
Documentation for MATLAB equivalent offset migration

John C. Bancroft

ABSTRACT

Equivalent Offset migration (EOM) has become a valuable algorithm for prestack migration. A prestack time version of EOM has been developed on the MATLAB platform, and this paper is essentially an introduction and the beginning of a user manual.

The algorithm assumes the input data is in SEG-Y format. A velocity file may also be used as input to define the location for the output gathers. A text file is used to define the SEG-Y files, and parameters. The output is a SEG-Y file that contains the CSP gathers that are ready for velocity analysis. These velocities are then used to apply moveout corrections to the CSP gathers, which are then stacked to complete the prestack time migration.

INTRODUCTION

Equivalent offset migration

Equivalent Offset migration (EOM) is based on Kirchhoff prestack migration and can be either a time or a depth migration (Bancroft, et al. 1998). Most applications are time migrations, but a successful prestack depth migration that includes anisotropy is in use (Bancroft and Vestrum, 1999).

Prestack time EOM forms prestack migration gathers that are referred to as common scatterpoint gathers (CSP) or equivalent offset gathers (EO). They are formed by summing all input traces into all the CSP gathers. The equivalent offset of an input trace is defined by the location of the source and receiver, relative to the location of the CSP gather. The input traces are summed or stacked into the gather with no time shifting. Moveout correction, which is part of the prestack migration process, is delayed until after the CSP gathers are formed. After the gathers are formed, velocity analysis produces accurate root-mean-squared (RMS) type velocities, which are then used to apply moveout corrections to the gathers, then stacking these gathers completes the prestack migration.

The velocities obtained with EOM are not stacking velocities that are affected by dip, but true RMS velocities that are independent of dip.

The formation of the CSP gathers is insensitive to velocities. A very reasonable solution can be obtained by assuming the velocity to be infinite, giving an asymptotic solution. The data can be improved slightly by using an approximate velocity. Typically, a single linear velocity for the project is all that is required for forming the optimum CSP gathers.

Once the gathers have been formed, \ accurate velocities are picked using conventional techniques. Note that these gathers are migrated gathers and the Fresnel zone has been reduced (in theory) to just one trace. Consequently, in structured areas, the velocities

may be picked at every trace if necessary. It is not necessary to form super gathers as the fold in each CSP gather is very high.

Parameter file

A parameter text file contains all the information to run a job. Its name must be EOMtextFile.m, which is read when the job commences.

The text file is assumed to be in a directory that is accessed by MATLAB.

Any text file could have been used, but I prefer MATLAB script file (*.m) that is available in the MATLAB Editor. The colours of the variable names, parameters, and comments conform to the MATLAB code. In the following example the number of samples in a SCP gather are defined using a parameter name, the number, and a comment, i.e.,

```
NsampCSP      1001      % Number of samples in CSP traces (Vel. and Gathers) .
```

Each line of the text file may contain a file path and name, a single parameter, or a number of parameters. A comment card or inline comment begins with a “%” sign. An example of an input file line is

```
InputSGYFile  
C:\Projects\Hussar\HussarInputFiles\INOVA_LowDwell_vert_radialfilt_4sec.sgy
```


and a multiple parameters line defining a CSP number and its *x* and *y* location is

```
FirstCSP      237  372770 5672443 % First CMP number, and location (x and y),
```

Comment lines can be intermingled with the parameter lines, and there is no prescribed order to the lines. An “End” line stops reading the parameters, however, lines can be added after this card for saving or storing unused lines.

An example of a complete “Job deck” is contained in the Appendix.

Running EOM-Matlab

The job may be run from the command window by entering EOMmain. It can also be run from the MATLAB Editor window by opening “EOMmain.m”, and making sure it is the selected window, then clicking on the the run icon  in the editor window illustrated below in Figure 1.

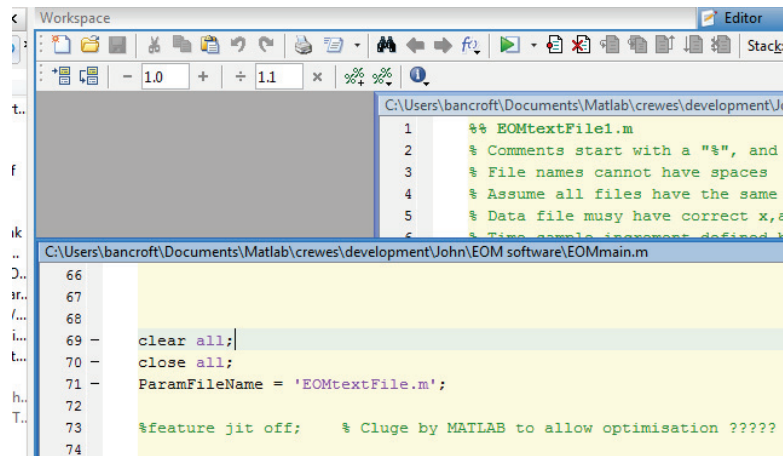


FIG. 1 MATLAB Editor with `EOMmain.m` in the active window.

