

# ACSE-9 Project Plan

Deborah Pelacani Cruz

Supervisors:

Dr. Mike Warner (Imperial College)

Dr. Christophe Ramananjaona (Total)

June 27, 2019

## Background Theory and Aims

Full-waveform inversion (FWI) is a local tomography inversion technique widely implemented in seismic data analysis for its capacity to produce high-resolution and high-fidelity subsurface models. As an iterative method, FWI starts from an initial model and then seeks to match its synthetic seismic waveforms, trace by trace, with an original field dataset [9].

The steps performed by FWI include a sequence of linearised local inversions and subsequent iterative updates of the starting model through the minimisation of an objective function. It starts by assuming the following forward problem:

$$\mathbf{G}(\mathbf{m}) = \mathbf{p} , \quad (1)$$

where  $\mathbf{m}$  is a column vector containing the physical parameters of a subsurface model, and  $\mathbf{p}$ , also a column vector, contains the predicted outcome of propagating a source wavelet through all points of the model. The operator  $\mathbf{G}$  is the representation of the physics acting on that subsurface. In the FWI case it is some sort of spatial and temporal projection of the wave equation that will propagate and disturb a source wavelet through the model in order to produce a seismic wave response.

The true subsurface model  $\mathbf{m}'$  can be obtained from an observed seismic wavefield  $\mathbf{d}$  by solving the following inverse problem:

$$\mathbf{m}' = \mathbf{G}^{-1}(\mathbf{d}) \quad (2)$$

There are several problems associated with this inverse formulation: (1) inexact or incomplete data, (2) data is unlikely to be noise free, (3) if  $G$  is a linear operator, its inverse is only well defined if the problem

is equidimensional, otherwise the solution is non-exact and/or non-unique and, (4) the problem is likely to be ill-posed such that small variations in the data may cause spurious variations in the model. However, the main problem associated with the inverse formulation is that  $\mathbf{G}$ , for several problems, is a non-linear operator with an inverse that is not analytically defined [9]. This means that Equation (2) cannot be directly solved and iterative methods are required instead.

Starting from an initial model  $\mathbf{m}_0$ , new models are generated by adding a perturbation  $\delta\mathbf{m}$  such that the new models are progressively closer to the true model.

$$\mathbf{m} = \mathbf{m}_0 + \delta\mathbf{m} \quad (3)$$

In FWI, we seek to find the model  $\mathbf{m}$  that produces a predicted field  $\mathbf{p}$  as close as possible to the observed data  $\mathbf{d}$ . The term '*as close as possible*' can be mathematically represented in several ways, but it should be described by a scalar objective function that values to zero when the predicted and observed data are identical. FWI can then make use of optimisation techniques in order to minimise this objective function and solve the inverse problem.

In traditional FWI, the chosen objective function is composed of the L2-norm of the residual data, i.e. the L2-norm of the difference between the predicted and observed data. It is usually combined with some method of regularisation, which serves the purpose of improving the convergence, stability and roughness of the models, or providing it with extra physical constraints. Although sophisticated FWI algorithms make use of more elaborate objective functions, for the sake of simplicity we will continue this introduction with a simple L2-norm objective function with no regularisation:

$$f(\mathbf{m}) = \frac{1}{2} \|\mathbf{G}(\mathbf{m}) - \mathbf{d}\|_2 \equiv \frac{1}{2} \|\mathbf{p} - \mathbf{d}\|_2 \equiv \frac{1}{2} \sum_{s=1}^{n_s} \sum_{r=1}^{n_r} \sum_{t=1}^{n_t} \|p_{s,r,t} - d_{s,r,t}\|_2 , \quad (4)$$

where  $n_s$ ,  $n_r$ , and  $n_t$  are the number of shots, receivers and time samples, respectively, for a given survey. It can then be shown [8] that the model perturbation is given by some approximation of the following:

$$\delta\mathbf{m} \approx -\left(\frac{\partial^2 f}{\partial \mathbf{m}^2}\right)^{-1} \frac{\partial f}{\partial \mathbf{m}} \equiv -\mathbf{H}^{-1} \nabla_{\mathbf{m}} f(\mathbf{m}) , \quad (5)$$

Ideally, if the model perturbation could be directly calculated from Equation (5), the model updated from Equation (3) would be the closest-to-true model that could be obtained with a first order approximation. In practice, however, the Hessian is a very large matrix and computing its inverse is usually not within computational reach or the matrix is not well enough posed so that its inverse exists and is stable. For this reason, the common approach is to approximate the Hessian with a matrix or scalar whose inverse is well defined and cheaper to compute, and then iteratively repeat the steps of Equations (5) and (3) such that each step will progressively provide a closer-to-true model.

