A MATLAB version of Equivalent Offset Migration (EOM)

John C. Bancroft

ABSTRACT

Equivalent offset migration continues to be a viable method of prestack migration that is used in anisotropic prestack depth migrations, but is more commonly used in prestack time migrations. The continued interest has prompted further developments in a MATLAB computing environment. This version uses a script or text file to define files and parameters. The input data are required to be in SEGY format that have been corrected to a horizontal datum, all statics applied, but with no moveout correction. The output is a SEGY file of CSP gathers, which are ready for velocity analysis. The moveout velocity of these gathers is of the RMS type that is independent of dip. Moveout correction, amplitude scaling, and stacking complete the prestack migration.

INTRODUCTION

Equivalent offset migration is based on Kirchhoff prestack migration but forms prestack migration gathers that are referred to as common scatterpoint (CSP) gathers or equivalent offset (EO) gathers. These gathers are formed by summing the prestack data into a gather at an offset defined by the equivalent offset h_e. There is no time shifting of the input data when it is summed into the EO gather. EO gathers are formed at the location of each migrated trace (Bancroft et al. 1998).

Very little knowledge of the subsurface velocities are required when forming the gathers, however, after the gathers are formed, velocity analysis provides accurate velocities. Coherent reflection energy is constructively reinforced from all data within the migration aperture and lie on hyperbolic paths with moveout defined by the RMS velocities, V_{rms}. These velocities are independent of the geological dip and are quite different from the stacking velocities, V_{stk}, obtained from a conventional CMP gather, i.e. $V_{stk} = V_{RMS} / \cos(dip)$. The RMS velocities derived from the EO gathers are suitable for other prestack migrations, but are also used to apply moveout correction to the EO gathers where scaling and stacking complete the prestack time migration.

The equivalent offset h_e is obtained by equating the traveltime estimated from the prestack double-square-root equation to one containing a single square-root equation. The single square-root equation assumes a coincident source and receiver location at the offset h_e.

The EO method of prestack time migration may be modified for different prestack migration applications such a imaging of converted wave data, estimating statics, amplitude versus offset analysis, 2D or 3D data, crooked lines, and even anisotropic prestack depth migration.

EXAMPLE OF EO VELOCITIES

The semblance plots in the following figure illustrate the insensitivity of the velocities used when forming the EO gather. The first semblance plot in Figure 1a, was formed from a CMP gather. The second plot (b) used an infinite velocity when forming the EO gather, and the last figure (c) incorporates velocities from one iteration of velocities derived from EO gathers. The data is from Alberta.