Memory management

Earlier versions of MATLAB stored data in contiguous memory arrays for faster access. It appears that this is not the case in WINDOWS 7. It has become easier to use very large arrays, however, paging of the data between the CPU memory and the disk can become a problem. To prevent paging (if it is possible), I restrict the amount of memory that EOM uses, and read only part of the input data file and compute only part of the output CSP gathers that fit within a pre-allocated value. Consequently, the amount of memory used by EOM-MATLAB is restricted by a parameter line

```
CPUMemAlloc 4000 % Maximum memory allocated by the user, in megBytes.
```

If using Windows, I recommend opening the Window Task Manager and identify how much user memory is available, then allocate 75% of that value. This will become a default option.

Of the memory that is available, more memory is allocated to the CSP gathers to reduce the number of time the input data file must be read.

Program outputs

The main output of the program is a SEG Y file that contains the CSP gathers.

```
CspgSGY C:\Projects\Hussar\HussarCSPgathers\CSPgHus09Nov2011EO2LinVel.sgy
```

There are other outputs in the form of MATLAB figures that contain the central CSP gather. If velocity information is provided, the moveout corrected central SP gather can also be displayed, along with a preliminary brute stack. At this time they are not saved as SEG Y files.

Runtime text is produced in the MATLAB Command Window. The amount of text can vary from a few lines that identify the selected processing steps, to a verbose amount that details the parameters and some local values, such as the geometry, to verify that it has been read correctly. These options are selected by various values of the “Idebug” line that is defined later.

EOM options

The basic definition of the equivalent offset h_e is

$$h_e^2 = x^2 + h^2 - \frac{4x^2h^2}{T^2V_{RMS}^2}.$$

where x is the distance between the CMP location and the CSP (migration) location, h the source-receiver offset, T the time of the sample, and V_{rms} the RMS velocity. The last term in this equation is time and velocity dependent. It is usually small and can be ignored in most runs. If ignored, then each input trace can be saved efficiently to reduce runtimes. There are a number of options available to choose shorter runtimes or more accurate gathers.

Velocity information

In a typical run, a fast version of EOM is used, and no velocity information is required. However, a slower, more accurate, version does use RMS velocities as part of the equivalent offset (EO) computation. These velocities may be entered as a SEG Y file, a constant velocity, or a linear velocity.

An input SEG Y file may be used to define the RMS velocities at CMP locations, and can be used to define the output locations of the CSP gathers. This file must contain the (x, y) locations of the CMP gathers. Options allow a subset of the CMP locations to be used such as an integer increment, or a range of CMP gathers.

When using a SEG Y file, the locations of the CSP's can be organized in any manner, as long as their location is defined. In a 3D project, the gathers could be organized as:

- meandering lines,
- arbitrary 2D lines at any azimuth,
- multiple 2D lines, or
- random location for an initial velocity analysis.

Most of the problems with coding involve the SEG Y files, and the different versions that can be expected. Defining the trace geometry (x, y) can be difficult as the type of data and their locations can vary. At this time we use the standard SEG Y format with the (x, y) locations stored in both the source and receiver locations. Several iterations of setup can be expected, and the data should be verified.

If a SEG Y velocity file is not used, the CSP gathers must be in a straight line that is defined by the first and last CSP numbers and their (x, y) locations.

Since the velocity information has a slight effect on the equivalent offset, only one smooth velocity profile is typically required to accurately form the gathers. The velocity analysis after the gathers are formed provides accurate RMS velocities.

Runtimes

This MATLAB version runs very slowly and we are continually trying to make it faster. For example, a slight change in moveout correction code changed the run times from 80 s to 30 ms. Vectorization of the code is essential.

DESCRIPTION OF PARAMETERS

The parameters are defined using lines in the parameter text file *EOMtestFile.m*. Many of the parameters have default values, and do not need to be defined. Some parameters are not used at this time. Some parameters could be defined more efficiently, and hopefully will be soon. (An example is the option to produce a brute stack using the input or defined velocities. The options for applying moveout correction could be included with the stack option.)

CPUMemAlloc 4000 % Maximum memory allocated by the user in megBytes.

Allocate memory to prevent disk paging in MB. Use 75% of the maximum memory available. (See the task manager).

Default 1000

InputSGYFile

C:\Projects\Hussar\HussarInputFiles\INOVA_LowDwell_vert_radialfilt_4sec.sgy

Define the input SEGY data file.

Default: 'None'

VelSGYFile

C:\Projects\Hussar\HussarInputFiles\INOVA_LowDwell_vert_radialfilt_4sec.sgy

Define the input SEGY velocity file. This file is (can be) used to define the velocities for each output sample in the CSP gathers.

