Writing Kirchhoff migration/modelling in a matrix form

Abdolnaser Yousefzadeh and John C. Bancroft

ABSTRACT

Kirchhoff prestack migration and modelling are linear operators. Therefore, it is possible to apply them to the seismic data as a matrix-vector multiplication instead of using an operator.

This paper presents a visual method of construction modelling, $\boldsymbol{G}$, and migration, $\boldsymbol{G^T}$, operators in an explicit (matrix) form without going to the mathematical background of Kirchhoff migration or Green functions.

The advantages and disadvantages of using the explicit form of migration operator instead of the implicit forms are discussed.

INTRODUCTION

The seismic exploration method is used for the exploration, evaluation and characterization of the deep hydrocarbon reservoirs. The sampled seismic wavefield must be processed to be interpretable for geophysicists and geologists. In the seismic processing, noises and unwanted signals must be attenuated. Then after normal move out (NMO) correction, all traces corresponding to the same common midpoint (CMP) position are stacked to give an image of the earth subsurface.

There are many other steps in the seismic data processing to give the best possible image of the subsurface. Proper migration of seismic data is a major step. In the CMP design of seismic reflection method, all reflectors are assumed to be horizontal and horizontally homogenous. This may not be true in the real earth subsurface. In the simplest case, ignoring this fact leads to an image in which the dipping reflectors are not correct positioned and also have wrong dip angles. Another problem is the presence of unwanted diffracted energies resulted from underground scatterpoints such as faults edges.

Migration of seismic data not only moves the dipping events to the correct spatial position with the true dip angels, but also collapses the diffracted energies to the scatterpoints. Comparing to the other steps in seismic processing, migration is a complex and expensive procedure. Different methods of migration of seismic data are available. Kirchhoff (Hagedoorn, 1954, Schneider, 1978), Stolt (Stolt, 1978), Gazdag (Gazdag, 1978), Gaussian Beam (Hill, 1990), and Revere Time (Baysal, et. al., 1983) migrations are common methods.

Kirchhoff prestack migration is one of the most frequently used methods in the oil industry. It has many advantages over the other methods such as easy to program, implement, low in cost, working with non constant background velocity, handling converted waves, giving desired (time or depth) output, ability to work on some part of the image, and working with irregular or incomplete data. It can be implemented on the stacked seismic section or prestack data, in both cases the result is an image which ideally shows the earth's reflectivity model as a function of time or depth. The Kirchhoff migration operator works by moving the diffracted energies to the scatterpoint for all possible image points.

The idea of using Kirchhoff modeling/migration operators in the explicit forms is explained. Readers need a little background knowledge on Kirchhoff migration and MATLAB programming language to follow the procedure mentioned in this paper easier. In order to keep this paper short, many aspects of Kirchhoff migration and modelling are ignored. For more information readers are being referred to Yousefzadeh (2008), Claerbout (2005), and Yousefzadeh and Bancroft (2010).

PRESTACK KIRCHHOFF TIME MODELLING, MIGRATION, AND INVERSION

For programming and performing, Kirchhoff is one of the simplest methods of migration. Considering each subsurface point as a diffraction point, Kirchhoff migration collapses all diffracted energies to the scatterpoints. The result is an image of reflectivity with true dip for all events. Kirchhoff prestack modeling, on the other hand, sprays the energy from the scatterpoint to the seismic data. In this paper, the mathematical theory of Kirchhoff modeling and migration is skipped and only some practical issues related to construction of the modeling and migration operators are discussed. Kirchhoff prestack time migration uses Double Squares Root (DSR) equation in order to calculate the migration time for each sample in each trace. DSR equation is:

$$t_m = \sqrt{\frac{t_0^2}{4} + \frac{(x+h)^2}{V_{rms}^2}} + \sqrt{\frac{t_0^2}{4} + \frac{(x-h)^2}{V_{rms}^2}}, \qquad (1)$$

where h is half-offset, the horizontal distance between source and receiver, x is the horizontal distance between the midpoint and the scatterpoint, V_{rms} is root mean square (rms) velocity, and t_0 and t_m are zero offset and migration travel times, respectively. Kirchhoff modelling uses DSR equation to produce seismic data for the given reflectivity model. Proper weighting functions, filtering, interpolation, and dip and aliasing controls are also necessary considerations in the Kirchhoff migration.

Kirchhoff modelling and migration are linear operators. They can be expressed in a matrix-vector multiplication scheme. Let the earth reflectivity model be defined by $\boldsymbol{m}$, and the Kirchhoff modelling operator by $\boldsymbol{G}$. The seismic data, $\boldsymbol{d}$, is given by

$$\boldsymbol{d} = \boldsymbol{Gm}. \qquad (2)$$

The inverse of this calculation should return the reflectivity back,

$$\boldsymbol{m} = \boldsymbol{G}^{-1}\boldsymbol{d}\,. \tag{3}$$

However, calculation of $\boldsymbol{G}^{-1}$, is not always a possible or easy task. First of all, $\boldsymbol{G}$ may not be a square matrix and invertable. Other problems in calculating $\boldsymbol{G}^{-1}$ are the large size of $\boldsymbol{G}$ matrix and the data are not an ideally representation of the model.

The transpose (adjoint) of $\boldsymbol{G}$ matrix is $\boldsymbol{G}^{T}$, and is easier to calculate than the exact inverse. The transpose is less sensitive to the imperfection of data; therefore, it does a better job than the inverse (Claerbout, 2005). All migration methods are essentially transpose process. In the case of Kirchhoff modelling, the transpose process is prestack Kirchhoff migration which is defined by

$$\widehat{\boldsymbol{m}} = \boldsymbol{G}^{T}\boldsymbol{d}, \tag{4}$$

where $\widehat{\boldsymbol{m}}$ is the migrated image and $\boldsymbol{G}^{\mathbf{T}}$ is the migration operator.

By defining Kirchhoff modelling (equation 2) as a forward operator and Kirchhoff migration (equation 4) as its transpose, seismic imaging can be considered as an inversion problem. Substitution of $\boldsymbol{d}$ from equation 2 into equation 4 results in

$$\widehat{\boldsymbol{m}} = \boldsymbol{G}^{T}\boldsymbol{G}\boldsymbol{m}. \tag{5}$$

In least squares migration, $\boldsymbol{m}$ can be estimated by minimizing the difference between the observed data $\boldsymbol{d}$, and the modeled data $\boldsymbol{G}\widetilde{\boldsymbol{m}}$, expressed by $|\boldsymbol{G}\widetilde{\boldsymbol{m}} - \boldsymbol{d}|$, where $\widetilde{\boldsymbol{m}}$ is an approximation to $\boldsymbol{m}$. Finding $\boldsymbol{m}$ is feasible by solving the following normal equation:

$$\boldsymbol{G}^{T}\boldsymbol{G}\boldsymbol{m} - \boldsymbol{G}^{T}\boldsymbol{d} = \boldsymbol{0}. \tag{6}$$

This gives the least squares minimum norm solution, $\boldsymbol{m}_{\boldsymbol{LS}}$:

$$\boldsymbol{m}_{\boldsymbol{LS}} = (\boldsymbol{G}^{\mathbf{T}}\boldsymbol{G})^{-\mathbf{1}}\boldsymbol{G}^{\mathbf{T}}\boldsymbol{d}. \tag{7}$$

In this paper we show how to construct prestack Kirchhoff modelling, migration, and inversion matrix.

