

Five Dimensional Interpolation:exploring different Fourier operators

Daniel Trad

CREWES-University of Calgary

Summary

Five-Dimensional interpolation has become a very popular method to pre-condition data for migration. Many different implementations have been developed in the last decade, most of them sharing a similar dataflow and principles. In this abstract, I explore three different ways to implement the mapping between data and model in the context of Fourier transforms. These three methods are multidimensional Fast Fourier transform, Discrete Fourier Transforms, and Non-Equidistant Fast Fourier transforms. I incorporate the three operators inside the same inversion algorithm, to be able to perform a fair comparison between them.

Introduction

Most implementations for 5D interpolation work by calculating a transform by inversion with sparseness constraint and using this transform to map back to a new seismic geometry (Liu and Sacchi 2004). Applying the sparseness to the transform has an effect of filtering out sampling artifacts, because these are, by the definition of the transform, non-sparse. Looking from this high level of generality, all these different implementations are very similar. This similarity extends also to other dataflow details like overlapping five dimensional windows. Usually some version of either a conjugate gradient inversion or an anti-leakage inversion are applied to enforce the sparseness constraint. So, what do all these implementations differ on? The most common difference among them is on the transformation to connect data and models, in other words, how they represent the data and which basis functions are utilized. Some implementations use Fourier transforms, some use Hankel transforms, others curvelets, frames, Radon, Prediction Filters (Fxy) and Green functions based on modeling/migration operators.

Many implementations used in the industry are based on Fourier transforms because they are very efficient and very good in representing small details of the data. One factor that strongly influences the efficiency and flexibility of all these transforms is whether they use exact input spatial locations or some approximation achieved by multi-dimensional binning, either direct or with some intermediate interpolation. When using exact locations, these transformations become computationally expensive to invert. When using binning these transformations are efficient and can handle large window sizes and gaps, but lose precision and flexibility to adapt to narrow azimuths and long offsets. When binning 5D data from standard geometries, the resulting grids turn to be very sparse. What is interesting about Fourier transforms is that they seem to adapt well for either approach. The subject of this paper is to explore how Fourier interpolation performs in one or the other case. I compare a five-dimensional interpolation with three different operators applying the three approaches mentioned above, binning, exact locations, and approximated with intermediate approximations. Special care is taken of using the same inversion and dataflow for the three cases, leaving differences strictly confined to the differences in operators.

Operator Implementation

The three more common choices for Fourier transformations are:

- Fast Fourier transform (FFT) with multidimensional binning and grid adaptation.
- Discrete Fourier transform (DFT) using exact spatial locations
- Non-Equidistant Fast Fourier transform (NFFT) using interpolation plus FFT.

Fast Fourier transform operator

A multidimensional FFT can be applied on each frequency slice but binning is required because FFT algorithms assume data are regularly sampled. This binning is not trivial for several reasons as explained in Trad (2009, 2014). A complete 5D grid implies a data set acquired with line interval equal to group interval, which is much denser than is usually done in practice. A typical 3D survey will fill only a small percentage, typically close to 3%, leaving for the algorithm to infill 97 % of the remainder cells.

Binning along inline crossline directions is relatively straightforward, being a sensible choice the bin size, which is usually calculated to match the Fresnel zone size after migration. On the other hand, binning along offset and azimuth, or inline crossline offsets can be quite complicated, depending on the fold of the data, the shot patch pattern and the offset range. Optimal binning is critical for the FFT because coarse binning introduces time jittering that affects sparseness in the Fourier space, and creates a problem with traces that fall into the same bins. Fine binning makes the model size large and difficult to solve. For example, reducing binning by half in each direction forces interpolation to infill $2^4=16$ times as many grid cells.

An industrial application of this algorithm has to adjust the binning size along offset and azimuth to deal with these issues. For example, in this work the user pre-defines the percentage of alive traces required, usually 3% for wide azimuth data or 5% for narrow azimuth data. With this information, the algorithm calculates the optimal size of the offset/azimuth binning intervals to achieve this target. More sophisticated schemes are possible, but this simple scheme helps to adjust the grid as the data fold changes on different areas of the survey. Another problem with binning is how to adjust the aperture across dimensions such that there are no extrapolated traces, but instead new traces are created only inside the region of support of the acquisition. For example, the azimuth range can be made automatic, depending of the actual coverage and the minimum/maximum offsets along inline/crossline directions can adjust to the patch size in the data.