Default: 'None'

VelSGYFileVP

C:\Projects\Hussar\HussarInputFiles\INOVA_LowDwell_vert_radialfilt_4sec.sgy

Define the input SEGY P-velocities for converted wave data. Not used at this time.

Default: 'None'

VelSGYFileVS

C:\Projects\Hussar\HussarInputFiles\INOVA_LowDwell_vert_radialfilt_4sec.sgy

Define the input SEGY S-velocities for converted wave data. Not used at this time.

Default: 'None'

VelGeomAcmp VelGeomAx VelGeomAy

Define the geometry of the first CMP location to override the parameters provided in the SEGY velocity file, if there are problems with the SEGY file.

VelGeomAcmp = first CMP, VelGeomAx = x, VelGeomAy = y

Default: 0.0

VelGeomBcmp VelGeomBx VelGeomBy

Define the last CMP geometry to override the parameters provided in the SEG Y velocity file.

VelGeomBcmp = first CMP, VelGeomBx = x, VelGeomBy = y

Default: 0.0

NtrVelIn 753; % Number of sample in the SEG Y velocity trace

Override the number of traces in the input SEG Y velocity file.

Default: 0

NsampVelIn 2001; % Number of sample in the SEG Y velocity trace

Override the number of sample in the input SEG Y velocity file.

Default: 0

TsampVelIn 0.002; % Time sample interval for SEG Y velocity traces

Over ride the time sample increment in the SEG Y velocity file.

Default: 0

ScaleDataIn 10.0; % Scale the input data (SEG Y problems)

Scale the input data if required (SEG Y problem)

Default: 1

ScaleDataXYIn 0.01; % Scale the input data X and Y location (SEG Y

Scale the geometry of the input data, i.e. the x, and y locations (SEG Y problem).

Default: 1

ScaleVelIn 1.0; % Scale the amplitude of the velocities (SEG Y problems)

Scale the amplitudes of the velocity file (SEG Y problem).

Default: 1

ScaleVelXYIn 0.01; % Scale SEG Y velocity locations X and Y (SEG Y problems)

Scale the velocities of the input SEG Y velocity file (SEG Y problem).

Default: 1

CspgSGY C:\Projects\Hussar\HussarCSPgathers\CSPgHus09Nov2011E02LinVel.sgy %

Name of the output file for the CSP gathers

Default: 'None'

```
Velocity      12  2200  5900      % A linear velocity for all CSP's
```

Velocity options:

1 Use SEGY velocity file. CSP gathers match CMP gathers in the velocity file.

2 N Use SEGY file. Use every N^{th} CSP gather

3 N1 N2 Use SEGY file. CSP gathers in a range of CMPs from N1 to N2.

```
Velocity  3  500  505
```

4 CMP Use SEGY file. Random location of CSP's. One per line.

```
Velocity  4  245
```

```
Velocity  4  255
```

```
Velocity  4  265
```

11 V Specify a constant velocity. Requires CSP geometry.

11..V1 V2 Specify a linear velocity. Requires CSP geometry.

Default: 1

```
FirstCSP    237  372770  5672443 % First CMP number, and location (x and y)
```

First CSP number and x, y location. Can be used to override the SEGY velocity file.

Default: 0

```
LastCSP     1128  369462  5669460 % Last CMP number, and location (x and y)
```

Second CSP number and the x, y location. Can be used to override the SEGY velocity file.

Default: 0

```
CSPincNum   2        % Increment number of CPS's
```

Increment the CSP number when defining the velocity with parameters.

Default: 1

```
Obliquity   1        % Angle of obliquity degree from normal. Default = 0 degrees
```

Obliquity angle in degrees from the 2D line (use with extreme caution).

Default: 0

```
EOMethod    4  1        % Gathers:[ Type : Sides ] Sides = 1, 2, or n = 3D multi
```

Type of EO calculation to be used, and N sided gathers. (Only 1 sided is available at this time).

1 1 Asymptote only. Fastest.

2 1 Asymptote with bin interpolation and some amplitude scaling.

3 1 Sample by sample h_e . Very accurate, extremely slow.

4 1 Copy an array for each bin. Best use, but much slower than 2.

Default: 1 1

TincType4 0.050 % Time increment for Itype 3

If EO type 4 is used, then this number is the size of the pre-calculated array in seconds.

Default: 0.050

Bins 501 5 % Number of CSP bins and bin increment

Number of bins and the bin increment in feet or meters.

Default: 0 0

NsampCSP 1001 % Number of samples in CSP traces (Vel. and Gatherers)

Number of samples in the CSP gather. Must be less than or equal to the input data.

Default: 0

TsampCSP 0.002 % Input time sample increment

Time sample increment of the CSP gathers. Must be the same as the input data, and in the SEG Y velocity file if it is used.

Default: 0

Dim2D3D 2 % Dimension, 2D or 3D, = 2 or 3. [2]

Type of project, 2D or 3D. There is a difference in the scaling between a 2D and a 3D project.