The  $\nabla_{\mathbf{m}} f(\mathbf{m})$  term in Equation (5) represents the direction to which perturb the starting model and rough

approximations will likely direct the objective function towards local minima. For a robust local inversion, this gradient needs to be calculated as precisely as possible. The power of FWI lies in using the adjoint formulation [7, 8] for computing this gradient. It does so in a much cheaper way than perturbing each individual model parameter, and does not trade off for accuracy. The adjoint expression for computation of the gradient is:

$$\nabla_{\mathbf{m}} f = -\mathbf{u}^T \left( \frac{\partial \mathbf{A}}{\partial \mathbf{m}} \right)^T \delta \mathbf{u} , \quad (6)$$

where  $\mathbf{A}$  is a discretisation matrix of the wave equation in the model space, which can be as elaborate as needed,  $\mathbf{u}$  is the forward propagating wavefield defined at all spaces in the model, and  $\delta \mathbf{u}$  is the wavefield produced by back propagating the residual data through the model [9, 7]. Overall, Equation (6) tells us that the gradient of the objective function is given by the zero-lag temporal cross-correlation of the forward propagating wavefield, originated from a particular source, with the back propagating wavefield from the data residuals, weighted by the derivative of  $\mathbf{A}$  with respect to the model. A general FWI workflow is summarised in Figure 1.

FWI has been widely implemented in various fields of industry and academia. For the oil and gas sector it has been shown to be a successful method in the characterisation of reservoir properties (e.g. [6, 5]). During production optimisation, however, it is critical that not only the subsurface is characterised, but also its changes as fluids are injected and hydrocarbons removed. For this reason, the combination of FWI and timelapsing techniques has been an extensive area of research (e.g. [10, 3, 2, 4]). The aim of this project is to review and implement pre-established timelapsing techniques into FWI such that the recovered models show accurate and stable differences between baseline and monitor survey models. These timelapsing techniques are to be implemented in the FULLWAVE3D algorithm, developed by Mike Warner and his team, in order to have their impacts measured during FWI inversion.

In this project we investigate the impact of four timelapsing techniques, hereby referred to as parallel, sequential, double difference (DDWI) and joint inversion. These techniques are based on the theories of Yang et. al (2015) [10] and Maharramov et. al (2016) [4]. The first three techniques can be implemented with conventional FWI runs and do not require any intervention to the FULLWAVE3D architecture. These are schematized in Figures 2, 3 and 4, respectively. The method of joint inversion, on the other hand, requires modifications on the objective function and gradient, as it aims to invert the baseline and monitor models simultaneously. The raw objective function to be implemented for joint inversion is:

$$f(\mathbf{m}) = \alpha \|\mathbf{p}_b - \mathbf{d}_b\|_2 + \beta \|\mathbf{p}_m - \mathbf{d}_m\|_2 + \gamma \|\mathbf{W}(\mathbf{m}_m - \mathbf{m}_b)\|_2 , \quad (7)$$

where the first two terms represent the traditional FWI implementation on the baseline and monitor models, respectively, and the last term represents a regularisation that constrains the length of the changes between the two models. The constants  $\alpha$ ,  $\beta$  and  $\gamma$  are regularisation constants and define the weight of each term towards the overall functional value.  $\mathbf{W}$  is a mask matrix that restricts the model changes to the area of the model where these changes are expected.

The FULLWAVE3D architecture allows for straightforward runs of forward and inversion operations, called synthetic and tomography modes, respectively. The synthetic mode allows for the production of synthetic seismic data given a velocity model, used to compute the residual of a model with respect to a true dataset during inversion. The synthetic mode is also useful for quality checking the finalised models. The tomography mode is the one that performs FWI optimisation and produces the model updates. Figure 5 shows a sustainable workflow to sustainably make use of the software for FWI.

The software requires three main input files for its simplest run: (1) a set of observed data, used to compute the residuals at each iteration, (2) a source wavelet, used to propagate through the model and generate a predicted wavefield, and (3) a velocity model, used as the true model for a synthetic run or as a starting model for a tomography run.

Another useful utility of FULLWAVE3D is to be able to customise FWI parameters, such as the gradient computation, through external plug-ins, or EPICMods. These will be utilised when implementing the joint inversion timelapsing technique.

## **Project Plan**

The project is currently being conducted at Total's Geoscience Research Center (GRC) in Aberdeen as part of the agreements of my sponsorship. The GRC team is composed of experts in both FWI and timelapsing fields, which is an overall advantage for the success of this project. An overview of the project plan shown in Table 1.

As shown in Table 1, the first three weeks of the project were dedicated to familiarisation with the relevant software and understanding of the data provided. It was also planned that the literature review would be written throughout these weeks to develop and solidify the understanding of the theory behind the relevant topics. These tasks will be particularly relevant when performing the joint inversion, as the objective function and gradient calculations will have to be derived by hand and it must be implemented in an unusual way in FULLWAVE3D.