Discrete Fourier transform

The simplest way to calculate the Fourier transform of irregular data is to use the mathematical definition, which for 4 spatial dimensions is:

$$U_j = \sum_{x1_{min}}^{x1_{max}} \sum_{x2_{min}}^{x2_{max}} \sum_{x3_{min}}^{x3_{max}} \sum_{x4_{min}}^{x4_{max}} \exp(-ik_j x_i) \times u_i \quad (1)$$

In this equation u_i is the value of the sample at position x_i and U_k is the value of the DFT at wavenumber k_j . For 5D interpolation these are all four-dimensional vectors. Because the k axis is usually made regular, the k_j value is replaced by an index j times a Δk interval. Equation 1 contains one nested summation for each dimension, which makes this formulation very slow for 5D. Therefore, although precise and straightforward, the DFT method may not necessarily be the best choice for wide azimuth data. To speed up this algorithm we can reduce the operator to calculate only the wavenumbers inside the FK volume with physical significance, with minimum/maximum wavenumbers variable with temporal frequency. Many details are required to make this formulation practical: pre-calculate the operator or use sin-cos tables, careful alignment of memory allocations, loop vectorization, and multi-threading. Parameterization for this approach is analogous to the case of binning. Although in principle there are no sampling intervals, in practice we need them to calculate the Nyquist frequencies on each spatial dimension. The algorithm has flexibility to adjust to variable spatial ranges, making possible to adjust for example for streamer acquisitions where the azimuth range changes continuously with offsets.

Non-Equidistant Fast Fourier transform

A more efficient approach than the DFT for calculating the Fourier spectrum of irregular sampled data is to use some intermediate, fast, and localized interpolation to move the traces to the bin centers in the 4D spatial grid, and then apply a multi-dimensional FFT. We can think of the binning + FFT method described first as a nearest neighbor pre-interpolation and the NFFT as a higher order pre-interpolation. In this work, I have used the method called Non-equidistant Fast Fourier transform (NFFT, Duijndam and Schonewille, 1999), and the libraries from the Technical University of Chemnitz (Keiner et al., 2009). The irregular data are (irregularly) convolved with a smoothing window to relocate samples onto

a regular grid. A multidimensional FFT is applied to map the resulting regular (bandpass filtered) signal to the Fourier domain. The effect of the window is removed by deconvolution, which can be done in the Fourier domain since all samples are now in regular locations. One issue is that the library has some constraints for the window parameterization: it requires polynomials with an even number of parameters, and a minimum oversampling factor for these windows. This requires careful calculation of padding factors, and research is still ongoing to obtain a general an optimal parameterization.

Examples

Figure 1 shows a test for regularization where the input traces are located at irregular intervals, and the input traces are decimated four times. There is no normal moveout correction (NMO) applied, to illustrate the behavior for curved events. We see that the three operators succeed in regularizing the data and binning errors are not visible but there are some subtle differences in the spectral content for the most curved parts of the events, with the best results for the DFT. For this small example (2D data, 3D interpolation) the differences in running time are negligible. In this example, I used shot/receiver coordinates as spatial dimensions, which is feasible in some cases like 2D. In 3D geometries, usually acquisition coordinates are too far apart and commonly we use midpoints x/y, and either offset xy or offset azimuths, as in the following example.

For sparse 3D data, we have to use large interpolation windows to be able to deal with large minimum offsets. In Figure 2 we see a comparison for a real data set from Brooks, Alberta (CO₂ sequestration project). In Figures 2a and 2b we see the test window in the midpoint domain (red square) and the shots and receivers contributing to it. The original geometry was orthogonal, with 100 meters line spacing and 10 meter group spacing (5x5 bin size), but I eliminated from the input every second receiver line, making now the spacing 200m (or 40 bins). I set the output geometry to 100 meter line interval to recover the missing receiver lines. To allow each window to cover a full box I set the window size to 40 inlines x 40 crosslines. For the other dimensions, I use 20 offsets and 8 azimuths, with intervals of 50 meters and 45 degrees. Actually, these intervals are too large for the binning approach, but the implementation automatically handles this issue creating an interpolation grid of 25 meters and 22.4 degrees respectively.

Figure 2c shows a portion of the original data with all traces belonging to a removed receiver line (none of these traces were in the input). Figure 2d shows the corresponding traces created by using the FFT algorithm. We can identify the same reflectors in both, although the FFT result is a bit cleaner as expected. Figure 2e contains the traces created by the DFT algorithm. The main difference with the FFT result is stronger filtering, because of the slower convergence of the algorithm with DFT (all tests used 30 iterations). Also, because of the high memory requirement of the DFT operator the DFT Nyquist frequencies were somewhat reduced, which can explain some of the additional filtering. Figure 2f shows the result for the NFFT algorithm. There is some low frequency energy that does not exist in the original traces, which may be explained by insufficient numerical regularization at low frequencies. The computational time for the FFT approach was only 8 minutes, the DFT approach took 3 hours, the NFFT approach took 50 minutes. For this second test the best results have come from the FFT approach, but that is influenced by the large window size, and the simple structure of the data tests.