Default: 2

FoldGather 1 % Use of CSP gather fold counter. 0 = No, 1 = Yes

Use of a fold gather. This balances the amplitudes of the CSP gather by maintaining a fold count for each sample in the SP gathers. It doubles the amount of memory required for the CSP gathers if used. It can be a fold value for low noise data, or the square-root of the fold for very noisy data. Testing is required. More coding is required to make the square-root optional.

Default: 1

NMO 1 % Normal moveout to CSP gathers: 0 = no, 1 = yes

Apply moveout correction to the CSP gather for a brute stack

Default: 1

DipLim 20 30 % Dip limits in degrees for moveout (stretch limit)

Dip limits on the data after moveout correction and only used for the brute stack. Dips are tapered to zero over the defined range.

Default: 50 60

AmpCos 1 % Cosine amplitude scaling
 Apply amplitude scaling of T0/T to the CSP gathers. Recommended
 Default: 1

AmpT0 1 % Amplitude scaling for T0
 Apply amplitude scaling for T0.
 Default: 0

AmpT 0 % Amplitude scaling for T
 Apply amplitude scaling for T.
 Default: 1

StackOpt 1 % Stack CSP gathers: 0 = no, 1 = yes
 Option to stack the CSP gathers after moveout correction.
 Default: 1

RhoFilter 1 % Apply rho filter to stack: 0 = no, 1 = yes
 Apply a rho filter (45°) to the stacked section
 Default: 1

SaveCSPg 1 % Save CSP gathers to SEG Y file, 0 = no, 1 = yes
 Save the CSP gather to a SEG Y disk file. Required unless testing.
 Default: 1

Idebug 3 % Debug level: 0 = Minimal,
 Set a debug level
 0 Minimal output to the MATLAB Command Window
 1 Main section headings
 2 Basic parameter information
 3 All non loop parameters
 4 Include some loop values such as geometry, velocities, and data.
 5 Custom use for decoding
 Default: 1

End % Nothing read after first "End"
 Last used line in the file. Other lines may follow but will not be used.

Comments on the input SEGY data file

The input data is assumed to be on the main disc drive in SEGY format. The data can be in CMP or shot record format, or any other order. The binary header should contain the time sample rate, and the number of data samples in a trace. The number of traces is computed from the size of the SEGY file, taking into account the text headers (3600 bytes) and the binary header of 400 bytes. Each trace header must contain the (x, y) location for the source and receiver. The trace should also contain the CMP number.

The file headers are first read to compute the memory requirements of the input data, and its geometry. The seismic data is not read at this time as we don't know how much can be read into the processor memory.

Data considerations

It is desirable to have as much CSP gather memory as possible to minimize the number of times that the input data is read in. If all the CSP gather data does not fit into the processor memory, the data are divided into a number of groups. Each group of CSP gathers is completely computed using all the input data, then written out to disk. Scaling moveout correction and stacking can also be applied at this time.

If all the input data does not fit into processor memory, then the data are read in as groups that are called bunches of input traces.

The geometry for the input data (source and receiver locations, CMP number, etc), the geometry of the velocities, and the geometry of the CSP gathers, are read into separate arrays and not divided into groups or bunches. This allows the geometry to be plotted after the CSP gathers are computed.

Due to difficulty with some SEGY velocity files, parameters can be included to override, or re-define the CMP (x, y) locations.

CSP locations may not be the same as CMP locations, e.g., when looking at a few CSP gathers for velocity analysis.

CMP and (x, y) geometry: The program only requires the (x, y) location of the source and receiver for each input trace, and the (x, y) location of the output CSP gathers. It does not require the CMP numbers, but they are used to tie the CSPs to the project geometry. They are also included as a backup for the input data on the input traces. If the geometry is defined using the parameter file data then the locations are computed linearly between the specified CMPs.

Related software:

EOMtextFile1.m	Text file containing input parameters
ReadParamFile.m	Read the parameter text file into EOMmain.m
EOMcspGathers.m	Form the CSP gathers
EOMmoveout.m	Apply moveout to the CSP gathers
EOMstack.m	Stack the CSP gathers
EOMrhoFilter.m	Apply a Rho filter to the stacked data

Programming considerations

First sample number (1) is defined as time zero.

Data arrays:

Parameters ***	Structure of parameters
GeomIn	Input data geometry (NgeomInPar,NtrIn)
InData	Input data (NsampIn, NtrIn)
GeomVel	Geometry for SEG Y file of velocities
VelData	Velocities in trace format (NsampVel, NtrVel)
GeomCSP	Geometry of CSP gathers (NgeomCSPpar,NumCSPs)
VelAry	Velocitiesmatching output CSP's (
CSPgath	CSP gathers (NsampOut,Nbins,Ncspg)
NMOGath	Moveout corrected CSP gathers (NsampOut,Nbins,Ncspg)
Stack	Stack of CSP gathers (NsampOut,Ncspg)

Global arrays:

Parameters, InData, CSPgath, NMOGath, Stack, VelAry

Flow chart

Flow chart for defining the output CSP gathers from the text file and the SEGY velocity file.