CONSTRUCTION OF $\boldsymbol{G}$ IN AN EXPLICIT FORM

In order to create synthetic seismic data with a Kirchhoff modelling operator, we need to have a form of $\boldsymbol{G}$, implicit or explicit. In the implicit form, modelling is a computer program that acts on the reflectivity model as input to produce synthetic data in the output. This program, ignoring some details such as filtering, aperture, antialiasing, and interpolation, diffracts energies from apex to the flanks of all possible hyperbolas. In the explicit form, an algorithm is used to construct $\boldsymbol{G}$ in a matrix form, and then the matrix is multiplied directly to the reflectivity model, $\boldsymbol{m}$, to produce data, $\boldsymbol{d}$. Modelling matrix may be saved in the memory of computer and then multiplied to the model several times it is necessary. Table 1 shows a simplified Kirchhoff prestack time modelling pseudo code written in MATLAB.

Table 1. A simple Kirchhoff prestack time modelling subroutine showing how to create data from given reflectivity.

```
Function  Kichhoff_forward_(D,M, other parameters)
% This is a code to do Kirchhoff prestack time modelling.
% D: Data
% M: Model
% initializing the output vectors:
   D(:)=0;
% loops start here
for ix  = 1:Nx                 % loop over x, CMPs
   for itr = 1:Ntrac           %loop over traces.
      for iz = 1:Nz            % loop over z
         time = DSR;          %calculation of migration time by DSR equation
         it   = floor(time/dt); % it is the sample #
         if (it <= nt)
            Wg= ...            %some migration weight function
               D(it,itr)=D(it,itr)+M(iz,ix)*Wg;
         end
      end
   end
end
```

A simple explanation about our technique for the construction of $\boldsymbol{G}$ matrix, assume a simple 2D earth model with one reflection/scatterpoint at the surface with the amplitude of 1. Let's consider a seismic source right on the top of this scatterpoint and eight receivers in both sides of the source, as shown in Figure 1.

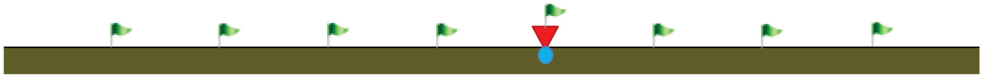


FIG. 1. A 2D model including one source (red triangle) on the top of a scatterpoint (blue circle) and eight receivers (flags) on the earth surface.

The assumed model includes only one scatterpoint with the amplitude of 1 at the surface in the position of row #1 and column # 5. We may show this model by a 1 by 1 matrix as:

$$\tilde{\boldsymbol{m}}_{1,5} = [1], \tag{8}$$

or as a part of a larger matrix such as:

$$\boldsymbol{m} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \qquad (9)$$

We expect that data resulted from this geometry have eight traces and six sample per trace with an assumed sampling rate. Ignoring any amplitude change during wave propagation, data will have the following form:

$$\boldsymbol{d} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \qquad (10)$$

This is equal to spraying the amplitude of the scatterpoint to the flanks of a hyperbola. If the modelling operator has the same shape as the data, $\boldsymbol{d}$, by multiplying the modelling operator to $\boldsymbol{m}$ we can create data. Therefore, the modelling operator for a scatterpoint in row #1 and column # 5 of the model space, $\widetilde{\boldsymbol{G}}_{1,5}$, has the same shape as the data:

$$\widetilde{\boldsymbol{G}}_{1,5} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \qquad (11)$$

Multiplication of the modelling operator with the model should be mathematically possible and creates the data,

$$\widetilde{\boldsymbol{G}}_{1,5}\, \widetilde{\boldsymbol{m}}_{1,5} = \boldsymbol{d}. \qquad (12)$$

However, $\widetilde{\boldsymbol{G}}_{1,5}$ and $\boldsymbol{d}$ are matrices with Nt by $Ntrc$ size, where Nt is the number of samples per trace and $Ntrc$ is the number of traces, and $\widetilde{\boldsymbol{m}}_{1,5}$ is a 1 by 1 matrix.

Multiplication in equation 12, becomes possible by vectorizing $\widetilde{\boldsymbol{G}}_{1,5}$ and $\boldsymbol{d}$ matrices. In order to have $\widetilde{\boldsymbol{G}}_{1,5}$ (or $\boldsymbol{d}$) as a vector, $\widetilde{\boldsymbol{G}}_{\boldsymbol{v}1,5}$ (or $\boldsymbol{d}_{\boldsymbol{v}}$), we put column $i, (\ i > 1)$ of $\widetilde{\boldsymbol{G}}_{1,5}$ (or $\boldsymbol{d}$) matrix below column $i-1$ of that matrix for all columns,

$$\widetilde{\boldsymbol{G}}_{\boldsymbol{v}1,5} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ \vdots \\ 1 \\ 0 \\ 0 \end{bmatrix}, \tag{13}$$

$$\boldsymbol{d}_{\boldsymbol{v}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ \vdots \\ 1 \\ 0 \\ 0 \end{bmatrix}. \tag{14}$$

Now $\widetilde{\boldsymbol{G}}_{\boldsymbol{v}1,5}$ and $\boldsymbol{d}_{\boldsymbol{v}}$ are vectors with $Nt\ \times Ntrc$ elements and we can have equation 12 as the following matrix-vector multiplication:

$$\widetilde{\boldsymbol{G}}_{\boldsymbol{v}1,5}\ \widetilde{\boldsymbol{m}}_{1,5} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ \vdots \\ 1 \\ 0 \\ 0 \end{bmatrix} [1] = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ \vdots \\ 1 \\ 0 \\ 0 \end{bmatrix} = \boldsymbol{d}_{\boldsymbol{v}}. \tag{15}$$

Now suppose that our model which is a row vector with Nx elements, includes second scatterpoint right below the first receiver as shown in Figure 2.

