A FOCI migration algorithm for uneven surfaces

Rolf Maier

ABSTRACT

A C and Fortran version of FOCI is being re-released with additions allowing for changes in surface elevation.

INTRODUCTION

The compiled version of the FOCI depth migration code, originally released last year (Maier et al., 2006), has been amended to allow for changes in topography. The method used to introduce elevations follows a scheme also used by other depth migration modules. The “elevations” that are expected are positive depths taken from an arbitrary datum. In the current case, these elevations, to keep using this term, are read from the SEGY headers or from a text file.

METHOD

The velocity file is expected to have data starting at datum (zero), but velocity values are only used once the surface elevation has been reached. Accordingly, the time traces of the shot input file are migrated from the surface downward. Any data migrated above the surface are deleted. Velocities may be specified either as text files ordered by location or trace, separated by as many or as few linefeeds as may be practical. Velocities are expected at regular intervals corresponding to the trace locations. No extrapolation is performed outside of the given velocity range, and no data will be migrated into a region where no velocities exist.

As in the original release, the input parameter file is full of comments, all of which may be deleted when no longer needed, as may be some non-essential parameters. All parameters may be rearranged at will. To create a parameter file type `foci.exe -t`. Issuing this command will print a template which clearly marks numeric parameters which must be set with a value starting with `-999`, except in the case of file names, which must always be set.

DISCUSSION

Figure 1 shows a small portion of a migration of Sam Gray’s model 94 data, with time power scaling applied to the data before migration.

The Fortran code was originally written with Fortran 2003 in mind which has on important improvement over Fortran 90 and Fortran 95, it mandates that the compiler delete space pointed to by pointers which are getting out of scope. This is, of course, always true for automatic storage, but space allocated on demand fell through the cracks. Most of these memory leaks have been removed; a few minor ones may persist.

As implemented, the code reads and writes SEGY files, but users will want to port the code into their own processing systems. The installation notes which accompany the

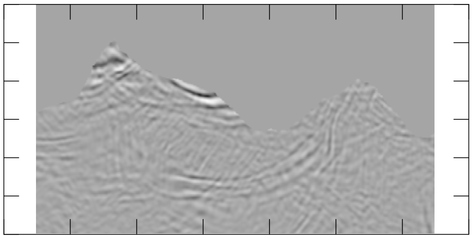

FIG. 1. Small section of a stack of migrated shots of model 94.

software contain a description of those parts of the code which need to be altered, or which may be discarded. This should assist in porting the code to work within a processing package.

ACKNOWLEDGEMENT

I wish to thank Hanxing Lu for balancing the data.

REFERENCES

Maier, R., Margrave, G., Sun, W., and Lamoureux, M., 2006, A practical implementation of FOCI – FORFOCI depth migration: CREWES Research Report, **18**.