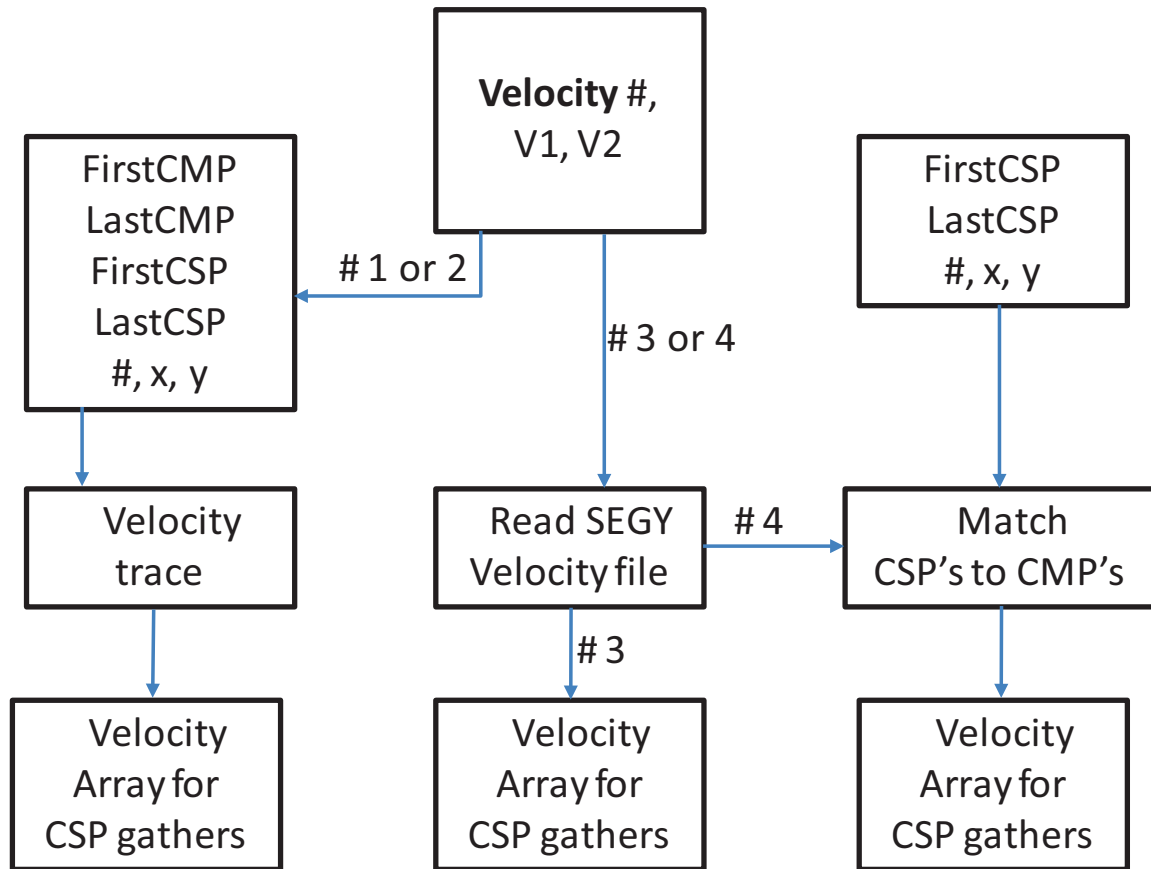
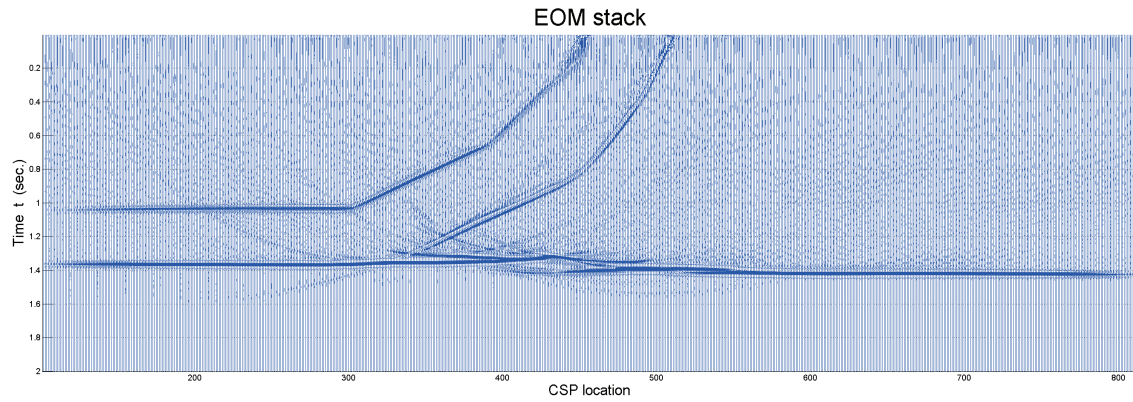


FIG. 2 Flow diagram for velocities.

