Acoustic finite difference parameter analysis and modelling in MATLAB

David Cho, Chad Hogan and Gary F. Margrave

ABSTRACT

The modelling of seismic energy is a valuable tool in seismology. It is useful in the understanding of seismic wave propagation as well as seismic imaging and inversion problems. Modelling will generally involve a tradeoff between cost and accuracy due to certain aspects such as grid dispersion and compute time. Thus a balance must be found between the two and parameters chosen that are best suited for a specific situation. In addition, since most industry applications use SEG-Y for writing seismic data, it is useful to output any modelled results into SEG-Y for a more convenient transition between software applications.

By using MATLAB and the CREWES MATLAB software package, an analysis was performed on various modelled data sets to determine an appropriate sampling for the spatial grid. Different bandwidth filters were also applied to the seismograms to determine their effect on the data. Increasing the sampling of the spatial grid enhances the results but requires a more intensive computation process. In addition, by applying an appropriate filter, erroneous data can be rejected by removing parts of the high frequency numerically dispersive signal.

After parameter analysis, a data set was modelled and loaded into VISTA for processing. The processing results did suggest that the modelled data was successfully written to SEG-Y.

INTRODUCTION

A fundamental problem in seismology is the determination of wave phenomena due to an impulsive source that triggers the particle motion in some arbitrary medium. This forward modelling problem serves as a basis for understanding seismic wave propagation as well as an accessory for the more important seismic imaging and inversion problems. In addition, given that the model used in the wavefield simulations is the solution to the inverse problem, seismic inversion algorithms can be evaluated using the modelled data.

Due to certain aspects that are associated with modelling such as grid dispersion and compute time, the choice of parameters is crucial in generating an effective data set. Wavefield simulations are computed digitally, thus requiring the data to be sampled at discrete intervals. As the number of samples increases, a more accurate solution will be obtained. However, this will result in a lengthy computation process. Thus the user must choose the optimal parameters to achieve a balance between cost and accuracy. In addition, the majority of industry applications use SEG-Y for writing seismic data. Thus it would be useful to output any modelled results into SEG-Y, making data transitions between software a more convenient task. This paper demonstrates the use of MATLAB to perform an analysis of the modelling parameters. Subsequently the seismic response was modelled with an output to SEG-Y.

THEORY

Finite Difference

The two-dimensional scalar wave equation given by

$$\nabla^2\Psi(x,z,t) = \frac{\partial^2\Psi(x,z,t)}{\partial x^2} + \frac{\partial^2\Psi(x,z,t)}{\partial z^2} = \frac{1}{v^2(x,z)}\frac{\partial^2\Psi(x,z,t)}{\partial t^2} \quad (1)$$

governs the behaviour of seismic energy in a constant density, acoustic medium, where Ψ is the pressure field, t is the time, v is the velocity, x and z are the spatial coordinates and ∇^2 is the spatial Laplacian operator. However, the number of exact solutions to the wave equation is very small and an approximate method of solution is usually required to carry out wavefield simulations. Several numerical methods have been developed over the years such as finite difference (e.g. Kelly et al, 1976), finite element (e.g. Marfurt, 1984) as well as pseudospectral methods that are carried out in the Fourier domain (e.g. Kosloff and Baysal, 1982).

An algorithm within the finite difference toolbox of the CREWES MATLAB software package was used to compute the model response of a seismic disturbance. The main function in this modelling package is *afd_snap* which will forward propagate a 2-D wavefield a single step in time for a certain velocity model using a finite difference approach. The equation for time stepping the wavefield is given by

$$\Psi(x,z,t+\Delta t) = [2+\Delta t^2 v^2(x,z)\nabla^2]\Psi(x,z,t) - \Psi(x,z,t-\Delta t)\,, \quad (2)$$

which is the direct result of a second-order, central finite difference approximation to the time derivative in equation (1). *Afd_snap* requires as input the time step size (Δt) and spatial grid size (Δx), a velocity model, and two wavefield snapshots at time t and t-Δt to forward propagate the wavefield to t+Δt. A stability condition must also be met in order to maintain a bounded solution and is given by the inequality

$$\frac{v_{\max}\Delta t}{\Delta x} \le \sqrt{3/8}\,, \quad (3)$$

where $v_{\max}$ is the maximum velocity and Δt and Δx are the time and spatial grid sizes respectively. The user will not usually invoke *afd_snap* directly but rather *afd_shotrec* which will generate a shot record by using *afd_snap* recursively to forward propagate the wavefield to some user defined maximum time and extracting samples from user defined receiver locations. *Afd_shotrec* will then filter the seismograms using parameters defined by the user. The CREWES acoustic finite difference modelling facility is documented in Youzwishen and Margrave (1999).

Grid Dispersion

The group velocity of seismic energy is a wave number dependent quantity. Due to the sampling of the spatial grid, different spectral components will travel at different velocities, resulting in an artificial dispersion known as grid dispersion (e.g. Holberg, 1987). The underlying cause is that the finite difference operators for numerical differentiation become increasingly inaccurate at shorter wavelengths. To remedy this problem, a very fine sampling may be required to reduce the effects of this numerically induced dispersion.