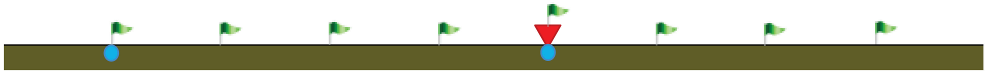


FIG. 2. Model includes one source (red triangle) and eight receivers (flags), and two scatterpoints (blue circles) on the earth surface.

With the mentioned method of vectorization, the reflectivity for the second model with both scatterpoints look likes,

$$\widetilde{\boldsymbol{m}}_{v1,:} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{16}$$

where $\widetilde{\boldsymbol{m}}_{v1,:}$ stands for the first row and all columns of the reflectivity model, in the vectorized form. Ignoring the scatterpoint which is below the source, seismic data modeled from only second scatterpoint look likes this:

$$\boldsymbol{d} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{17}$$

Consequently, the modelling operator for the second scatterpoint in the row # 1 and column # 1, $\widetilde{\boldsymbol{G}}_{1,1}$, will be:

$$\widetilde{\boldsymbol{G}}_{1,1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{18}$$

or as the vectorized form,

$$\widetilde{\boldsymbol{G}}_{\boldsymbol{v}1,1} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{19}$$

The data resulted from presence of both scatterpoints is the summation of data in Equations 8 with the data in equation 16:

$$\boldsymbol{d} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{20}$$

To do operator-matrix multiplication as we did for the case of one scatterpoint we should be able to do the following multiplication:

$$\widetilde{\boldsymbol{G}}_{1,:}\, \widetilde{\boldsymbol{m}}_{\boldsymbol{v}1,:} = \boldsymbol{d}_{\boldsymbol{v}}, \tag{21}$$

where $\widetilde{\boldsymbol{G}}_{1,:}$ is the corresponding modelling operator for all scatterpoints in the first row of the model. This can be easily done if we define $\widetilde{\boldsymbol{G}}_{1,:}$ as

$$\widetilde{\boldsymbol{G}}_{1,:} = [\widetilde{\boldsymbol{G}}_{\boldsymbol{v}1,1}\, \widetilde{\boldsymbol{G}}_{\boldsymbol{v}1,2} \quad \cdots \quad \widetilde{\boldsymbol{G}}_{\boldsymbol{v}1,5} \quad ...]. \tag{22}$$

Then multiplication of $\widetilde{\boldsymbol{G}}_{1,:}$ by $\widetilde{\boldsymbol{m}}_{\boldsymbol{v}1,:}$ gives the resulted data from modelling of all points in the first row of the reflection matrix:

$$\widetilde{\boldsymbol{G}}_{1,:}\,\widetilde{\boldsymbol{m}}_{\boldsymbol{v}1,:}=\begin{bmatrix}0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\1&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&1&0&0&0\\\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots\\0&0&0&0&1&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\end{bmatrix}\begin{bmatrix}1\\0\\0\\0\\1\\0\\0\\0\end{bmatrix}=\begin{bmatrix}0\\0\\0\\1\\0\\1\\\vdots\\1\\0\\0\end{bmatrix}=\boldsymbol{d}_{\boldsymbol{v}}. \tag{23}$$

This is the result of the seismic experiment for the first row of model. Result of modelling of other rows in the image domain must be added to the current data as well. The mentioned procedure can be extended for the other rows of model by defining the modelling diffraction matrix for all elements in each row, and placing all next to each other,

$$\boldsymbol{G}\;=[\widetilde{\boldsymbol{G}}_{\boldsymbol{v}1,1}\;\widetilde{\boldsymbol{G}}_{\boldsymbol{v1},2}\quad\ldots\quad\widetilde{\boldsymbol{G}}_{\boldsymbol{v}1,\mathrm{Nx}}\;\widetilde{\boldsymbol{G}}_{\boldsymbol{v}2,1}\;\widetilde{\boldsymbol{G}}_{\boldsymbol{v}2,2}\ldots\;\widetilde{\boldsymbol{G}}_{\boldsymbol{v}\mathrm{Nz},\mathrm{Nx}}\;], \tag{24}$$

where Nx and Nz are the numbers of columns and rows in the model.

Model needs to be vectorized by the same method,

$$\boldsymbol{m}_{\boldsymbol{v}}=\begin{bmatrix}\widetilde{\boldsymbol{m}}_{\boldsymbol{v}1,:}\\\widetilde{\boldsymbol{m}}_{\boldsymbol{v}2,:}\\\vdots\\\widetilde{\boldsymbol{m}}_{\boldsymbol{v}Nz,:}\end{bmatrix}. \tag{25}$$

Each column of the $\boldsymbol{G}$ represents a point in the migrated image. Therefore, matrix $\boldsymbol{G}$ includes $Nt\times Ntrc$ rows and $Nx\;\times Nz$ columns:

$$\boldsymbol{G}=\begin{array}{c}Nt\\\times\\Ntrc\end{array}\underbrace{\begin{bmatrix}&&&&\cdots&&&&\\\vdots&0&0&0&0&1&\ldots&0&0&0\\&&&&\cdots&&&&\end{bmatrix}}_{Nx\;\times\;Nz}. \tag{26}$$

The result of multiplication of $\boldsymbol{G}$, a $Nt\;\times Ntrc$ by $Nx\;\times Nz$ matrix, to the $\boldsymbol{m}_{\boldsymbol{v}}$, a $Nx\;\times Nz$ vector is $\boldsymbol{d}_{\boldsymbol{v}}$**,** a $Nt\;\times Ntrc$ vector:

$$\boldsymbol{G}^{\boldsymbol{T}}\boldsymbol{m}_{\boldsymbol{v}} = \boldsymbol{d}_{\boldsymbol{v}} \;\Rightarrow\; \begin{matrix}Nt\\ \times\\ Ntrc\end{matrix}\left\{\underbrace{\begin{bmatrix}\cdots&\cdots&\cdots&\cdots\\ \cdots&\cdots&\cdots&\cdots\\ \cdots&\cdots&\cdots&\cdots\end{bmatrix}}_{Nx\ \times\ Nz}\right.\quad \begin{matrix}Nx\\ \times\\ Nz\end{matrix}\left\{\begin{bmatrix}\cdots\\ \cdots\\ \cdots\\ \cdots\end{bmatrix}\right. = \begin{matrix}Nt\\ \times\\ Ntrc\end{matrix}\left\{\begin{bmatrix}\cdots\\ \cdots\\ \cdots\end{bmatrix}\right.. \tag{27}$$

The resulted $\boldsymbol{d}_{\boldsymbol{v}}$ vector must be reshaped to show the data in the conventional form. In MATLAB software, vectorization is done by "$d = d(:);$" command and undone by using the "reshape" function. In order to call a sample in the vectors $\boldsymbol{d}_{\boldsymbol{v}}$ and $\boldsymbol{m}_{\boldsymbol{v}}$, it is useful to know that the sample number $(j-1) \times N_i + i$ in a vector is equal to sample (i, j) in the corresponding matrix form, where N_i is the number of rows.

