Phase-encoded inversion with randomised sampling

Chris Leader and Robert Clapp

ABSTRACT

Inverse imaging and full-waveform inversion can be accelerated using phase encoding. By combining subsets of these data a series of super shots can be created, reducing the dimensionality of the problem. For certain geometries this can lead to more efficient data-space residual reduction, relative to, say, unencoded least-squares reverse time migration. Using an iteration-dependent random subset of these super-shots reduces the cost of each iteration while preserving the macroscopic convergence characteristics of conventional phase-encoding. Consequently, the cost of the system as a whole is significantly reduced while favourable residual reduction is maintained.

INTRODUCTION

Contemporary exploration seismic data sets are extremely large, to the point that data movement and I/O are significant bottlenecks in imaging and inversion. This is exacerbated by the fact that these surveys cover huge target volumes; each individual seismic source will have sampled a large subsurface domain, and there can be over 100,000 of these sources. Hence ways to accelerate how high fidelity images can be created are continually being researched.

Herein will be discussed how one can use incoherent sums to speed up imaging through inversion, by both reducing data movement and reducing the number of individual source simulation propagations performed during each iteration (Morton and Ober, 1998). This is often referred to as phase-encoding and it has been shown to increase the efficiency of seismic simulation (Ikelle et al., 2007), wave equation imaging (Romero et al., 2000) and full waveform inversion (Krebs et al. (2009); Gao et al. (2010)). Of course by combining shots in this fashion many more iterations must be performed to reduce crosstalk; nonetheless, in terms of cost and computational efficiency phase-encoding has some far-reaching benefits due to reduced dimensionality.

By using a randomly oscillating single sample encoding function the most efficient results (in terms of convergence) are observed (Krebs et al., 2009). This paper suggests that super-shots can also be encoded by using iteration-dependent randomised subsampling. If convergence properties are conserved then the overall cost will be reduced. This is similar to the work on stochastic gradients by Aravkin et al. (2012b), Aravkin et al. (2012a) and Friedlander and Schmidt (2012) but has more potential computational gains.

METHOD

Scaling, shifting and summing groups of shots together can create a series of super-shots; this is the essence of phase-encoding. Romero et al. (2000) showed, in the context of migration, that using a random encoding sequence and limiting the number of combined sources gave the best image. However, in this study it was suggested that one should not blend more than ten shots together. Krebs et al. (2009) then showed that if iteration-dependent single-sample encoding is used more efficient results are obtained; this efficiency comes from the fact the sources are no longer delayed, they are simply scaled. Moreover many more than ten shots can be combined in such a scheme and desirable convergence is still observed. In the case of fixed-spread geometry all shots can be combined into one super-shot (Leader and Almomin, 2012) and very efficient residual reduction is observed, as a function of cost.

These aforementioned methods propagate every super-shot in all iterations, however this is not necessary. By also encoding super-shots with a randomly oscillating sequence of 1, 0, and -1, it is possible to achieve similar convergence characteristics while performing fewer propagations. This has implications for the overall efficiency of the inversion due to this combined reduced dimensionality and randomised sampling.

This follows a similar idea to Friedlander and Schmidt (2012), where it is shown that by controlling the size of data subsets it is possible to observe steady convergence rates whilst using only sections of the data. This is then extended by Aravkin et al. (2012b) where they show that by using a semi-stochastic method it is possible to use as little as 40% of the data at random (at 20% the cost of a conventional approach) and still recover a useful model.

As an example, 5,000 shots could be combined into 100 super-shots. An encoding sequence for these 100 super-shots that randomly selects 1, 0 or -1 can be designed, which on average will be the equivalent of ignoring a third of the data per iteration. Because the ignored portion of the data changes between iterations (and many iterations are performed), under-sampled areas in early iterations will quickly become balanced. Furthermore, only two-thirds of these super-shots are being propagated on average, reducing cost by (roughly) a third. Using less than a third of the data is also possible.

The power of this concept is very dependent on the geometry of the problem. A dense Ocean Bottom Node (OBN) survey would be ideal, since receiver locations are stationary and a significant portion of the sources sample a high proportion of the image domain. Consequently there is a not a large aperture increase after shots are combined. A sparse streamer type survey, like Wide-Azimuth Towed Streamer (WATS) would not be a good choice. In this case distant separated sources will not share common receiver points or common midpoints, meaning there is no advantage to phase encoding.

Phase-encoding has desirable convergence properties as a function of cost in early iterations, however it can plateau at later iterations (due to the inherently noisy nature of the blended data.) A powerful option is to perform encoding at early iterations and then to use Least-Squares Reverse Time Migration (LSRTM) after a convergence threshold has been met. LSRTM can often converge to within, say, 10% of the initial residual, whereas for phase-encoded inversion the scheme might plateau at around 30%.