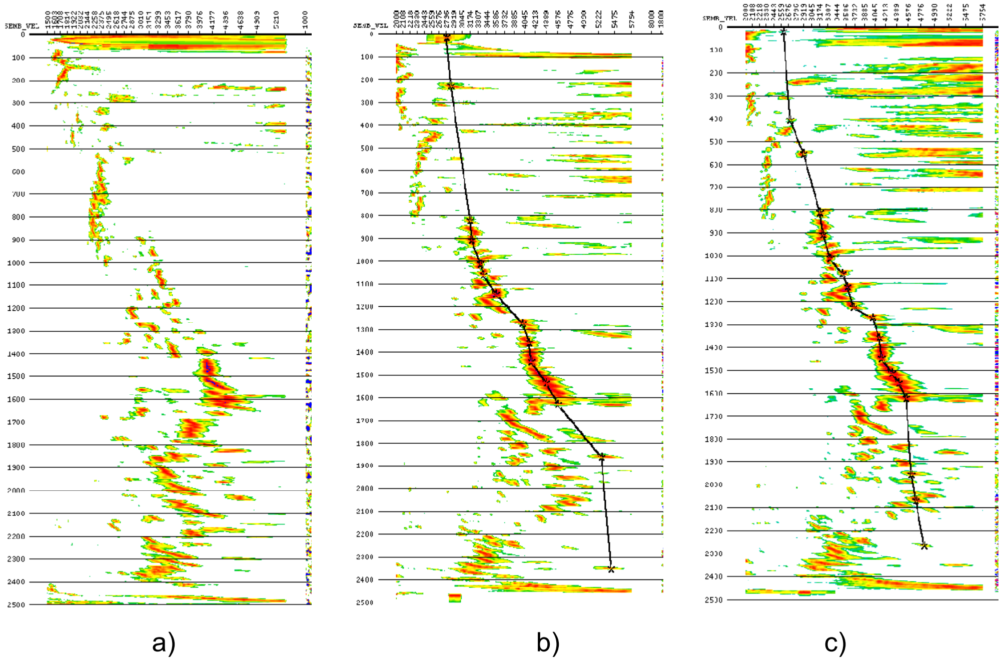

FIG. 1 Examples of semblance plots for a) a CMP gather, b) an EO gather using infinite input velocities, and c) an EO gather using one iteration of EO estimated velocities.

EXAMPLE OF EO PROCESSING

Figure 2 is an example from a synthetic numerical data set of an overthrust that has 86 source records, each with 201 traces, and 1251 samples per trace. The velocity model varied from 2815 to 3231 m/s, but only one velocity of 3000 m/s was used to form the CSP gathers. The following prestack time migration used the current MATLAB version with a single velocity, and took approximately 10 minutes on a desktop computer.

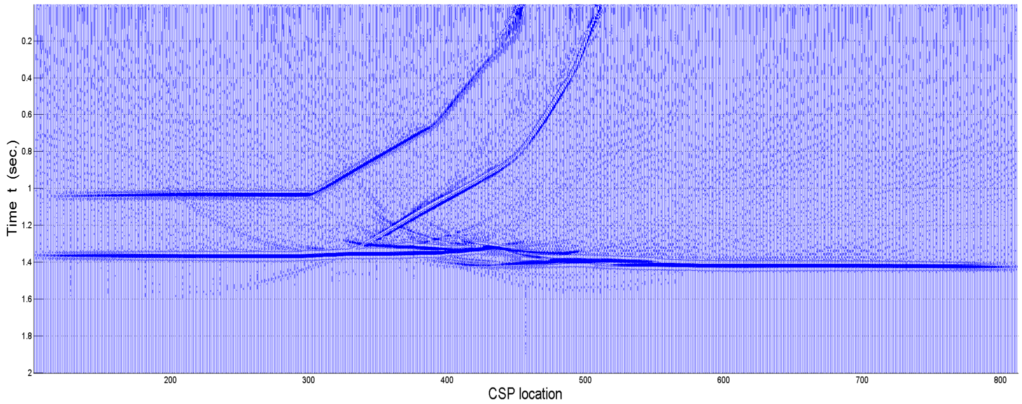

FIG. 2 An example of EO prestack time migration

BASICS OF THE CURRENT MATLAB VERSION

Why MATALB?

MATLAB is a standard programming language used by CREWES and is very flexible for displaying data. It is simple to add components and is user friendly. MATLAB is efficient for vectorized programming and runs on parallel machines. Development is straight forward and the code can then be exported to other languages if required.

What does the code do?

The code reads the input data from a SEGY file and outputs EO gathers to a SEGY file. A prestack migration is also computed and displayed, with the moveout correction part based on the input velocities.

Alternative processing software such as PROMAX or VISTA is required to read the EO gathers and perform velocity analysis. This software must also be used to create a velocity file.

What is available now?

The current version is basic 2D or 3D P-P with simple amplitude scaling. It assumes pre-processing with static corrections applied, but no moveout correction. The code is standalone, i.e. SEGY in, SEGY out.

There are a number of ways to define the velocity model that vary from a constant starting velocity, to an accurate model produced by velocities picked from a previous run.

The software contains its own paging of input data. Large files can be processed in small computers, as long as the size of the SEGY file is compatible with the operating system.

What is coming?