Calculation of $\boldsymbol{G}$ we must include modelling weight (instead of having just ones) and other parameters. Weight can be added to $\boldsymbol{G}$ matrix by replacement of ones with proper amplitude weights. As seen in Table 1 weight are applied to the data in the most internal loop. Consequently, $\boldsymbol{G}$ can be calculated only by going to the most internal loop and replacing $\boldsymbol{G}$ elements by the calculated migration weight as seen in Table 2.

Table 2. Calculation of Kirchhoff prestack time modelling matrix.

```
Function  Kichhoff_modelling(G, other parameters)
% This is a code to calculate "G", the Kirchhoff prestack time modelling matrix.
% initializing the output vectors:
G=zeros(Nt*Ntrace , Nx*Nz);
% loops start here
for ix  = 1:Nx                    % loop over x, CMPs
   for itr = 1:Ntrac              %loop over traces.
      for iz = 1:Nz               % loop over z
         time = DSR;              %calculation of migration time by DSR equation
         it   = floor(time/dt); % it is the sample #
         if (it <= nt)
            Wg= ...               %some migration weight function
           G((itr-1)*nt+it , (ix-1)*Nz+iz)=Wg;
          end
      end
   end
end
```

CONSTRUCTION OF $\boldsymbol{G}^{\boldsymbol{T}}$ IN AN EXPLICIT FORM

Once modelling matrix, $\boldsymbol{G}$, is constructed, its transpose will be migration matrix, $\boldsymbol{G}^{\boldsymbol{T}}$. However, because the construction of migration matrix is easier to understand than the construction of modeling matrix, it is explained here as well.

Creating $\boldsymbol{G}^{\boldsymbol{T}}$, in the matrix form follow the same roles that we used to create $\boldsymbol{G}$, but in a reverse order. Assume one shot gather with one hyperbola in eight traces, each with six samples as:

$$\boldsymbol{d} = \begin{bmatrix} 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{28}$$

In order to do Kirchhoff migration, all samples on this hyperbola (2s) must be added and result ($8 \times 2 = 16$) put on the apex of the hyperbola, in the row # 1 and column # 5 of the image domain. In this case $\widetilde{\boldsymbol{G}}^{\boldsymbol{T}}_{1,5}$ considered a transposed version of $\boldsymbol{d}$, but with ones instead of the nonzero elements as seen in equation 29:

$$\widetilde{\boldsymbol{G}}^{\boldsymbol{T}}_{1,5} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \tag{29}$$

Index 1,5 in $\widetilde{\boldsymbol{G}}^{\boldsymbol{T}}_{1,5}$ means that we are looking for migration operator for the scatterpoint at row # 1 and column # 5 of our model. Then, multiplication of $\widetilde{\boldsymbol{G}}^{\boldsymbol{T}}_{1,5}$ by $\boldsymbol{d}$ gives the matrix:

$$\widetilde{\boldsymbol{G}}^{\boldsymbol{T}}_{1,5}\, \boldsymbol{d_v} = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}. \tag{30}$$

Summation of the diagonal elements of $\widetilde{\boldsymbol{G}}^{\boldsymbol{T}}_{1,5}\, \boldsymbol{d}$ matrix is equal to the number that must go to the row # 1 and column # 5 in the migration image:

$$\widehat{\widehat{\boldsymbol{m}}}_{1,5} = \sum[diag(\widetilde{\boldsymbol{G}}^{\boldsymbol{T}}_{1,5}\, \boldsymbol{d_v})] = 16. \tag{31}$$

This procedure is valid only for one hyperbola in data and one point in the image domain, $\widehat{\widehat{\boldsymbol{m}}}_{1,5}$. In practice we need a matrix to multiply to the whole data to get the desired migration image, as this equation shows:

$$\hat{\boldsymbol{m}}_{1,5} = \boldsymbol{G}_{1,5}^{\boldsymbol{T}}\, \boldsymbol{d}, \tag{32}$$

where $\hat{\boldsymbol{m}}_{1,5}$ is the migration image at row #1 and column # 5. Data is in a matrix shape with $Ntrc$ traces, each trace in a column, and Nt samples per column. To do matrix multiplication in equation 32, $\boldsymbol{d}$ must be vectorized to $\boldsymbol{d_v}$. To have the data as a vector, we put column i, ($i > 1$) of $\boldsymbol{d}$ below column $i - 1$ for all columns:

$$\boldsymbol{d_v} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 2 \\ \vdots \\ 2 \\ 0 \\ 0 \end{bmatrix}. \tag{33}$$

To be able to do multiplication in equation 32, $\widetilde{\boldsymbol{G}}_{1,5}^{\boldsymbol{T}}$, must also be a row vector. If we put all rows of $\widetilde{\boldsymbol{G}}_{1,5}^{\boldsymbol{T}}$ next to each other, we achieve the desired vector:

$$\widetilde{\boldsymbol{G}}_{\boldsymbol{v}1,5}^{\boldsymbol{T}} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad \dots \quad 1 \quad 0 \quad 0]. \tag{34}$$

Now multiplication of $\widetilde{\boldsymbol{G}}_{\boldsymbol{v}1,5}^{\boldsymbol{T}}$ by $\boldsymbol{d_v}$, is equal to the summation along the diagonal elements of $\widetilde{\boldsymbol{G}}_{1,5}^{\boldsymbol{T}}\boldsymbol{d}$ matrix in equation 30. Therefore, instead of calculating the diagonal elements of $\widetilde{\boldsymbol{G}}_{1,5}^{\boldsymbol{T}}\boldsymbol{d}$ matrix, we can do multiplication of $\widetilde{\boldsymbol{G}}_{\boldsymbol{v}1,5}^{\boldsymbol{T}}$ row vector to the $\boldsymbol{d_v}$ column vector. This is equivalent to the summation of amplitudes along the flanks of diffraction hyperbola:

$$\hat{\hat{\boldsymbol{m}}}_{1,5} = \widetilde{\boldsymbol{G}}_{\boldsymbol{v}1,5}^{\boldsymbol{T}}\boldsymbol{d_v} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad \dots \quad 1 \quad 0 \quad 0] \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 2 \\ \vdots \\ 2 \\ 0 \\ 0 \end{bmatrix} = [16]\,. \tag{35}$$

Vector $\boldsymbol{d_v}$ has $Nt \times Ntrc$ elemnents. $\widetilde{\boldsymbol{G}}_{\boldsymbol{v}1,5}^{\boldsymbol{T}}$ is a vector with the same length. The resulted $\hat{\hat{\boldsymbol{m}}}_{1,5}$ is just a number corresponding to one point in the image domain, the hyperbola apex. In reality, there are $Nx \times Nz$ possibilities for the position of the hyperbola apex. Therefore, if $\widetilde{\boldsymbol{G}}_{\boldsymbol{v}i,j}^{\boldsymbol{T}}$ row vector is for migration of one point in the image domain at row # i and column # j, then for each image point we have to have another row vector for $\widetilde{\boldsymbol{G}}^{\boldsymbol{T}}$. This can be done by adding additional rows to $\boldsymbol{G}^{\boldsymbol{T}}$.

