FREQUENCY DOMAIN MIGRATION OF LARGE DATA SETS IN A MINICOMPUTER

Richard Ottolini

Frequency domain migration (FDM) has been used to migrate a several million sample data set at a rate up to a half million samples per hour in a PDP 11/70 minicomputer. [See Bloxsom in this SEP report]. The FDM algorithm is adequately described in Stolt [Geophysics, Feb., 1978] and Lynn, [SEP 11]. In FDM a one way scalar wave equation solution maps the double fourier time section image into the desired double fourier depth section image.

Advantages

FDM handles all dips and resolves to the nyquist sampling density, which is 2 points per wavelength instead of the 8 in finite difference schemes. FDM is vector oriented and has small core memory requirements, making it suitable for minicomputers and array processors.

Disadvantages

We are using the constant velocity version although there are promising solutions to the variable velocity problem. [Stolt, op. cit. and Jacobs, in this SEP report] zero padding in both lateral and time dimensions making fourier images 4 to 9 times the original data set size. The 50% redundancy of real-to-complex FFTs (negative frequencies are the complex conjugates of positive frequencies) and performing the time FFTs a vector at a time reduces storage overhead to only 1.5 - 3 times the original data set size. Producing a double fourier image requires an intermediate transpose. Large data sets must be transposed out of core using a transpose buffer twice the size of the fourier image. Therefore, until the wrap-around problem is solved the minimum diskspace needs are 5 times the original data set size.

Implementation

Our FDM program has five main stages:

(1) lateral FFT
(2) transpose
(3) time FFTs and migration mapping
(4) transpose
(5) inverse lateral FFT

A fast FORTRAN coding for the 3rd stage appears in Ottolini's slant-midpoint paper of this SEP report.

Without an array processor the FFTs double the total migration time, consuming a 60% portion of time; with an array processor, the FFTs consume only 20% of the time. Utilizing FFT symmetry reduces overall run time 50%. Out of core transpose time increases with data set size n as n*log(n) and is the bottleneck of kilo-trace data sets. Excluding FFTs, mapping and accompanying interpolation are the biggest number crunchers. Noting mapping is redundant for positive and negative time frequencies and using linear interpolation instead of geometric [See Lynn, SEP 11] speeds up the third stage three-fold. If the data or fourier image is very regular, such as for a delta function, then non-geometric interpolation seems poor.

An actual run of a 325 trace - 1875 time sample - 15 second COCORP data set took 3.5 hours on a PDP 11/70 with disk memory and Floating Point Systems array processor. [See Newkirk in this SEP report]. It was zero padded to 1024 traces and 4096 time samples. Shallow seismic data would permit migration of a 850 trace - six second -

2 ms. data set under the same disk space and time constraints. The resulting FFT images contained a million floating point numbers. 40 minutes were spent performing FFTs, 120 minutes for transposes, and 150 minutes for the migration mapping.