AFD_SHOTREC MODIFICATION

Afd_shotrec only allows for a horizontal surface at the free surface boundary. Thus the existing algorithm cannot account for any effects generated by varying topography. A modified version of *afd_shotrec* called *afd_shotrec_topo* was developed to address this issue. *Afd_shotrec_topo* will require a velocity model that includes a vector specifying the topography relative to some absolute datum, which is taken as the upper edge of the velocity grid. The free surface reflection is now achieved by a velocity contrast at the topographic surface. The new function will require a user defined receiver spacing and will place receivers on the topographic surface accordingly. It will also return a vector with the proper receiver elevations. The wavefield simulations in this paper were carried out using this new function.

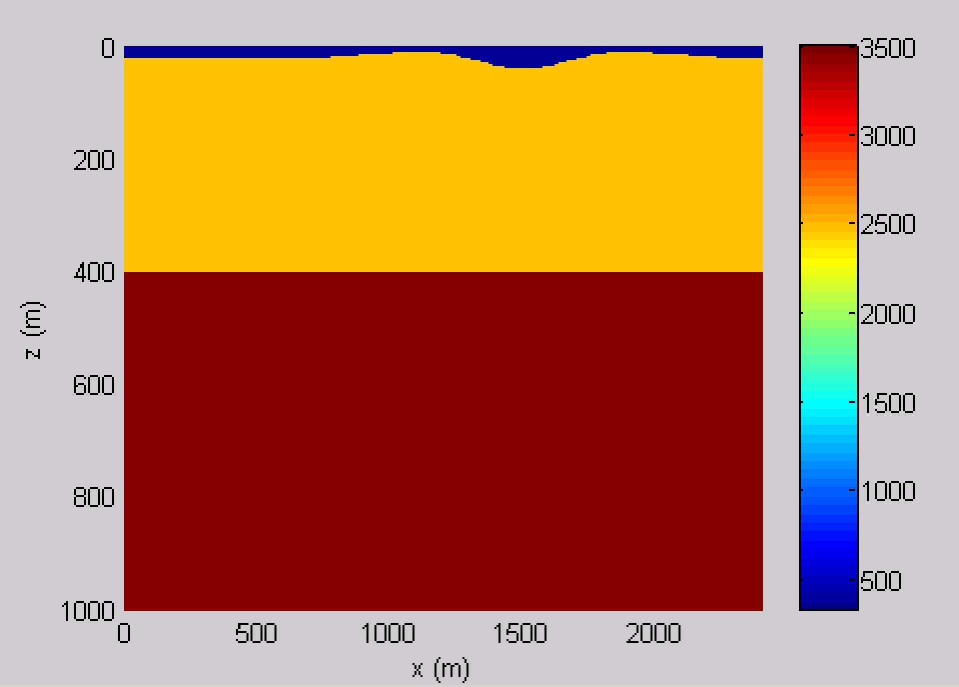

FIG. 1. Two layer velocity model with a varying topographic surface.

To demonstrate the new topography code, a shot record was generated using a simple two layered velocity model with a varying topographic surface as shown in Figure 1. The result of the topography is the introduction of statics due to a non horizontal recording plane. Figure 2 shows the results.

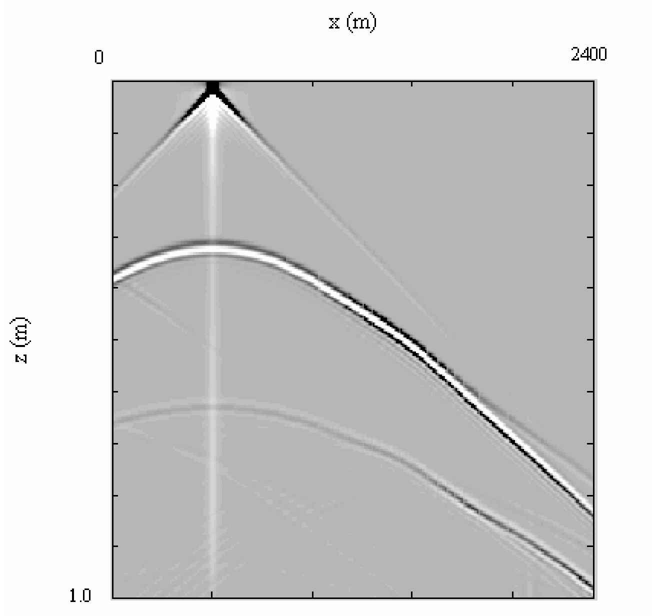

FIG. 2. Shot record with a non horizontal recording plane.

MODELLING PARAMETER ANALYSIS

To examine the effects of various modelling parameters, a velocity model resembling a reef structure was explored. Appendix A shows the MATLAB script used for generating the velocity model. Figure 3 shows the velocity model where a horizontal topographic surface was used.

Using the reef model, a series of shot records were generated with an impulsive source that had variable spatial grid sizes to determine each record’s quality and compute time. A total of ten shot records were generated where the spatial grid sizes were decreased by integer steps ranging from the largest sampling of ten meters to the finest sampling of one meter. A time step size must also be defined but cannot be chosen independently of the spatial step size due to the stability condition given by equation (3). Thus a constant ratio of $\Delta t/\Delta x$ was maintained for the various shot records.