If each row of the new $\boldsymbol{G^T}$ represents a point in the migrated image then $\boldsymbol{G^T}$ will include $Nx \times Nz$ rows and $Nt \times Ntrc$:

$$\boldsymbol{G}^{\mathbf{T}} = \begin{array}{c} Nx \\ \times \\ Nz \end{array} \underbrace{\begin{bmatrix} & & & & \cdots & & & & \\ 0 & 0 & 0 & 0 & 1 & \ldots & 0 & 0 & 0 \\ & & & & \cdots & & & & \end{bmatrix}}_{Nt \quad \times \quad Ntrc}. \tag{36}$$

The result of multiplication of $\boldsymbol{G^T}$, a $Nx \times Nz$ by $Nt \times Ntrc$ matrix, to the $\boldsymbol{d_v}$, a $Nt \times Ntrc$ vector is a $Nx \times Nz$ vector:

$$\boldsymbol{G^T d_v} = \boldsymbol{\widehat{m}_v} \Rightarrow \begin{array}{c} Nx \\ \times \\ Nz \end{array} \left\{ \underbrace{\begin{bmatrix} \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix}}_{Nt \quad \times \quad Ntrc} \right. \quad \begin{array}{c} Nt \\ \times \\ Ntrc \end{array} \left\{ \begin{bmatrix} \cdots \\ \cdots \\ \vdots \\ \cdots \\ \cdots \end{bmatrix} \right. = \begin{array}{c} Nx \\ \times \\ Nz \end{array} \left\{ \begin{bmatrix} \cdots \\ \cdots \\ \cdots \end{bmatrix} \right. . \tag{37}$$

This vector must be reshaped as a Nx by Nz matrix to form the migrated image on an interpretable section. In vectorizing $\boldsymbol{d_v}$, we put the seismic traces below eachother by putting the second trace below the first one and the third trace below the second one and so on. If we wanted the same vectorization to be valid for the image, $\boldsymbol{\widehat{m}_v}$, we must consider it in the construction of $\boldsymbol{G^T}$ matrix. Each row in $\boldsymbol{G^T}$matrix represents one sample in the image vector $\boldsymbol{\widehat{m}_v}$. We put all rows corresponded to first column of the image matrix below each other and then repeat this procedure for the next column of the image matrix, therefore, $\boldsymbol{\widehat{m}_v}$ is vectorized in the same manner as $\boldsymbol{d_v}$ is vectorized.

It is important to mention as disused for the modelling operator, in the calculation of $\boldsymbol{G^T}$, migration weight and other migration parameters must be included. Weight can be added to $\boldsymbol{G^T}$ matrix by replacement of ones with proper amplitudes. Migration weights are applied to the data in the most internal loop of migration algorithm. Consequently, $\boldsymbol{G^T}$can be calculated only by going to the most internal loop and replacing the $\boldsymbol{G^T}$elements by the calculated migration weight as seen in Table 3.

Table 3. Calculation of Kirchhoff prestack time migration matrix.

```
Function  Kichhoff_migration(G, other parameters)
% This is a code to calculate Gt, the Kirchhoff prestack time migration matrix.
% initializing the output vectors:
Gt=zeros(Nx*Nz,Nt*Ntrace);
% loops start here
for ix  = 1:Nx                 % loop over x, CMPs
   for itr = 1:Ntrac            %loop over traces.
```

```
        for iz = 1:Nz                     % loop over z
            time = DSR;               %calculation of migration time by DSR equation
            it   = floor(time/dt); % it is the sample #
            if (it <= nt)
                Wg= ...                   %some migration weight function
               Gt((ix-1)*Nz+iz , (itr-1)*nt+it)=Wg;
             end
        end
    end
end
```

However, with interpolation and antialiasing, the calculation of $\boldsymbol{G}$ and $\boldsymbol{G}^{\boldsymbol{T}}$ is more complicated. For example in case of adding linear interpolation, the calculation of $\boldsymbol{G}^{\boldsymbol{T}}$ needs to be done with dividing the weight functions to two different adjacent samples in $\boldsymbol{G}^{\boldsymbol{T}}$as shown in Table 4.

Table 4. Calculation of Kirchhoff prestack time migration/modelling matrix with linear interpolation.

```
function Kichhoff_migration_matrix (Gt, other parameters)
% This is a code to calculate Gt, the Kirchhoff prestack time migration matrix, with linear interpolation.
% initializing the output vectors:
Gt=zeros(Nx*Nz,Nt*Ntrace);
% loops start here
for ix  = 1:Nx                          % loop over x, CMPs
    for itr = 1:Ntrac                   %loop over traces.
        for iz = 1:Nz                     % loop over z
            time = DSR;               %calculation of migration time by DSR equation
            it   = floor(time/dt); % it is the sample #
            if (it <= nt)
                    dtt  = 1-time/dt+it;
                    Wg= ...                   %some migration weight function
                    Gt((itrc-1)*nt+it,(ix-1)*Nz+iz)=Gt((itrc-1)*nt+it,(ix-1)*Nz+iz)+dtt*Wg;
                    Gt((itrc-1)*nt+it+1,(ix-1)*Nz+iz)= Gt((itrc-1)*nt+it+1,(ix-1)*Nz+iz)+(1-dtt)*Wg;
             end
        end
    end
end
```

The technique of time modelling/migration matrix construction mentioned above, can be easily extended to the depth modelling/migration. In Kirchhoff depth migration, instead of using the DSR equation, we use a ray tracing program to calculate the migration time for each sample. Therefore, by replacing the modelling/migration time from DSR equation in Table 1, by time from ray tracing program, we will have a depth migration matrix.

EXPLICIT FORM OF CONVOLUTION AND CROSS CORALATION

In the previous section we showed how to create Kirchhoff modelling, and its transpose, migration operators in the matrix forms. These two, together make the necessary pair for some analyses such as least squares Kirchhoff migration.