- Improved dip limiting capability.
- More amplitude corrections such as fold balancing and AGC.
- Ability to handle converted wave data.
- Velocity analysis.
- Tilt correction for 2D and 3D data.
- Apply static corrections to the input data.
- Azimuthal anisotropic velocity analysis for 3D data.
- Non-hyperbolic anisotropic data.

Data format

The input prestack data is required in SEGY format, and the output EO gathers is also in SEGY format.

A velocity model of RMS velocities in SEGY format can be supplied to match the desired EO gathers. The input velocities can also be assumed to be infinite, a constant value for the entire line, or one linear velocity for the entire line.

Problems

Ensuring the input data is SEGY compatible.

Ensuring the input data has all statics corrections applied.

A processing sequence

I suggest that the first run creates one, or a few EO gathers using a best guess velocity. This run takes seconds, depending on the size of the input data.

Pick the velocities from this or these EO gathers.

Use these velocities to create a velocity model for input to a second run of EO gathers at large incremental locations for building a velocity model that depends on the geological structure. This run may take up to a minute, depending on the size of the input data.

A third run with EO gathers all desired CMP locations. Pick final velocities at locations that can be accurately interpolated to follow the model. In highly structured areas these analysis locations may be at every few CMP locations.

Use the final pick velocities for moveout correction and stack.

The final moveout corrected gathers are prestack migration gathers that includes offset that may be well beyond the conventional CMP gather. Care should be taken when muting the data.

Input parameters

All input parameters are handled with a text file that is referred to as a script in MATLAB. The text contains the names of the input and output files and the parameters that control the migration. The following is an example of the text file with the line comments removed:

```
InputSGYFile EOMdata\thrust1_rd.sgy     % Input SEGY daata file
VelSGYfile   EOMdata\rmsvel.sgy         % Input SEGY velocity file if needed
CspgSGY      EOMdata\CSPgathers29July2010.sgy % Output SEGY gather file
Velocity   1            %  SEGY file only
FirstCMP   102 -1000 0  % First CMP number, and location (x and y) If no velocity file
LastCMP    812 6100 0   % First CMP number, and location (x and y)
Tsamp      0.002        % Input time sample increment
FirstCSP   275  730 0   % First CSP: CMP location, X, Y
LastCSP    325  1230 0  % Last CSP:  CMP location, or X, Y
NumCSPs    0            % Number of CSPs to compute (Inc. might not be 1, =0 OK)
Bins       201 15.0     % Number of bins and increment in a CSP gather
TmaxCSP    2.0          % Maximum time (seconds) for CSP gathers and stack
EOmethod   3  1         % Gathers:[ Type : Sides ]  Sides = 1, 2, or n = 3D multi
TincType3  0.040        % Time increment for Itype 3
Dim2D3D    2            % Dimension, 2D or 3D, = 2 or 3.  [2]
FoldGather 1            % Use of CSP gather fold counter.  0 = No, 1 = Yes
DipLim     60 80        % Dip limits in degrees for moveout (stretch limit)
```

```
NMO        1            % Normal moveout to CSP gathers: 0 = no, 1 = yes
StackOpt   1            % Stack CSP gathers: 0 = no, 1 = yes
RhoFilter  1            % Apply rho filter to stack: 0 = no,  1 = yes
SaveCSPg   0            % Save CSP gathers to SEGY file, 0 = no, 1 = yes
Idebug     2            % Debug level:  0 = Minimal, 10 = verbose output
End                     % Nothing read after first "End"
```

The order of the lines in the text file are not important, but the last line must be, or is assumed to be the “End” line.

CONCLUSIONS

A MATLAB version of the equivalent offset prestack time migration was introduced.

ACKNOWLEDGEMENTS

We wish to thank the sponsors of the CREWES project for supporting this work.

REFERENCES

Bancroft, J. C., Geiger, H. D., and Margrave, G. F., 1998, The equivalent offset method of prestack migration, Geophysics, Vol. 63, N0. 6 (December 1998) P. 2042-2053.