Figure 4 shows unfiltered shot records for a spatial grid sampling of one and ten meters with a shot located on the left edge of the model. The ten meter grid (Figure 4a) exhibits a much more dispersive wavefield where the signal appears to “ring” throughout the shot record. The one meter grid (Figure 4b) produces an enhanced result where the signal is much sharper but required a compute time which was 500 times larger.

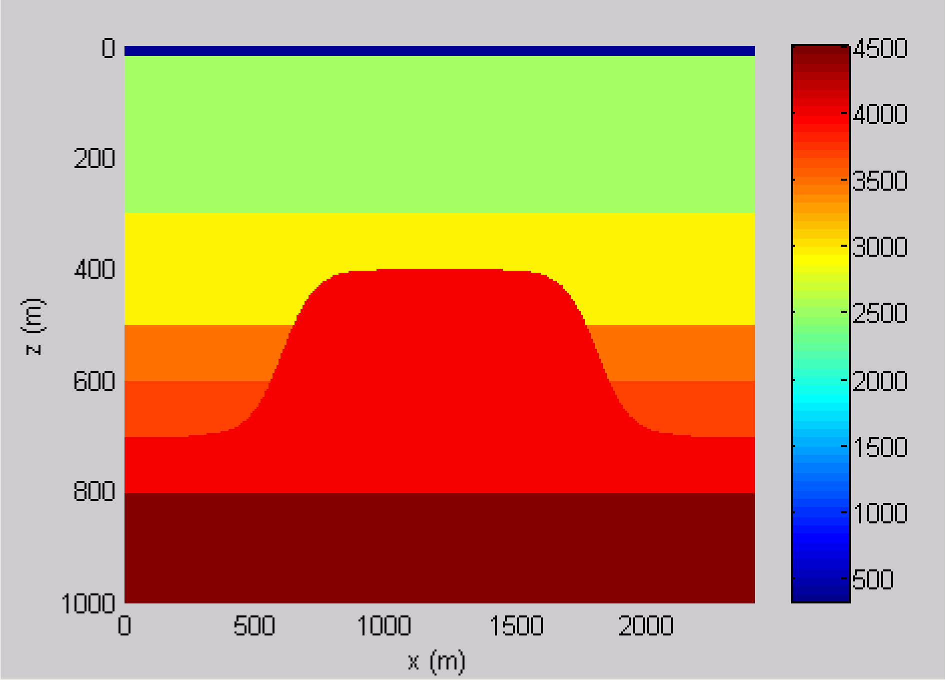

FIG. 3. Reef velocity model used in analysis and modelling.

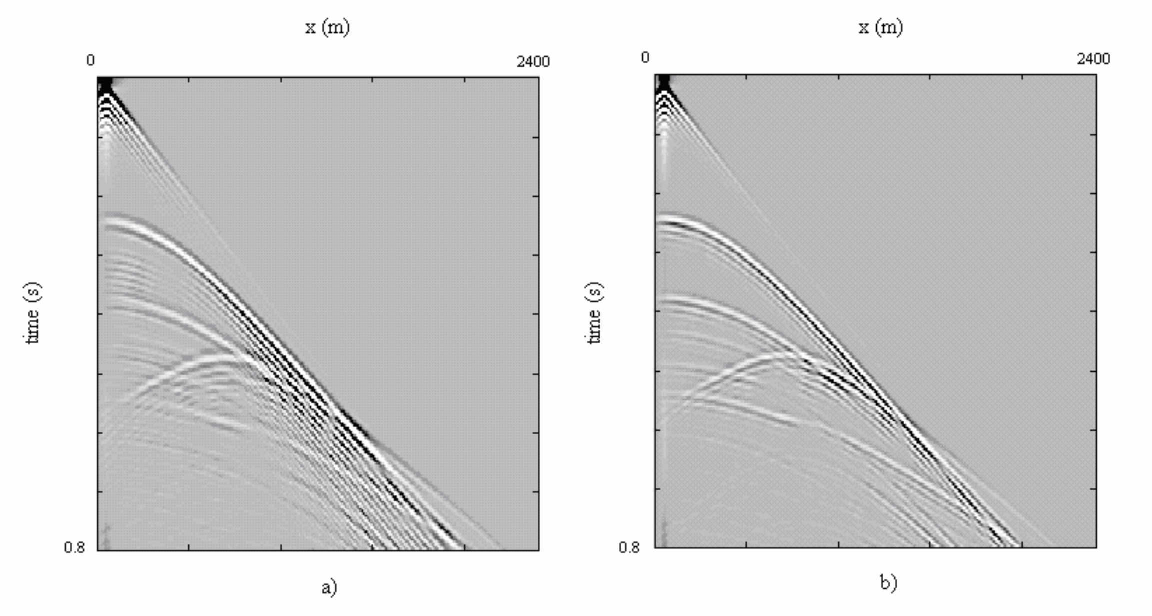

FIG. 4. Shot records for a 10m (a) and 1m (b) spatial sampling grid.