Conclusions

In this paper, we saw how three different Fourier operators behave during 5D interpolation. The first operator, standard multidimensional FFT with binning, seems to work well if binning is carefully implemented. DFT operator is more precise and flexible but computationally very expensive. This operator is a good benchmark tool to understand how approximations to exact locations affect the other operators. NFFT operator is a compromise in terms of speed of flexibility. When very large windows are required, like in the case of sparse orthogonal geometries, the efficiency and performance of the FFT operator is superior to the other two, but in the presence of complex structures or long offsets strong curvature in the events may require use of the DFT or NFFT approach.

Acknowledgements

I thank CMC Research Institutes, Inc (CMC). for access to the seismic data and CREWES sponsors for contributing to this seismic research. I also gratefully acknowledge support from NSERC (Natural Science and Engineering Research Council of Canada) through the grant CRDPJ 461179-13.

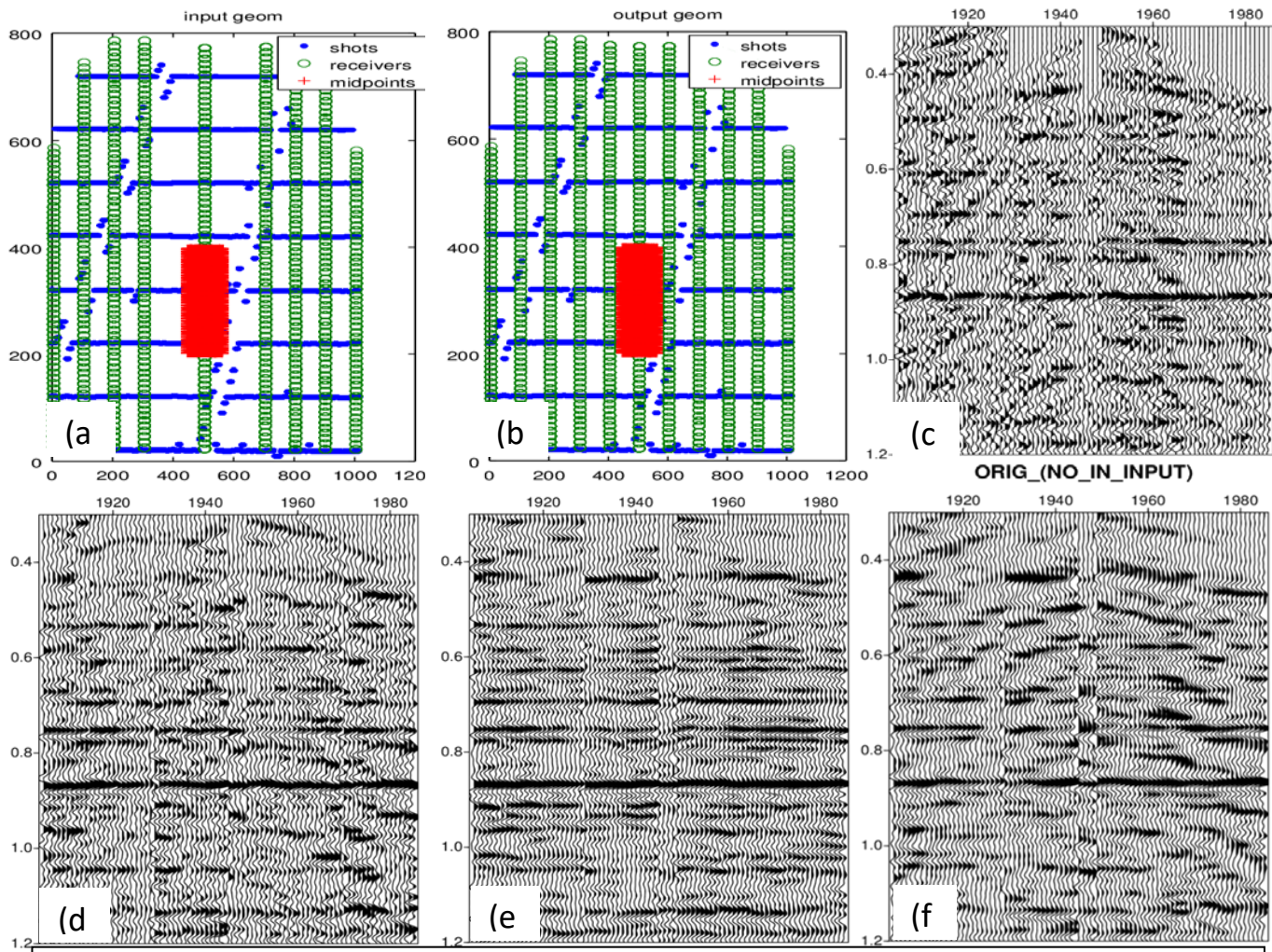
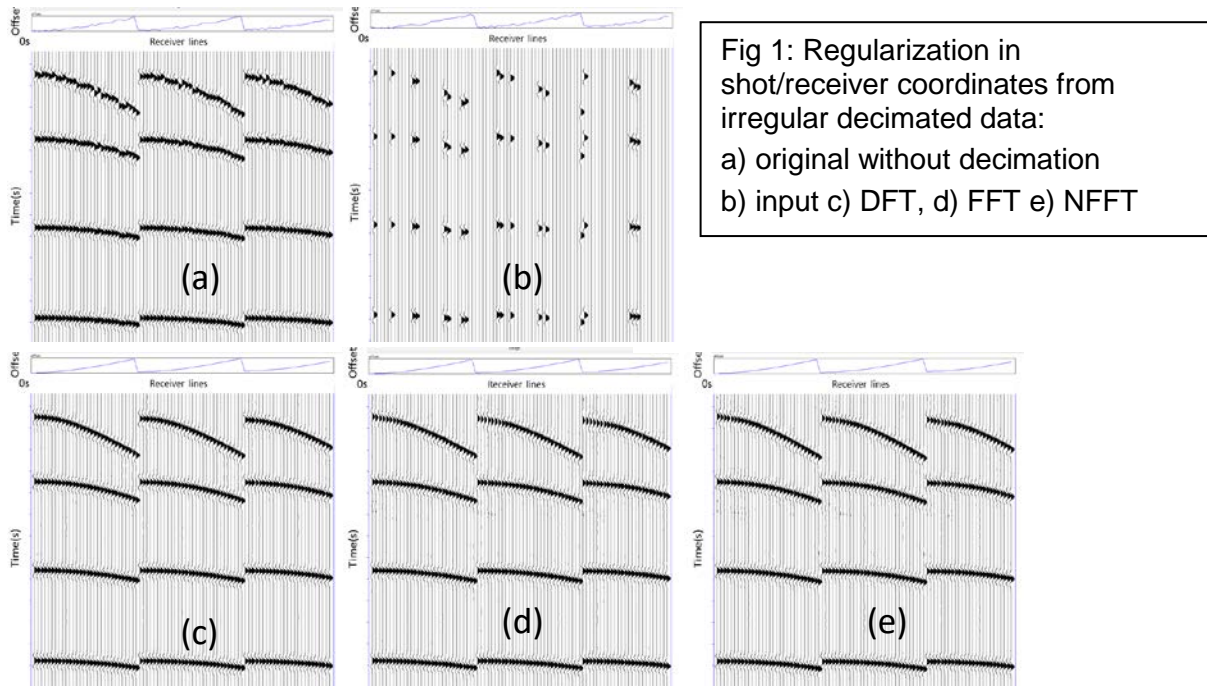