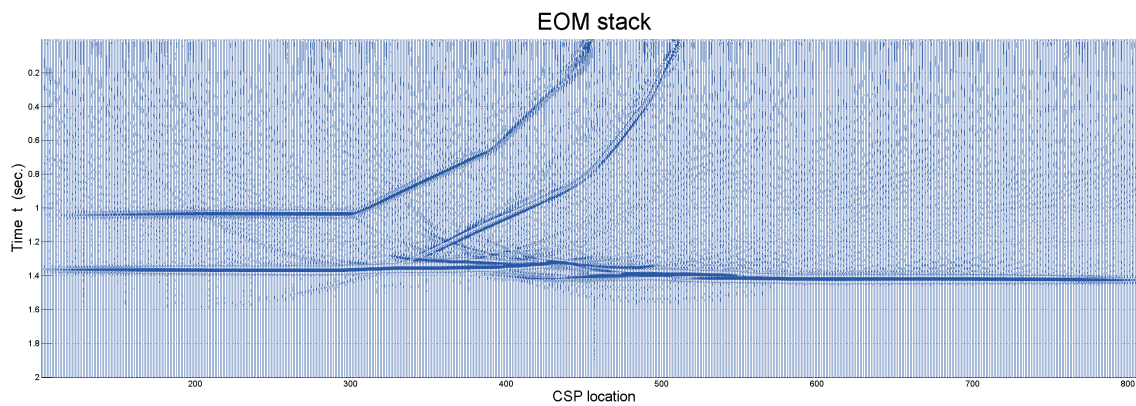
EXAMPLES OF EOM PROCESSING

The following are some tests using alternative software. I have retained them in a raw form to compare runtimes with the MATLAB version. The data will become a test set for the MATLAB version as soon as possible.

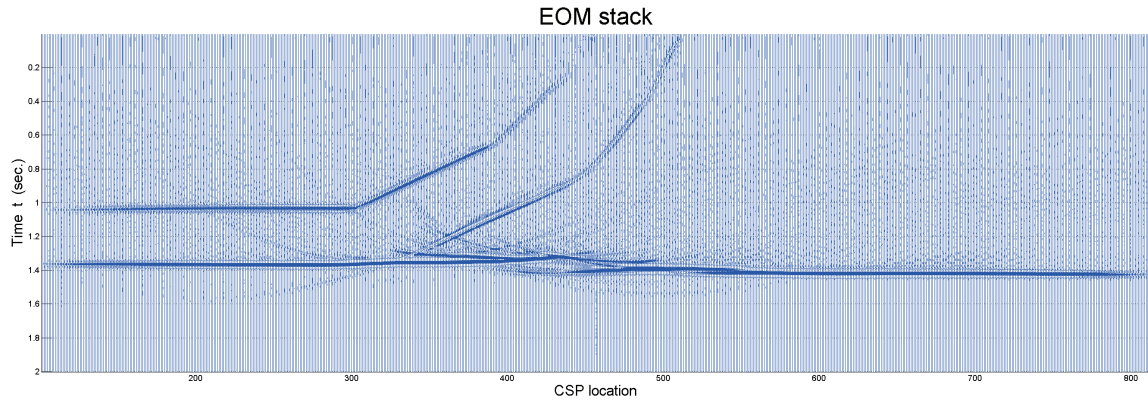
EOMmethod = 1, sqrt t scaling, no elliptical scaling: Elapsed time is 557 seconds.



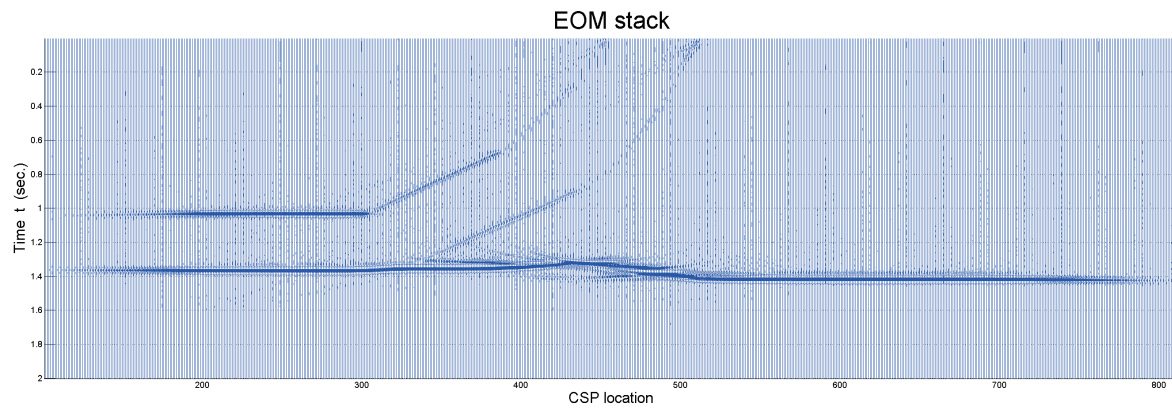
With elliptical scaling EOMmethod = 2. No elliptical amplitude scaling.