Figure 5 shows the average amplitude spectrum of various shot records. The spectra were scaled relative to a reference to demonstrate their similarities and differences. The lower end of the spectrum is similar but the higher frequency content is inconsistent for varying dx. Given that the numerically dispersive signal is attributed to the high frequencies, a properly chosen filter could remove some of the effects of the artificial grid dispersion. Three Ormsby filters with the band pass of [0 10 30 40], [0 10 40 50] and [0 10 50 60] were applied to the unfiltered shot records to determine the effect on the seismograms. To quantify the quality of the shot records generated with the various grid sizes and filters, a crosscorrelation was performed for each shot record with a pilot shot record taken as the one with the finest grid spacing or highest accuracy (dx=1). A trace equalization was performed for each shot record and the average maximum correlation amplitude of a trace by trace crosscorrelation was taken as a measure of quality relative to the pilot shot record.

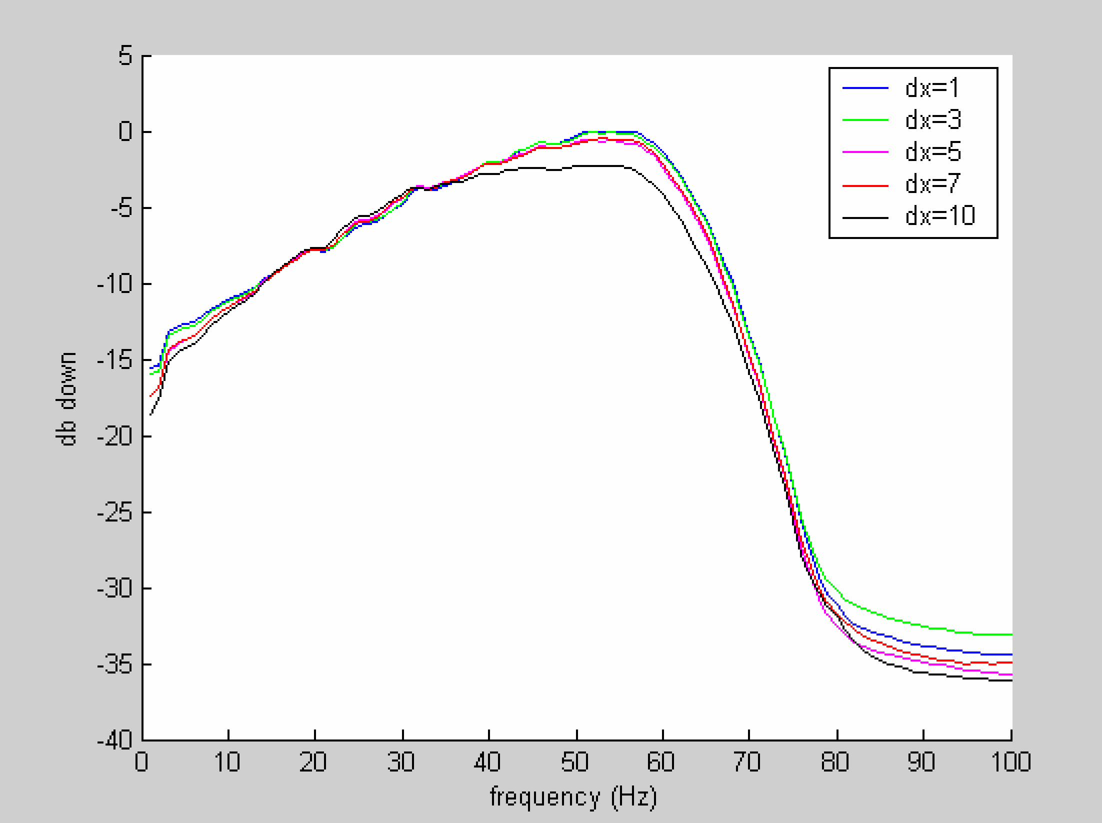

FIG. 5. Average amplitude spectrum of shot records with various grid sizes.

Figure 6 and Table 1 show the crosscorrelation results for the various grid sizes and filters along with the time required for generating each shot record. The correlation curve for the unfiltered shot records exhibits a general increase as dx decreases. The effect of the filters is a flattening of this curve. A narrower bandwidth in the low frequency range will produce a flatter curve, resulting in a higher correlation amplitude with larger dx. However, the removal of the high frequency grid dispersion will be accompanied by the removal of high frequency signal. This effect is undesirable as it limits the ability to resolve small structures, thus when choosing a filter, a larger bandwidth would be preferential.

