Introduction

Full waveform inversion (FWI) is an increasingly popular algorithm for automatically improving earth models in seismic exploration. The method is incredibly computationally expensive because the earth model is built up with many iterations of forward modelling and reverse time migration (RTM). Current research is helping to reduce the number of iterations and improve the robustness of convergence, but the prohibitive cost of FWI makes running real world datasets impractical for many researchers. Furthermore, information regarding the geological region can be used to accelerate convergence. Therefore, an FWI implementation must be both high performance, and flexible. This commercial case study outlines our approach to minimizing the cost of a flexible implementation.

The key feature of our approach is that we divide the algorithm into low cost and high cost steps. Fortunately, the low cost steps in the FWI algorithm are also the ones subject to the most research. The extremely high cost full wave modelling steps are comparatively consistent between variations of the FWI algorithm. Therefore, the full wave modelling is hardware optimized and all other steps can be quickly rewritten in Python.

Method

In its most basic form, the FWI algorithm compares a measured shot record with a synthetic shot record and uses full wave modelling based on a velocity model to transform the differences between them into an error in the current velocity model. This volume of errors is used to improve the velocity model. A more detailed overview of the algorithm is given in Figures 1 and 2.

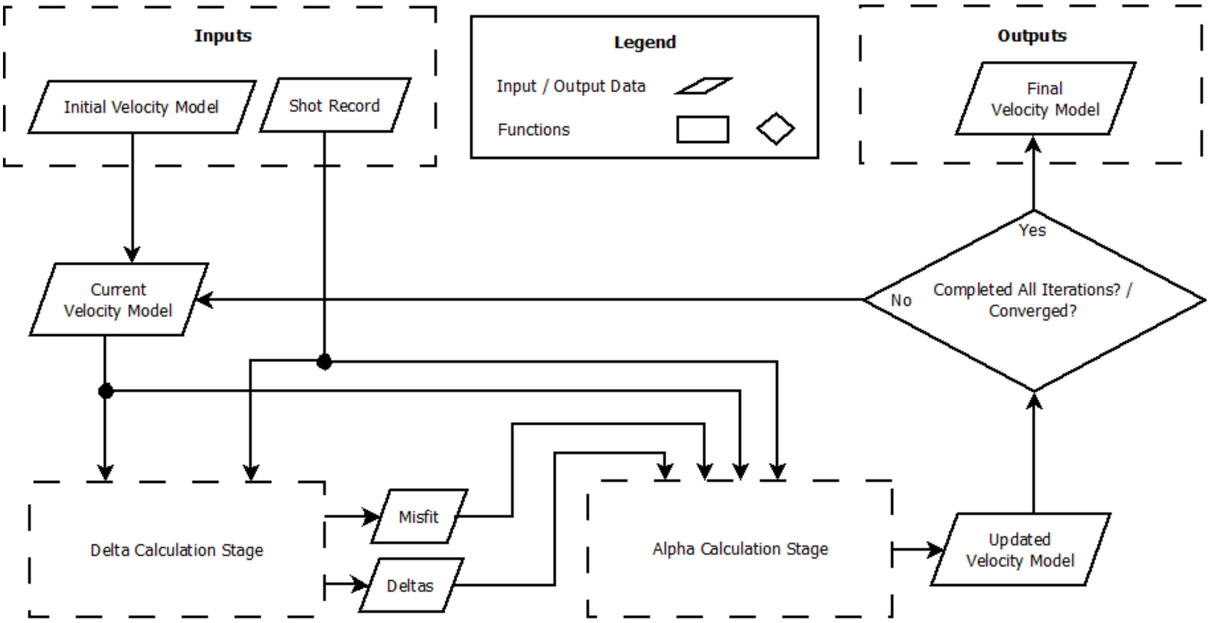

Figure 1 *The FWI algorithm can be broken up into two main stages. In the delta calculation stage, a synthetic shot record is modelled, and the difference between it and the measured shot record are used to generate a volume of changes to the velocity model (deltas) as well as a scalar summarizing the average error (misfit). In the alpha calculation stage, the deltas are scaled in order to minimize the misfit.*