However, least squares migration needs convolution and its transpose, cross correlation to be implemented to the data and model as well. In the same manner that we calculated $\boldsymbol{G}$ matrix, but in a more convenient way, we may calculate the convolution matrix, $\boldsymbol{W}$, and multiply that with the data. Convolution matrix is well known to geophysicists and other math-related scientists.

In order to multiply convolution matrix to a data matrix with the size of Nt by $Ntrc$, convolution matrix should have size of $(Nt + Nw - 1)$ by Nt, where Nw is the number of wavelet samples. The size of convolved data will be $(Nt + Nw - 1)$ by $Ntrc$. To construct a convolution matrix, a zero matrix with the size of $(Nt + Nw - 1)$ by Nt is chosen and then elements of wavelet replace the first zeros in the first column. For the second column wavelet replaces the zero elements starting from the second row. This replacement continues for all columns.

Multiplication of these matrices is equal to the convolution of each trace in data with the wavelet.

In the previous example of having $\boldsymbol{d}$ as a 6×8 matrix, if the wavelet vector is

$$\boldsymbol{w} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \tag{38}$$

then, convolution (wavelet) matrix has the following form:

$$\boldsymbol{W} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 & 0 \\ 3 & 2 & 1 & 0 & 0 & 0 \\ 0 & 3 & 2 & 1 & 0 & 0 \\ 0 & 0 & 3 & 2 & 1 & 0 \\ 0 & 0 & 0 & 3 & 2 & 1 \\ 0 & 0 & 0 & 0 & 3 & 2 \\ 0 & 0 & 0 & 0 & 0 & 3 \end{bmatrix}. \tag{39}$$

Multiplication of $\boldsymbol{W}$ and $\boldsymbol{d}$ returns convolved data back:

$$\boldsymbol{Wd} = \begin{bmatrix} 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 4 & 2 & 0 & 0 \\ 0 & 0 & 2 & 4 & 6 & 4 & 2 & 0 \\ 0 & 0 & 4 & 6 & 0 & 6 & 4 & 2 \\ 0 & 2 & 6 & 0 & 0 & 0 & 6 & 4 \\ 2 & 4 & 0 & 0 & 0 & 0 & 0 & 6 \\ 4 & 6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{40}$$

Transpose of convolution matrix, $\boldsymbol{W}$, is cross correlation matrix, $\boldsymbol{W^T}$, which can be extracted from $\boldsymbol{W}$. After vectorizing the $\boldsymbol{W^T}$, it may be applied to the data before migration.

With migration and convolution matrices, Kirchhoff prestack migration may be written in a complete form of:

$$\widehat{\boldsymbol{m}} = \boldsymbol{G^T W^T d}. \tag{41}$$

When considering the wavelet, Kirchhoff modelling will be expressed by:

$$\boldsymbol{d} = \boldsymbol{WGm}. \tag{42}$$

FEASIBILITY OF WORKING WITH THE MATRIX FORM OF $\boldsymbol{G^T}$, $\boldsymbol{G}$, AND $\boldsymbol{G^T G}$

When $\boldsymbol{G}$ and $\boldsymbol{G^T}$ are available in the matrix form, they can be used for least squares prestack Kirchhoff migration. It seems easier to use the matrix form when we need to perform Kirchhoff migration and modelling. It is also more feasible to add some regularization to such kinds of inversion as well.

However, $\boldsymbol{G}$ and $\boldsymbol{G^T}$ are very large matrices. The size of $\boldsymbol{G}$ and $\boldsymbol{G^T}$ is equal to the size of the model multiplied by the size of data. It means for a 10 km length 2D line with 75000 traces in 1000 CMPs and each trace having 1000 samples, size of $\boldsymbol{G}$ will be $Nx \times Nz \times Ntraces \times Nt = 7.5 \times 10^{13}$ samples. If these data are being loaded in memory as double precision numbers, they would need 600 TB of memory which is far from the memory size of any computer. The problem is more severe if 3D data is being used. In addition to the size, matrix $\boldsymbol{G}$ is not sparse enough to save and apply sparse techniques. The inversion matrix, $\boldsymbol{G^T G}$ matrix is even denser than the modelling matrix.

Suppose the assumed geometry includes only one source, 15 receivers, 15 CMPs, 200 pseudo depth samples and our data has 300 samples per trace. This geometry is shown in Figure 3. $\boldsymbol{G}$ will be a 3000 by 4500 matrix. Non-zero elements of this matrix are shown in Figures 4a and 4b. Migration matrix, $\boldsymbol{G^T}$, is a 4500×3000 matrix. Non-zero elements of this matrix are shown in Figures 5a and 5b. Figures 6a and 6b show the inversion matrix with the mentioned geometry.