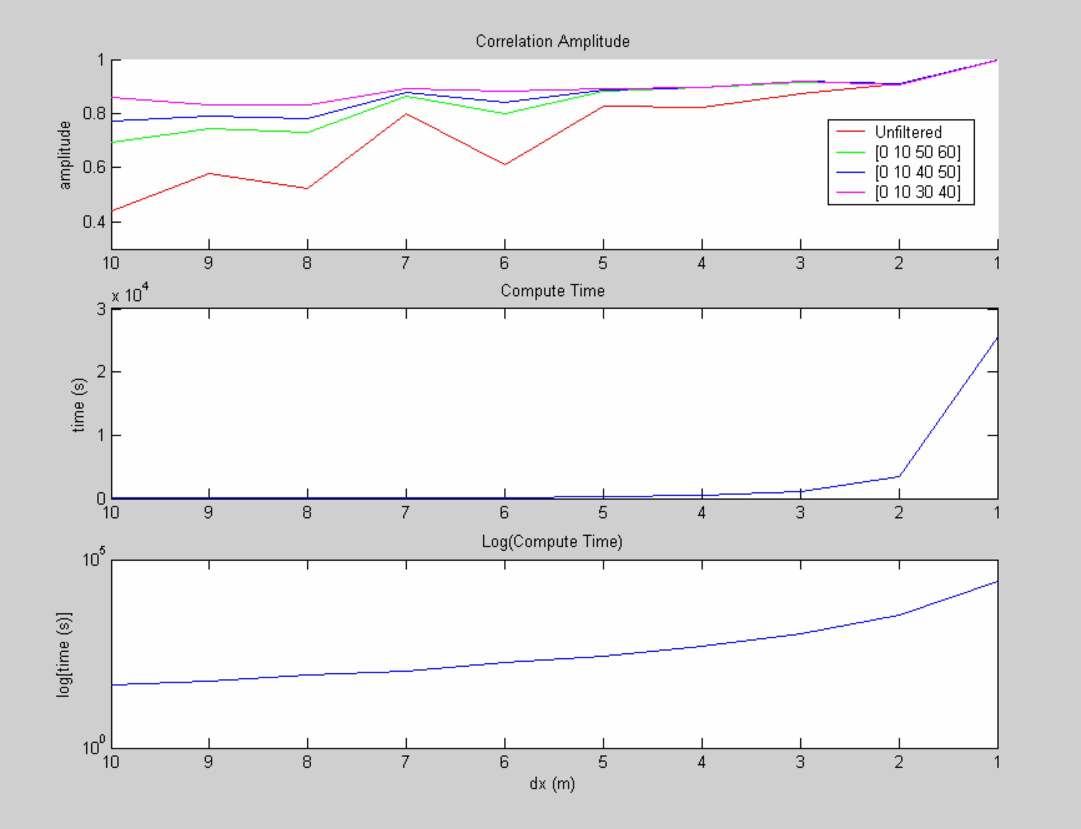

FIG. 6. Correlation amplitude and compute time for various grid sizes and filters.

Table 1. Correlation amplitude and compute time for various grid sizes and filters.

dx (m)	Unfiltered	[0 10 30 40]	[0 10 40 50]	[0 10 50 60]	Compute Time (s)
10	0.439	0.859	0.774	0.692	48.24
9	0.580	0.831	0.790	0.745	59.06
8	0.522	0.832	0.781	0.732	83.87
7	0.800	0.892	0.878	0.864	111.36
6	0.609	0.881	0.843	0.800	183.07
5	0.827	0.893	0.888	0.884	277.35
4	0.825	0.898	0.897	0.894	509.42
3	0.875	0.921	0.918	0.915	1070.90
2	0.912	0.908	0.910	0.911	3422.70
1	1	1	1	1	25541.00

From Figure 6 and Table 1, the user can determine which modelling parameters are best suited for the specific model. Figure 6 shows the filtered correlation curves converging for dx values of five and below. This suggests that the differences in the shot records due to the grid dispersion lie primary outside the bandwidth of the various filters for these values. In addition, the filtered correlation curves begin to flatten for dx values of five and below, suggesting that a further decrease in dx will not improve the data by a significant amount. In this range, the numerical values for the correlation amplitudes show that the smallest and largest differ by 0.03 for dx=5 and dx=3 respectively. This is only a 3% improvement relative to the pilot shot record and came at the expense of a compute time which is four times larger, thus dx=5 and the largest bandwidth filter given by [0 10 50 60] was chosen as the modelling parameters.

MODELLING WITH SEG-Y OUTPUT

Using the parameters chosen above, a survey was performed over the reef model of Figure 3.

The script shown in Appendix B was created in MATLAB to simulate the shots and write the modelled data into SEG-Y. The script will require as input a velocity model, header information and modelling parameters defined by the user. Proper SEG-Y text and binary headers will first be generated and the script will subsequently shoot at various user defined shot locations by implementing *afd_shotrec_topo* recursively. The resulting geometry and trace data will then be written to their respective sections in the SEG-Y file.