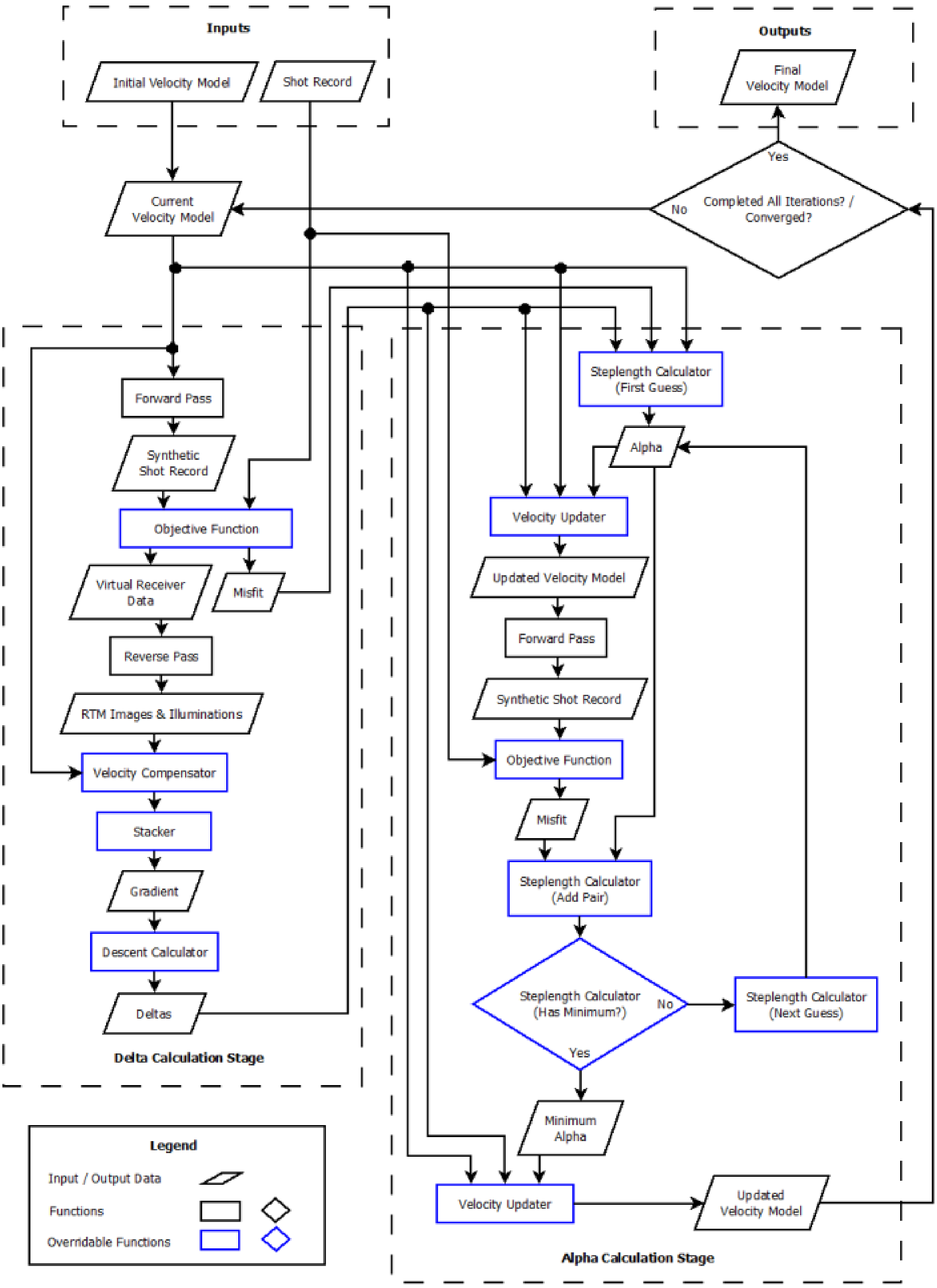

Figure 2 *Detailed flow of the FWI algorithm. Rhombuses represent intermediate data and rectangles represent functions. Blue functions can be overridden with a Python script. The only functions that cannot be overridden are the highly optimized forward and reverse passes of the full wave modelling.*

The biggest reason for the overwhelming cost of FWI is the two different loops within the workflow. The outermost loop iteratively applies changes to the velocity model. It includes the delta calculation phase, and the alpha calculation phase. The delta calculation phase has the same computational cost as an RTM on that shot record (Albeit at a relatively low frequency and likely only a subset of shots will be used). The alpha calculation iteratively repeats forward modelling runs until an appropriate scaling factor for the deltas has been found. Again, the forward modelling is by far the most expensive step and must be repeated for many shots. In most FWI algorithms, the frequency content (and cost) increases after several iterations velocity updates. Highly optimized implementations of RTM and forward modelling are commercially available, so the lowest hanging fruit for improving FWI is to reduce the number of iterations in each loop.

To harvest these computational savings, the HPC community needs to allow researchers to make the changes they want to, with the least amount of programming effort. There are many different details of the FWI workflow that can be altered within the above framework. One of the most common changes is to the objective function (Warner and Guasch 2014), but the flexibility is even more refined because it can be tailored to specific geology. An example to be presented shows the benefit of filtering the gradient of the velocity model using a priori information about the water bottom in a marine survey. Other examples will show how improvements to the steplength calculator can reduce the number of forward modelling runs in the iterations within the alpha calculation.

Conclusions

The key contribution of this work is that we show which parts of the FWI workflow can be exposed to researchers in an easy to develop programming language. Fortunately, the features that need to be the most configurable are also comparatively inexpensive. The true cost of FWI is in the RTM-like and forward modelling stages. Hardware optimization of these steps is required to keep the overall algorithm computationally tractable, but flexibility is equally important for both convergence and cost reduction. Any a priori information or algorithm improvement that reduces the total number of iterations has a significant impact on the final cost of an FWI project.

References

Warner, M. and Guasch, L. [2014] Adaptive waveform inversion – FWI without cycle skipping: Theory. *EAGE Extended Abstracts*, Amsterdam, 2014.