Algorithm 1 Conventional phase-encoding

```
while iter < n_iterations do
  while i_super_shot < n_super_shots do
    Create encoding function α of 1, −1
    while i_src < n_srcs do
      Encoded_data + = current_shot * α(i_src)
    end while
  end while
  Forward model, get data residual
  Apply adjoint, get gradient
  Forward model over gradient
  Perform model update
end while
Output final model
```

Algorithm 2 With super-shot encoding

```
while iter < n_iterations do
  Create encoding function β of 1, 0, −1
  while i_super_shot < n_super_shots do
    Create encoding function α of 1, −1
    while i_src < n_srcs do
      Encoded_data + = current_shot * α(i_src)
    end while
    Encoded_data ∗ = β(i_super_shot)
  end while
  Forward model, get data residual
  Apply adjoint, get gradient
  Forward model over gradient
  Perform model update
end while
Output final model
```

Above is some pseudo-code that demonstrates the difference between encoding the super-shots and not encoding them. One has the option of keeping the number of zeroes within the β encoding function constant or not. Choosing to do this would be purely to make normalisation easier for comparing the residual between iterations. Note this pseudo-code does not represent the way this problem was exactly implemented, it is for illustrative purposes.

SCALING THE RESIDUAL

To judge convergence a scalar measure of the data-space residual is used, as is prevalent in seismic data inversion. This is a useful metric because it allows misfit as a function of iteration (or cost) to be easily plotted and compared. However, in the case where an encoding function containing (a non-constant number of) zeroes is used it is clear normalisation is necessary.

The simplest option is just to scale the residual by the number of non-zero elements in the encoding function. This helps to smooth convergence properties, but because of the noisy nature of this process the residual can still locally increase. In fact the tests show that macroscopically this normalisation is not important, since the number of non-zero elements tends to an average of two-thirds naturally.

IMPLEMENTATION CONSIDERATIONS

An algorithm such as this, within each time-loop, massively favours computation over memory transfer; during each iteration data movement is minimised and strong bandwidth performance is observed. However, between iterations a serious disk and memory bottleneck can be created when summing and combining subsets of shots in preparation for propagating and imaging.

Choosing the two encoding functions is not iteration dependent, they are chosen using a pseudo-random number generator. Thus it is possible to prepare the encoding function and data for the next iteration while the current super-shots are being propagated and imaged. Implementing this is relatively simple by using a separate CPU thread which is spawned after the iteration has begun. This new thread creates the new encoding functions, reads and encodes the data, and stores these new data ready for the next iteration. Depending on data-size this could be resident on DRAM or written back to disk.

If a very large dataset is to be encoded into a much smaller set of super-shots this preparation time could be even longer than a full iteration of this randomly sampled phase-encoded inversion technique. In this case a separate CPU thread could prepare multiple future data combinations for the next several iterations.

RESULTS

Analysing how the introduction of zeroes to the encoding function alters convergence should be the primary goal, since the computational benefits are clear. Three methods were tested, referred to as conventional, β and β_2. For the conventional case the super-shots are not encoded (or the encoding function could be considered as a series of ones). The encoding function used for β featured a randomly varying number of non-zero elements in the sequence and β_2 fixed the number of non-zero elements at two-thirds. The residuals for the second case were scaled by the number of non-zero elements in β, as previously discussed.

This test was done in 3D using a random-boundary GPU based routine for least-squares wave-equation inversion. The tests used a fixed-spread OBN like geometry (for simplicity) and were composed of 50 inline shots across five crosslines, making 250 shots in total. The imaging domain was 10km x 5km x 4km in (x, y, z) respectively, sampled every ten metres. To measure how the introduction of zeroes can affect convergence all the data was encoded with a function randomly selecting 1, 0 or -1. The results for this can be seen in Figure 1.