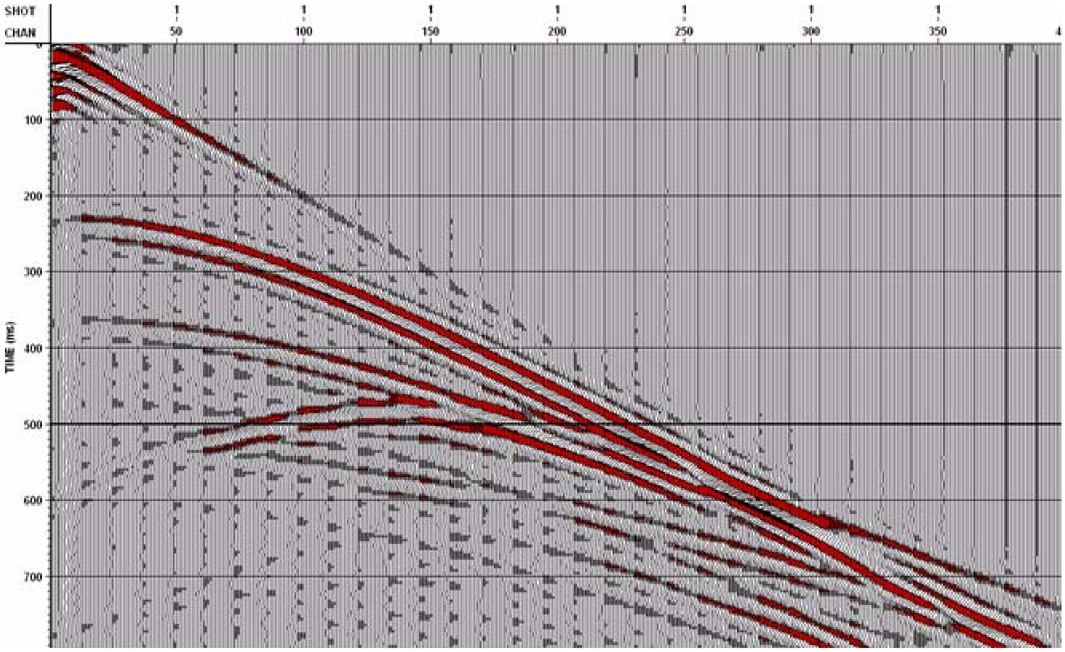

FIG. 7. Modelled shot record displayed in VISTA.

To ensure that the shots were written into SEG-Y properly, the simulated shots were loaded into VISTA and an attempt to process the data was made. Figure 7 shows a shot record displayed in VISTA.

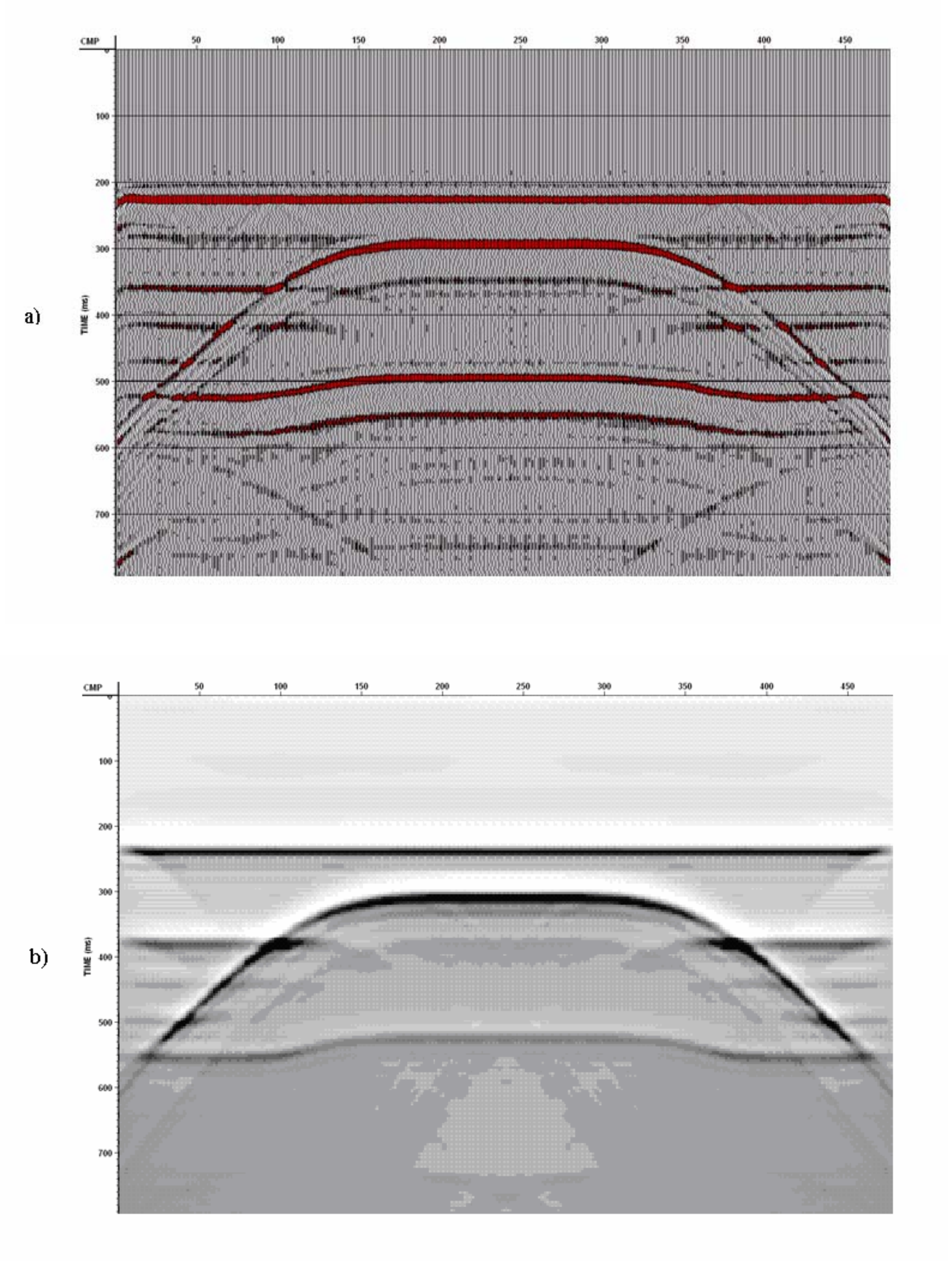

FIG. 8. a) CMP stack processed with VISTA. b) NIS image generated with *afd_explode*.