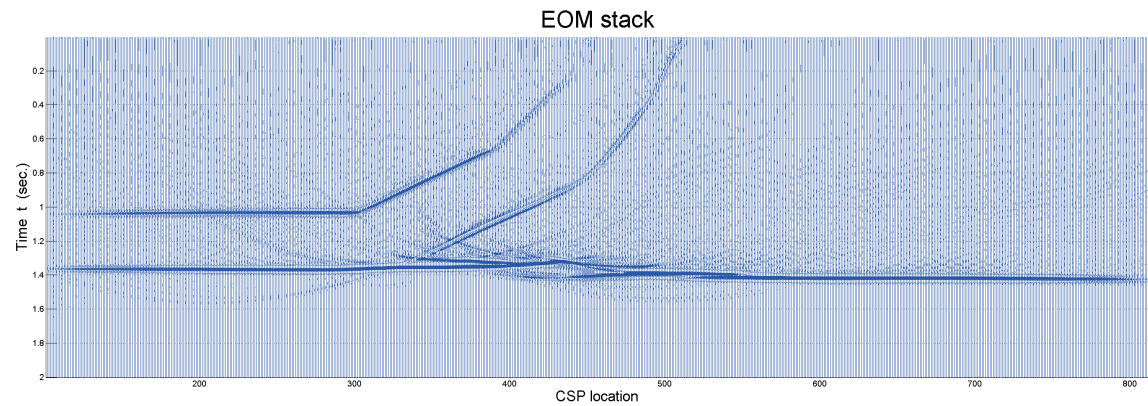
With elliptical AMP scaling: Elapsed time is 871 seconds.



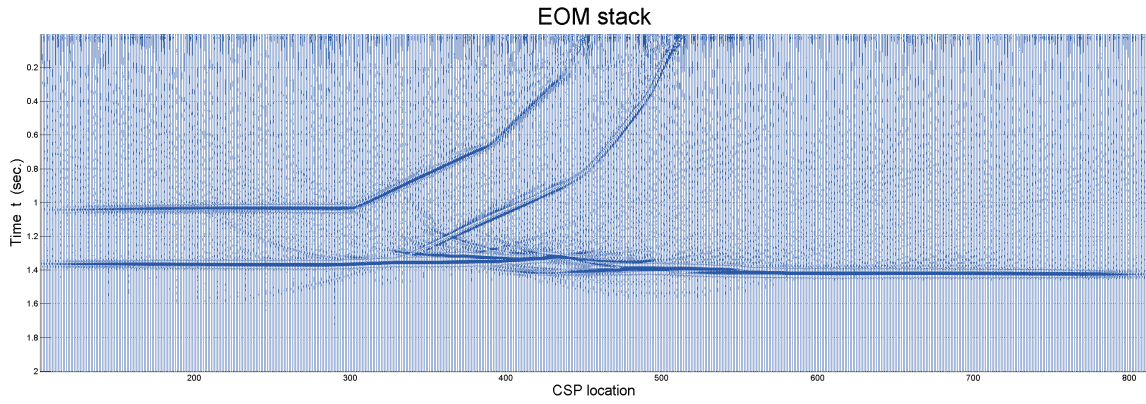
With linear bin weighting



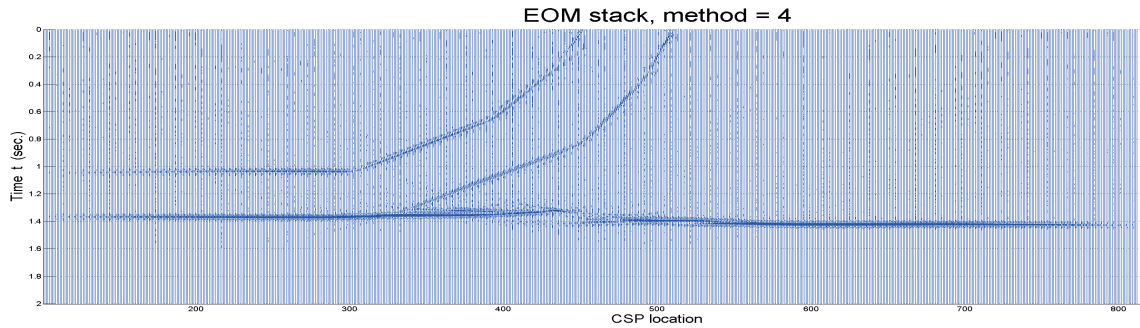
Bin weighting removed, $T_{amp} = \sqrt{1/iSamp}$



