Monte-Carlo Statics on Large 3D Wide-azimuth Data

D. Le Meur* (CGGVeritas)

SUMMARY

Estimation of surface-consistent residual statics on large 3D wide-azimuth data using a Monte-Carlo approach is a challenge. This non-linear method that uses Simulated Annealing is known for its efficiency to compute large magnitude statics but is also characterized by a high computation cost. However, by using advanced methods like High Performance Computing (HPC), the computation cost of this not “embarrassingly parallel” algorithm can be drastically reduced. This paper demonstrates that a non-linear approach to estimate large magnitude Monte Carlo statics is now possible with a reasonable turn-around and without splitting or chunking the statics computation into several swaths.
Introduction

Conventional approaches in evaluation of surface-consistent residual static corrections are mostly based on a linear inversion scheme that uses cross-correlation functions (Ronen, 1985). Rothman (1986) and Vasudevan et al. (1991) showed that the estimation of large magnitude statics is better handled with a non-linear scheme based on a Monte-Carlo method coupled with a Simulated Annealing approach. This non-linear method shows promising results on 2D and 3D narrow-azimuth surveys but it is also characterized by an extremely high computation cost. By using advanced High Performance Computing methods, the cost in computing Monte Carlo statics can now be drastically reduced with a reasonable turn-around and without chunking. In this paper, we demonstrate the capabilities of such an approach on a large 3D wide-azimuth land dataset.

Description of the Method

Different methods are used in computing surface-consistent residual statics (Marsden, 1993). Most of these methods are based on the use of the cross-correlation functions and solution of linear system equations at a local minimum. A non-linear approach, however, that uses the Simulated Annealing concept (Kirkpatrick et al., 1983) coupled with a Monte-Carlo technique (Metropolis et al. 1953) allows computing surface-consistent residual statics at the global minimum. In this technique, the cost function usually suddenly increases (called the crystallization step) when a certain set of statics and an ad-hoc temperature is achieved (cooling schedule). Different kinds of cost functions can be used; they are based on either the stack power (Rothman, 1986) or the coherence of the neighboring CMP gathers to optimize the reflections on the final CMP stack (Vasudevan et al., 1991).

The Monte-Carlo approach we chose to implement uses a cost function that is based on the coherence of the stack with some robust criteria to stabilize the results (Le Meur and Merrer, 2006). The cooling schedule is computed as a function of the number of iterations and the initial temperature $T_0$. This temperature is determined during a pre-processing step of tens of iterations that precede the non-linear inversion. The whole process contains hundreds of simulations, each of which is randomly explored for all the shotpoints and receiver stations to avoid bias and cycle-skipping (Dahl-Jensen, 1989). For each selected shot point or receiver station, a random static value is chosen and applied to the prestack data. The cost function variation reflected the influence of this random static shift through the neighboring CMP gathers: for a positive variation the static shift is accepted, for a negative one the static acceptance depends on the metropolis criterion thus avoiding a local minimum (Vasudevan et al., 1991). Several hundred simulations are necessary to obtain a whole process. When no variation of the energy from one simulation to the next is observed, the process stops. This extremely high computation cost could be decreased only by using an advanced High Performance Computing solution.

High Performance Computing implementation

Data access is the main bottleneck when non-linear methods based on Simulated Annealing are used on large 3D wide-azimuth surveys. At each simulation step, stations must be visited in a random order. For each station, two associated collections (shots or receivers and CMP gathers) are used to compute the cost function (Figure 1). As several hundreds of simulations are necessary for convergence, this means that the input prestack dataset has to be read several hundred times in a random order! Moreover, the algorithm is not “embarrassingly parallel”. Indeed, for a given simulation step, static corrections cannot be computed independently because stations overlap each other in stack domain, ie, the algorithm cannot be massively parallelized for stations. The real challenge for this method was that it finishes in a reasonable turn-around time. We succeeded in running our implementation in a few days on large 3D wide-azimuth surveys, which is a very similar turn-around as that obtained with linear inversion methods. We made it possible to perform the method efficiently and without chunking the input data by swaths by implementing a careful fine-grained multi-core parallelism as well as some software optimization. These High Performance Computing techniques allowed us to optimize and minimize the data access time and therefore speed up the efficiency of the non-linear inversion.
Examples

We first demonstrate the efficiency of our Monte-Carlo approach on simple 2D synthetic data. We then apply the same process on a large 3D wide-azimuth survey obtaining successful results in a reasonable turn-around time.

The 2D prestack synthetic data with four flat events was generated by adding various noise levels (Figure 2a). Before starting the computation of the surface-consistent Monte-Carlo statics, surface-consistent random perturbations were applied to shot and receiver stations (within the range of ±30 ms). The goal of this test was to estimate a statics set that corrects the surface-consistently disturbed data to the initial input which had four flat interfaces. CMP stacks were computed using static values obtained at different iterations (Figures 2b-2e). The change in the stack power through the iterative non-linear process was indicated by the energy function versus the number of iterations. For up to 100 iterations, the energy level stayed low (Figure 2b) providing non-optimal static values and defocused CMP stacks. For the crystallization stage (from ~100 to ~150 iterations), a jump of the energy showed a better set of statics and stack image (Figure 2c & Figure 2d). After the crystallization phase, the energy curve reached a plateau, and the process stopped when the convergence criterion was obtained. The surface-consistent Monte-Carlo statics were applied to the randomly perturbed input data and give a stack very close to the initial four flat events stack (Figure 2e).

The second example is a large 3D wide-azimuth survey of several tens of terabytes of input prestack data containing more than 100,000 shot and receiver stations and an offset range up to 3000 m in the inline and as well as broadside directions. The input data was previously corrected with an isotropic velocity field to flatten the gathers as much as possible (see Figure 3a) and sorted in offset-azimuth order to emphasize large magnitude residual statics (up to 40 ms). The surface-consistent Monte-Carlo solution was applied on the input data to correct the large magnitude shifts for the various offset or azimuth values (Figure 3b). Some small residual jitters remained from trace to trace; they could be related to small azimuthal velocity variations. The benefit of such surface-consistent Monte-Carlo statics is seen on the stack section. On the raw data, the shallowest reflectors show a lack of continuity (Figure 4a) that has been restored after application of the Monte-Carlo surface-consistent residual statics (Figure 4b). Moreover, the continuity of the reflectors in the deeper part is enhanced and shows sharper fault delineations (Figure 4b).

Conclusions

The Monte-Carlo method coupled with a Simulated Annealing process is a very powerful approach to compute large magnitude surface-consistent residual statics. The real challenge was to perform it on large 3D wide-azimuth surveys in a reasonable time schedule without chunking the input data in several swaths. This result was made possible by using advanced High Performance Computing methods.

References


Figure 1 - HPC processing flow implementation

Figure 2: a: CMP stack of the initial data. From b. to e: The energy function and the associated CMP stack section at different stages. (b) 2 iterations; (c) 100 iterations; (d) after 150 iterations; (e) after 218 iterations (final solution).
Figure 3: (a) CMP in offset-azimuth order without s.-c. residuals statics, (b) Same CMP after the application of s.-c. Monte-Carlo statics.

Figure 4: (a) Raw Stack without s.-c. residuals statics, (b) Stack after the application of s.-c. Monte-Carlo statics solution.