A series of processing steps was performed on the modelled shots to produce a common mid-point (CMP) stack. This stacked section approximates a conceptual model known as the zero offset section (ZOS). This in turn approximates a normal incidence section (NIS) which can be generated using a function called *afd_explode* within the CREWES finite difference toolbox. *Afd_explode* uses the concept of the exploding reflector model where all the reflectors are lined with explosives and the velocities are halved due to a one way travel time. At t=0, the charges are detonated, sending energy along normal incidence ray paths and is recorded at a recording plane defined by the receiver locations.

Figure 8 shows the CMP stack as well as the NIS image generated by *afd_explode*. The two images demonstrate a close resemblance suggesting that the modelled shot gathers were successfully written to SEG-Y.

CONCLUSIONS

An analysis of the modelling parameters was performed which examined the sampling of the spatial grid and the effect of various filters on the modelled data.

To examine the effects of the modelling parameters, a series of shot records generated by variable grid sizes was created to determine each record's quality and compute time. Various filters were also applied to determine its effect on the data. To quantify the quality of each shot record, a crosscorrelation was performed between each shot record and a high accuracy pilot shot record, giving a measure of quality relative to the pilot shot record. The results showed that an increase in sampling would produce an enhanced result, but required a more intensive computation process. In addition, the filters would increase the correlation amplitude by removing parts of the high frequency numerically dispersive signal generated by the sampling of the spatial grid.

By examining the results of the parameter analysis, modelling parameters were chosen and a survey was performed over the reef model. The modelled data was written to SEG-Y and subsequently loaded into VISTA for processing. A CMP stack was generated in VISTA along with a NIS image generated using the concept of the exploding reflector model. These two images were generated by different means but should resemble one another if the process by which they were created was performed properly. A comparison of these two images shows a close resemblance suggesting that the modelled data was successfully written to SEG-Y.

ACKNOWLEDGEMENTS

Much thanks to Chad Hogan and Gary F. Margrave for supervising this project, Kevin Hall and other CREWES members for their assistance and insightful discussions, the CREWES sponsors and GEDCO for their generous donation of VISTA.

REFERENCES

C.F. Youzwishen, G.F. Margrave, 1999, Finite Difference Modelling of Acoustic Waves in MATLAB: CREWES Report, **11**.

O. Holberg, 1987, Computational Aspects of the Choice of Operator and Sampling Interval for Numerical Differentiation in Large-Scale Simulation of Wave Phenomena: Geophysical Prospecting, **35**, 629-655.

K.R. Kelly, R.W. Ward, Sven Treitel, and R.M. Alford, 1976, Synthetic Seismograms: A Finite-Difference Approach, Society of Exploration Geophysicists.

Dan D. Kosloff and Edip Baysal, 1984, Forward Modeling by a Fourier Method, Society of Exploration Geophysicists

Kurt J. Marfurt, 1984, Accuracy of Finite-Difference and Finite Element Modeling of the Scalar and Elastic Wave Equations, Society of Exploration Geophysicists.

APPENDIX A

```
% Creates a velocity model resembling a reef structure
% The function ‘nearmod’ is a modification of the function ‘near’

dx=5;                                   %bin spacing for x and z
xmax=2400;zmax=1000;                    %maximum line length and maximum depth
x=0:dx:xmax;                            % x coordinate vector
z=0:dx:zmax;                            % z coordinate vector
vair=333;                               % air velocity
velrock=[2500 3000 3500 3700 4000 4500]; % velocity of various rock units
vlow=min(velrock);

%initialize velocity matrix as a constant matrix full of vair
vel=vair*ones(length(z),length(x));

topoz=ones(size(x))*20;           % function that defines topography as a function of x

%install topography
for k=1:length(x),
   vel(nearmod(z,topoz(k)):nearmod(z,zmax),k)=velrock(1);
end

%layered strata
for k=1:length(x);
   vel(nearmod(z,300):nearmod(z,zmax),k)=velrock(2);
end
for k=1:length(x);
   vel(nearmod(z,500):nearmod(z,zmax),k)=velrock(3);
end
for k=1:length(x);
   vel(nearmod(z,600):nearmod(z,zmax),k)=velrock(4);
end
for k=1:length(x);
   vel(nearmod(z,800):nearmod(z,zmax),k)=velrock(6);
end

%reef
lx=0:dx:1200;
lx1=1200+dx:dx:xmax;
za=[-150*tanh(0.008*(lx-600))+550, -150*tanh(0.008*(1800-lx1))+550];
for k=1:length(x),
   vel(nearmod(z,za(k)):nearmod(z,800),k)=velrock(5);
end
```