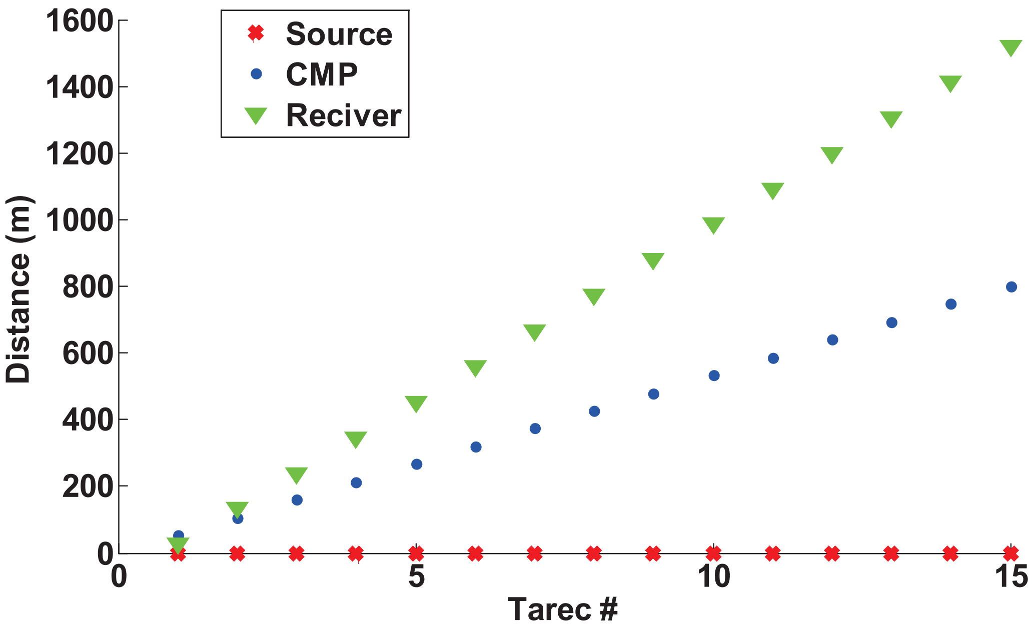
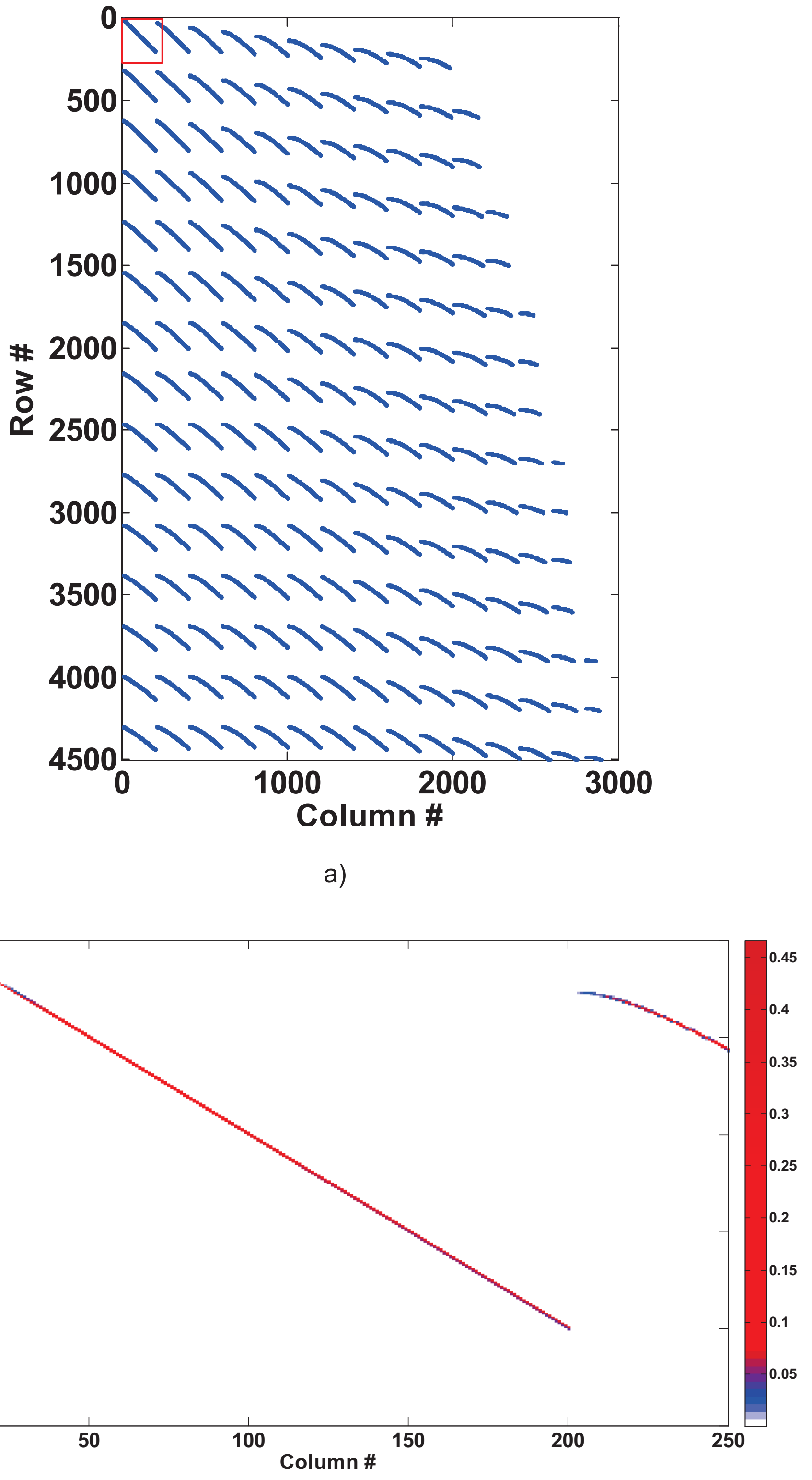


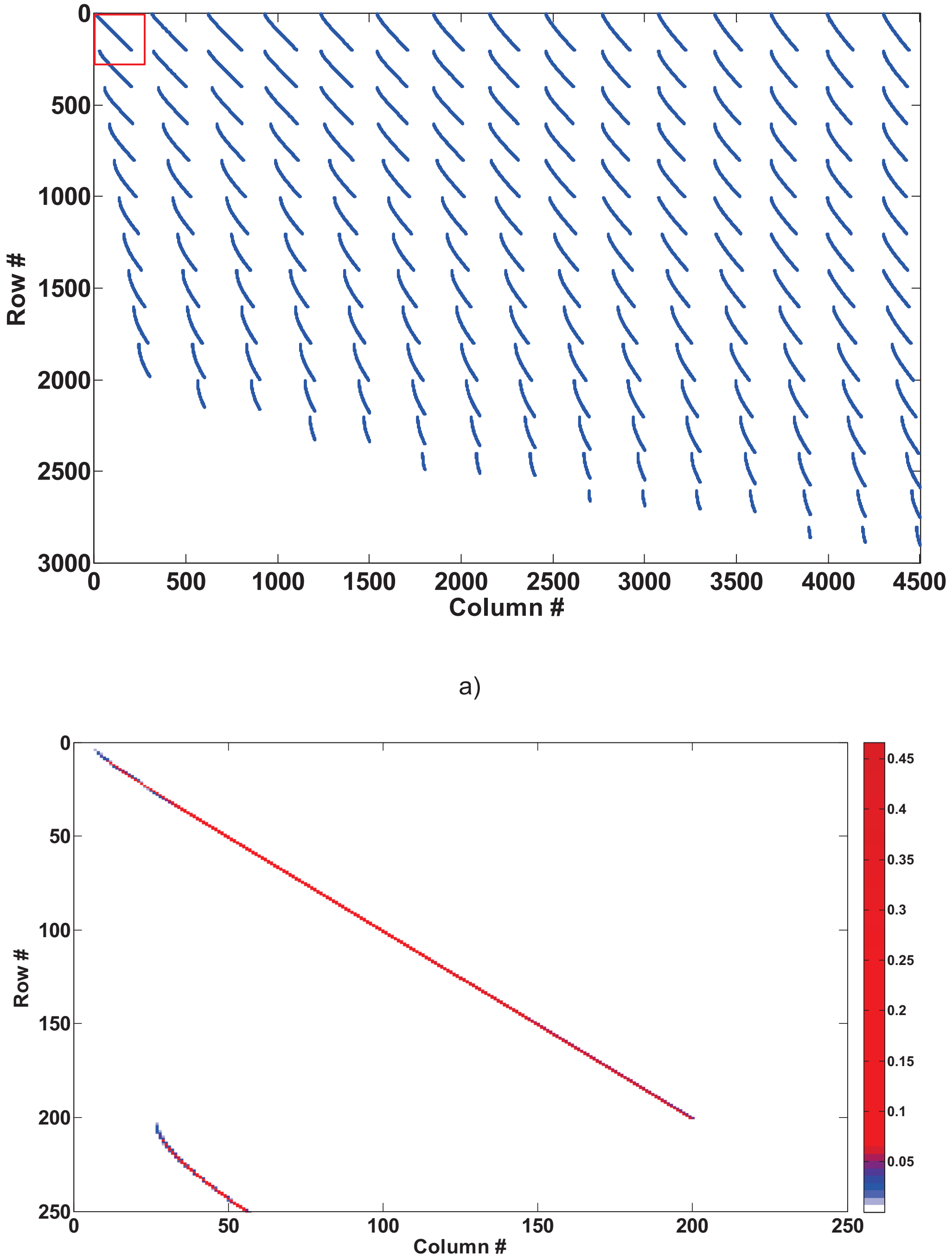
FIG. 3. Geometry of the example mentioned in text.