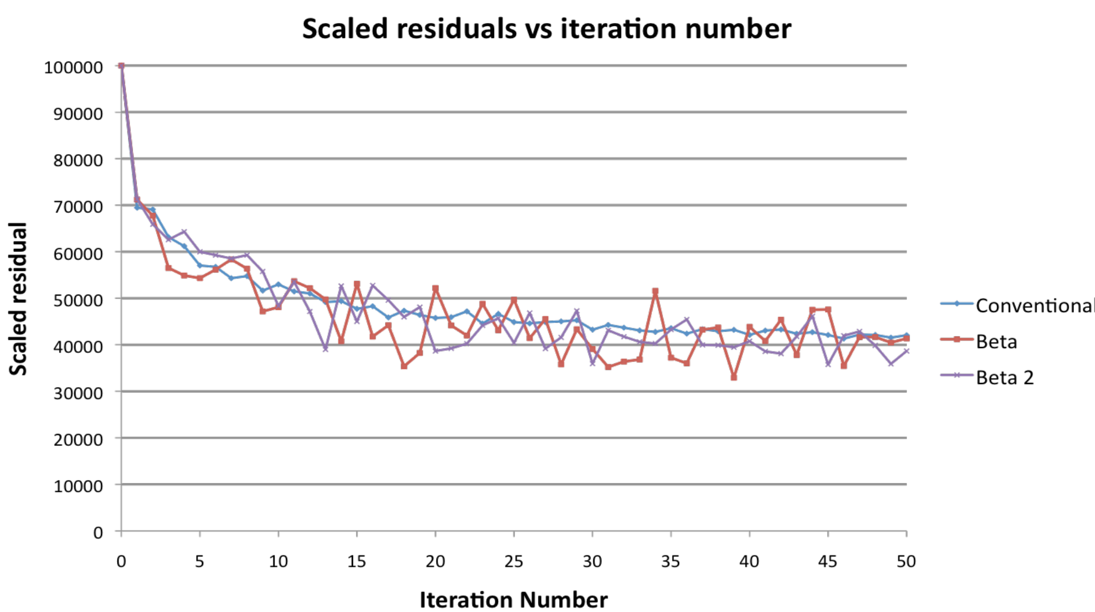

Figure 1: Convergence as a function of iteration for the three phase encoded cases. [**NR**]

It is clear that the convergence curve for the conventional case is a lot smoother, however it can occasionally increase. These localised increases are because iteration-dependent encoding creates a non-linear system. The next attribute to note is that β_2 results in a less erratic convergence curve than β, while both can see significant variation between iterations.

The general convergence trend is the same in all three cases and the residual decreases to roughly 40% of the initial value after 50 iterations. Interestingly, on average the residuals when using zeroes in the encoding are slightly lower than the conventional case, with β_2 being more consistent. If very high-fidelity imaging was desired one could move to LSRTM after these 50 iterations.

This demonstrates that when only using two-thirds of the data per iteration (randomly distributed over the domain) consistent convergence properties, relative to when the entire dataset is used, are seen. Thus by encoding super-shots it is possible save at least one-third of the iteration cost. If the horizontal axis in Figure 1 was changed to cost then the curves for β and β_2 would look even more desirable.

CONCLUSIONS

It is possible to reduce the cost of phase-encoded seismic inversion by also encoding the super-shot axis. If one samples the super-shot axis using an iteration-dependent randomised function then it is possible to use fewer super-shots per iteration - reducing dimensionality while preserving macroscopic convergence properties. Consequently the scheme becomes more efficient, as a function of cost, since the inversion scales with the number of super-shots. If the number of zero elements in a given encoding function is maintained as a constant then a more smoothly decreasing residual is seen.

FUTURE WORK

There are several aspects of this concept still under investigation. Currently tests are being done to analyse how a more complex geometry will affect these findings as a function of cost and if using less than two-thirds of the data still provides reasonable results, as suggested by Aravkin et al. (2012a).

REFERENCES

Aravkin, A., M. P. Friedlander, F. J. Herrmann, and T. van Leeuwen, 2012a, Robust inversion, dimensionality reduction, and randomized sampling: Mathematical programming, **134**, 101–125.

Aravkin, A. Y., M. P. Friedlander, and T. van Leeuwen, 2012b, Robust inversion via semistochastic dimensionality reduction: Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on, 5245–5248.

Friedlander, M. P. and M. Schmidt, 2012, Hybrid deterministic-stochastic methods for data fitting: SIAM Journal on Scientific Computing, **34**, A1380–A1405.

Gao, F., A. Atle, and P. Williamson, 2010, Full waveform inversion using deterministic source encoding: SEG Technical Program Expanded Abstracts, **29**, 1013–1017.

Ikelle, L. et al., 2007, Coding and decoding: Seismic data modeling acquisition and processing: Presented at the 2007 SEG Annual Meeting.

Krebs, J. R., J. E. Anderson, D. Hinkley, R. Neelamani, S. Lee, A. Baumstein, and M.-D. Lacasse, 2009, Fast full-wavefield seismic inversion using encoded sources: Geophysics, **74**, WCC177–WCC188.

Leader, C. and A. Almomin, 2012, How incoherent can we be? phase encoded linearised inversion with random boundaries, chapter 540, 1–5.

Morton, S. A. and C. C. Ober, 1998, Faster shot-record depth migrations using phase encoding: SEG Technical Program Expanded Abstracts, **17**, 1131–1134.

Romero, L. A., D. C. Ghiglia, C. C. Ober, and S. A. Morton, 2000, Phase encoding of shot records in prestack migration: Geophysics, **65**, 426–436.