APPENDIX B

```
% Run a survey and write to segy
% Geometry output is intended for VISTA, other programs may be off by a scale factor
% Segments of code written by Kevin Hall, Chad Hogan and Joanna Copper

% Input velocity model must include the following variables
% dx      = spatial sampling of velocity model and spatial stepsize for finite differencing (must be the same)
% x       = vector of sampled x coordinates
% z       = vector of sampled z coordinates
% topoz = vector that defines topography (same length as x)
% vel   = velocity matrix
% vlow  = minimum rock velocity
% xmax = maximum x value
% zmax  = maximum z value
reef_model          % calls the m-file reef_model in the current directory

% Setup header info and model parameters
shotspa             = 30;                           % shot spacing
xshots              = 3*dx:shotspa:xmax-3*dx;       % x vector of shot locations
yshots              = zeros(size(xshots));          % y vector of shot locations
xrec                = 0:5:xmax;                     % x vector of reciever locations
yrec                = zeros(size(xrec));            % y vector of reciever locations
dt                  = .004;                         % temporal sample rate
tmax                = 2*zmax/vlow;                  % maximum time
nSamples            = ceil(tmax/dt);                % number of samples in trace
nRecX               = length(xrec);                 % number of x reciever locations
nRecY               = 1;                            % number of y reciever locations
nShots              = length(xshots);               % number of shots in survey
sline               = 1;                            % shot line number (1 for 2D survey)
dtstep              = 0.000680;                     % time step for finite differencing
lap                 = 2;                            % 2nd order Laplacian (1), 4th order Laplacian (2)
shotd               = 10;                           % shot depth
filt                = [0 10 50 60];                 % ormsby filter
outfile             = 'reef_model.sgy';             % output segy filename

% Create SEGY output file
[OUT, message] = fopen(outfile, 'w', 'ieee-be');    %Force IEEE big endianess
if (message) warning (message); end

% Set Text Header and write to file
texthdr = SEGY_GetTextHeader;
texthdr = SEGY_ModifyTextHeaderLine(23,' ',texthdr);
texthdr = SEGY_ModifyTextHeaderLine(24,['ORIGINAL FILENAME = ',outfile],texthdr);
texthdr = SEGY_ModifyTextHeaderLine(25,'SOURCELINE = SHOTLINE*2-1',texthdr);
texthdr = SEGY_ModifyTextHeaderLine(26,'NON-STANDARD SEGY TRACE HEADERS',texthdr);
texthdr = SEGY_ModifyTextHeaderLine(27,'  SOURCELINE*1000 +SOURCESTN = BYTE 181, IEEE
FLOAT',texthdr);
texthdr = SEGY_ModifyTextHeaderLine(28,'  RECLINE*1000 +RECSTN = BYTE 185, IEEE
FLOAT',texthdr);

SEGY_WriteTextHeader(OUT, texthdr, 'ebcdic');

% Set Binary Header and write to file
binhdr              = SEGY_GetBinaryHeader;
binhdr.hdt          = dt * 1000000;         % dt in s, hdt in micros.
binhdr.dto          = binhdr.hdt;
binhdr.hns          = nSamples;
binhdr.nso          = binhdr.hns;
binhdr.format       = 5;                    % IEEE floating point, SEGY rev 1.

SEGY_WriteBinaryHeader(OUT, binhdr);

% Initialize Trace Header
trace               = SEGY_GetTrace;
% Next two lines assume nShots, nRecX, nRecY are the same for all survey lines
trace.fldr          = (sline-1)*nShots;             % First FFID in shotline.mat
trace.tracl         = (sline-1)*nRecX*nRecY*nShots; % First trace number of line in shotline.mat
trace.scalco        = -100;                          % Assume two decimal places for conversion to int
trace.ns            = binhdr.hns;                % number of samples
trace.dt            = binhdr.dto;                % sampling rate
trace.d1            = 1;                         % optional header at byte 181,
                                                 % used for sou_sloc.
trace.f1            = 1;                         % optional header at byte 185, float
                                                 % used for srf_sloc.

% Run shots and write to segy
for ind=1:length(xshots),
   zshots(ind)           = topoz(nearmod(x,xshots(ind)))+shotd;     % z position of shot
   ix                    = nearmod(x,xshots(ind));                  % bin location of xshot
   iz                    = nearmod(z,zshots(ind));                  % bin location of zshot
   snap1                 = zeros(size(vel));
   snap2                 = snap1;
   snap2(iz(1),ix(1))    = 1;
   [shotf,shot,t,zrec]   = afd_shotrec_topo(dx,dtstep,dt,tmax,vel,snap1,snap2,xrec,filt,0,lap,topoz);
                                                        % generate shot gathers
   shotGather            = shotf;
   trace.ep              = trace.ep+1;                  %source number
   trace.d1              = (sline*2-1)*1000 +ind +100;  %LLSSS L=sourceline,S=shotpoint
   trace.fldr            = trace.fldr+1;                %ffid
   trace.sx              = xshots(ind);                 %sx
   trace.sy              = yshots(ind);                 %sy
   trace.selev           = zshots(ind);                 %shot elevation
   trace.sdepth          = shotd;                       %shot depth
   trace.tracf           = 0;                           %trace number in file

   for nx = 1:nRecX
      trace.cdp = nx;
      for ny = 1:nRecY
         trace.cdpt  = ny;
         trace.f1              = (ny*2)*1000 +nx +100;        %LLRRR L=receiverline,R=receiverpoint
         trace.tracl           = trace.tracl+1;               %trace number in line
         trace.tracr           = trace.tracr+1;               %trace number in reel
         trace.tracf           = trace.tracf+1;               %trace number in file
         trace.gx              = xrec(nx);                    %receiver x
         trace.gy              = yrec(ny);                    %receiver y
         trace.gelev           = zrec(nx);                    %receiver elevation
         trace.data            = shotGather(:,trace.tracf);   %trace data

         SEGY_WriteTrace(OUT,trace,binhdr.hns);
      end
   end
end
fclose(OUT);
```