a)

b)

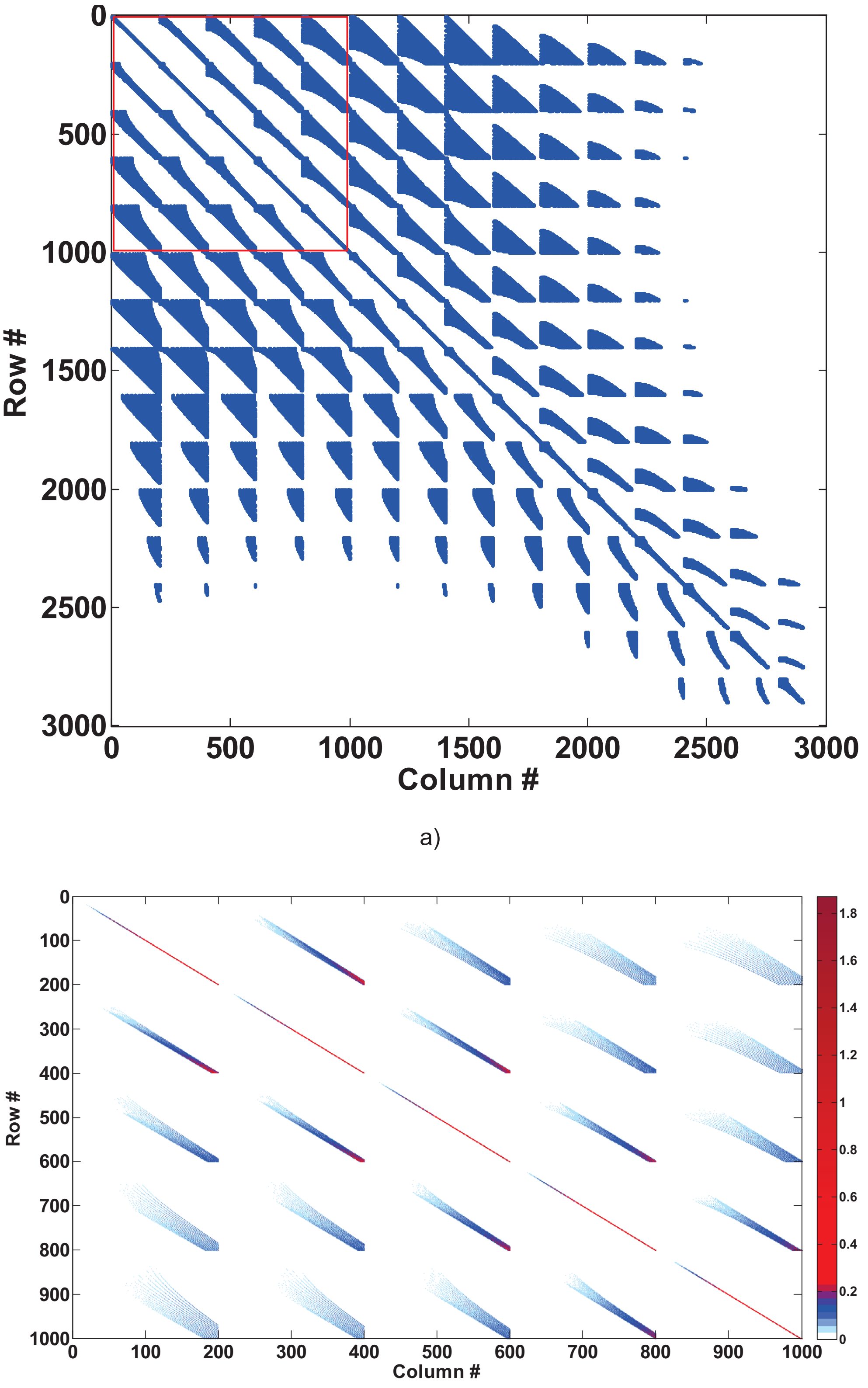
FIG. 4. a) Non-zero elements of matrix $\boldsymbol{G}$, b) Close up of the first 250 rows and columns.



a)

b)

FIG. 5. a) Non-zero elements of matrix $\boldsymbol{G}^{\mathrm{T}}$, b) Close up of the first 250 rows and columns.



a)

b)

FIG. 6. a) Non-zero elements of matrix $\boldsymbol{G}^T\boldsymbol{G}$, b) Close up of the first 1000 rows and columns.

CONCLUSION

A technique for creating Kirchhoff modelling, migration, and inversion matrices is shown without the requirement of going into the mathematics of Kirchhoff migration or Green functions. Construction of the modelling/migration matrices goes to the most internal loop of the Kirchhoff modelling/migration algorithms. Once they are calculated, they can be easily repeated just by multiplication of the matrix to the model or data vector.

However, the migration or modelling matrix is too big and dense which make it impractical for a real dataset.

ACKNOWLEDGMENTS

Authors wish to acknowledge the sponsors of CREWES project for their continuing support. They also appreciate Kevin Hall and Rolf Maier in the CREWES project.

REFERENCES

Baysal, E., Kosloff, D. D., and Sherwood, J. W. C., 1983, Reverse time migration: Geophysics, **48**, 1514-1524.

Claerbout, J. F., 2005, Basic Earth Imaging (Version 2.4): Stanford Exploration Project, http://sepwww.stanford.edu/sep/prof/.

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

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

Hill, N. R., 1990, Gaussian beam migration: Geophysics, **55**, 1416-1428.

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

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

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

Yousefzadeh, A., and Bancroft, J. C., 2010, Solving least squares Kirchhoff migration using multigrid methods: 80rd Annual International Meeting, Society of Exploration Geophysicists, Expanded Abstract, 3135-3139.