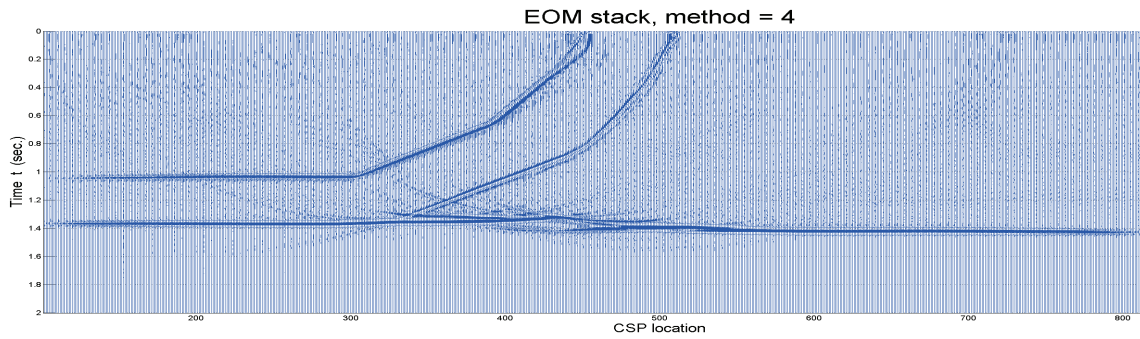
$T_{amp} = 1/iSamp$



Running EOMtype = 3 Elapsed time is 6475 seconds.



Elapsed Time: (0:00)



Elapsed Time: (0:00)

CONCLUSIONS

These pages are intended to become a user's manual for running EOM in a MATLAB environment. EOM uses an input parameter text file and SEGY data file, and produces an output SEGY data file that contains CSP gather. Velocities are not required, but can be input as a SEGY file or as parameters in the text file.

ACKNOWLEDGMENTS

The authors would like to thank all CREWES sponsors, staff and students for their support.

REFERENCES

- Bancroft, J. C., Geiger, H. D., and Margrave, G. F., 1998, The equivalent offset method of prestack time migration: *Geophysics*, 63, 2042-2053.
- Bancroft, J. C., and Vestrum, R., 1999, Anisotropic prestack depth migration using the equivalent offset method, CREWES Research Report, Vol. 11.

APPENDIX

An example of an input text file.

```

CPUmemAlloc 4000 % Maximum memory allocated by the user in megBytes.
InputSGYFile C:\Projects\Hussar\HussarInputFiles\INOVA_LowDwell_vert_radialfilt_4sec.sgy
ScaleDataIn 10.0; % Scale the input data (SEGy problems)
ScaleDataXYIn 0.01; % Scale the input data X and Y location (SEGy
NsampVelIn 2001; % Number of sample in the SEGy velocity trace
TsampVelIn 0.002; % Time sample interval for SEGy velocity traces
ScaleVelIn 1.0; % Scale the amplitude of the velocities (SEGy problems)
ScaleVelXYIn 0.01; % Scale SEGy velocity locations X and Y (SEGy problems)
CspgSGY C:\Projects\Hussar\HussarCSPgathers\CSPgHus09Nov2011EO2LinVel.sgy %
Velocity 12 2200 5900 % A linear velocity for all CSP's, t=0 until t ==
FirstCSP 237 372770 5672443 % First CMP number, and location (x and y)
LastCSP 1128 369462 5669460 % Last CMP number, and location (x and y)
CSPincNum 2 % Increment number of CPS's
Obliquity 1 % Angle of obliquity degree from normal. Default = 0 degrees
EOMethod 4 1 % Gathers:[ Type : Sides ] Sides = 1, 2, or n = 3D multi
% Type 1 = Asymptote offset, No vel, he^2 = x^2 + h^2
% 2 = Asymptote offset, ampscaling, (Vell)
% 3 = Sample by sample he, very accurate, very slow
% 4 = Bin copy (faster)
TincType4 0.050 % Time increment for Itype 3
Bins 501 5 % Number of CSP bins and bin increment
NsampCSP 1001 % Number of samples in CSP traces (Vel. and Gathers)
TsampCSP 0.002 % Input time sample increment
Dim2D3D 3 % Dimension, 2D or 3D, = 2 or 3. [2]
FoldGather 1 % Use of CSP gather fold counter. 0 = No, 1 = Yes
NMO 1 % Normal moveout to CSP gathers: 0 = no, 1 = yes
DipLim 20 30 % Dip limits in degrees for moveout (stretch limit)
AmpCos 1 % Cosine amplitude scaling
AmpT0 1 % Amplitude scaling for T0
AmpT 0 % Amplitude scaling for T
StackOpt 1 % Stack CSP gathers: 0 = no, 1 = yes
RhoFilter 1 % Apply rho filter to stack: 0 = no, 1 = yes
SaveCSPg 1 % Save CSP gathers to SEGy file, 0 = no, 1 = yes
Idebug 3 % Debug level: 0 = Minimal,
End % Nothing read after first "End"

```