S. H., Etgen, J., Dellinger, J., and Whitmore, D., 2000, Seismic migration problems and solutions: Geophysics, 66, No. 5, 1622-1640.

Gulunay, N., 2003, Seismic trace interpolation in the Fourier transform domain: Geophysics, 68, no. 1, 355-369.

Hagedoorn, J. G., 1954, A process of seismic reflection interpretation: Geophysical Prospecting, 2, 85-127.

Hestenes, M., and Steifel, E., 1952, Methods of conjugate gradients for solving linear systems: Nat. Bur. Standards J. Res., 49, 403-436.

"Kaczmarz method". Wikipedia The Free Encyclopedia. http://en.wikipedia.org/wiki/Kaczmarz _method (22 Jan 2009)

Kuehl, H., and Sacchi, M. D., 2001, Split-step WKBJ least-squares migration/inversion of incomplete data: 5th Soc. Expl. Geophys. of Japan Internat. Symp. Imaging Technology.

McCormick, S. F., 1987, Multigrid Methods, SIAM.

Millar, J., and Bancroft, J. C., 2004, Multigrid Deconvolution of seismic data, CSEG Conference Abstract.

Nemeth, T., Wu, C., and Schuster, G. T., 1999, Least-squares migration of incomplete reflection data: Geophysics, 64, no. 1, 208-221.

Plessix, R. E., 2007, A Helmholtz iterative solver for 3D seismic-imaging problems: Geophysics, 72, No. 5, P.SM185-SM194.

Porsani, M., 1999, Seismic trace interpolation using half-step prediction filters. : Geophysics, 64, no. 5, 1461-1467.

Saad,Y., 2000, Iterative methods for sparse linear systems, second edition.

Sacchi, M. D., 2005, Inverse problems in exploration seismology: Short Course note, MITACS 2005, Department of Physics, University of Alberta.

Schneider, W. A., 1978, Integral formulation for migration in two-dimensions and three-dimensions: Geophysics, 43, 49-76.

Shewchuk, J. R., 1994, An Introduction to the Conjugate Gradient Method Without the Agonizing Pain: School of Computer Science, Carnegie Mellon University.

Spitz, S., 1991, Seismic trace interpolation in the f – x domain: Geophysics, 56, no. 6, 785-796.

Stolt, R. H., 1978, Migration by Fourier transform: Geophysics, 43, 23-48.

Strang, G., 1986, Introduction to applied mathematics: Wellesley-Cambridge Press.

Stuben, K. 2002, Multigrid Tutorial, Institute for Pure & Applied Mathematics. http://www.ipam.ucla.edu/publications/es2002/es2002_1449.pdf (20/08/2009).

Tarantola, A., 1984, Inversion of seismic reflection data in the acoustic approximation: Geophysics, 49, no. 8, 1259-1266.

Yousefzadeh, A. 2008, Seismic Data Reconstruction with Regularized Least-Squares Prestack Kirchhoff Time Migration, MSc. thesis, University of Alberta.