Fig 2: Interpolation of receiver lines in inline/crossline/offset/azimuth coordinates. Red square is one window in inline/crossline. a) input geometry contributing to the red window, b) output geometry, c) one missing receiver line (not in input). d) estimated from FFT, e) DFT, f) NFFT

References

- Claerbout, J., 1992, Earth sounding analysis, Processing versus inversion: Blackwell Scientific Publications, Inc.
- Duijndam, A. J. W., and Schonewille, M. A., 1999, Nonuniform fast fourier transform: *Geophysics*, 64, No. 2, 539–551.
- Keiner, J., Kunis, S., and Potts, D., 2009, Using nfft 3—a software library for various nonequispaced fast fourier transforms: *ACM Trans. Math. Softw.*, 36, No. 4, 19:1–19:30. URL <http://doi.acm.org/10.1145/1555386.1555388>
- Liu, B., and Sacchi, M. D., 2004, Minimum weighted norm interpolation of seismic records: *GEOPHYSICS*, 69, No. 6, 1560–1568, <http://dx.doi.org/10.1190/1.1836829>.
- Sacchi, M. D., and Ulrych, T. J., 1996, Estimation of the discrete fourier transform, a linear inversion approach: *GEOPHYSICS*, 61, No. 4, 1128–1136.
- Trad, D., 2009, Five-dimensional interpolation: Recovering from acquisition constraints: *GEOPHYSICS*, 74, No. 6, V123–V132.
- Trad, D., 2014, Five-dimensional interpolation: New directions and challenges: *CSEG Recorder*, March, 40–46.