During the third week, in particular, it was aimed that the first inversion models would be produced and spurious behaviours identified and dealt with. These initial tests would then ensure stability and consistency for the next models. The first valid models will then serve to implement the parallel and sequential timelapsing techniques.

The fourth week assumes that the parallel and sequential models from week 3 have been tested and validated. Then the week will be dedicated to implementing the third timelapsing method of double difference waveform inversion.

Throughout weeks 5 to 7, the joint timelapse method will be designed to fit the FULLWAVE3D code and implemented on the baseline and monitor datasets.

Weeks 8 and 9 will be dedicated to running tests on different models, possibly adding features such as

regularisation and anisotropy, and study how they affect the methods already implemented. If time is available, the methods might also be tested on real datasets as opposed to synthetic. Week 10 has been reserved as a buffer in order to allow time to resolving unforeseen problems.

The plan also reserves time throughout the duration of the project for writing the technical report. Weeks 11 and 12 will be dedicated to finalising it during the weekdays.

## Coding Organisation

This project was designed for the application and manipulation of existing software rather than pure software development. The necessary coding for its completion consists mainly of stand-alone python/bash code for data pre-processing and visualisation. The data is stored in SEP and SEG-Y formats for seismic data and require customised tools for reading and manipulation. Patches have also to be developed for the FULLWAVE3D algorithm during the joint inversion implementation without interfering with its main architecture. The following tasks are to be completed with stand-alone code:

### Data Analysis and Visualisation Tasks

- Load and read SEG-Y files as *numpy* arrays, retrieving useful information such as number of traces, number of shots, number of time samples, etc.;
- Normalise data amplitude to mean of zero and standard deviation of one.
- Plot SEG-Y data as amplitude maps, with options of capping low and high amplitudes (Figure 6);
- Create and plot interleaving datasets of two independent SEG-Y files of equal dimensions, with the option of choosing the number of interleaving blocks (Figure 7). Useful for quality checking and comparison between observed and predicted data;
- Plot SEG-Y data as wiggle traces, with the option of scaling the wiggle amplitudes and skip traces in the x and y directions (Figure 8);
- Plot interleaving wiggle traces of two independent SEG-Y files of equal dimensions, also with the option of scaling the amplitudes and skip traces (Figure 9). Useful for quality checking and identification of cycle-skipping phenomena;
- Create and plot the difference in amplitude between two independent SEG-Y files of equal dimensions, with the option of jointly normalising data amplitudes between 0 and 1 for a straightforward comparison (Figure 10). Feature must also be able to save the difference into a new SEG-Y file to be used as input for DDWI;
- Create and save the stitching of two independent SEG-Y files of equal dimensions with padding in the centre (Figure 11). This will be used as the input and gradient calculation during the Joint Inversion timelapsing method;

- Read and plot a velocity model, saved also as a SEG-Y file, transforming grid points into depth and horizontal distances (Figure 12);
- Compute the root mean square error of a predicted velocity model against a true velocity model;
- Plot the spatial difference between a predicted and a true velocity model;
- Design a *saving* function that reads a project name and is able to produce all plots above in a consistent manner and save them in a specified folder with an unique filename.
- Compute a matching Wiener filter between a set of true and predicted stacked traces. Useful for retrieving a source wavelet that is appropriate for the particular inversion problem.

## FULLWAVE3D

- Correction, resampling and reformatting of data with bash scripts and Total's internal command-line based software, as required for input on FULLWAVE3D (Figure 13);
- Simple adjustment of FULLWAVE3D forward and inversion parameters with FORTRAN language;
- Use of EPICMods for customising the FWI functional during joint inversion timelapsing;
- Use of EPICMods for adjusting the calculation of the functional's gradient during joint inversion timelapsing.

## Progress to Date

The progress on the project so far is also shown on Table 1. Further description on the tasks performed during these first three weeks are presented below:

### Week 1:

- Familiarised myself with the structure of the FULLWAVE3D algorithm at Imperial College, producing synthetic data on a small velocity model and inverting a small 2D streamer dataset;
- Learned how to identify common mistakes that would cause FULLWAVE3D to behave unexpectedly;
- Implemented a full inversion run on Imperial's *cx1* cluster.

### Week 2:

- Familiarised myself Total's internal software including: SEP to SEG-Y file conversion, SEG-Y visualisation and resampling tools;

- Developed a customised SEG-Y file manipulation package on python, which would later be useful for the synthetics and inversion quality checking;
- Visualised, corrected and manipulated data for input on FULLWAVE3D;
- Studied and adjusted the FULLWAVE3D input parameters so that preliminary synthetic and inversion results could be produced;
- With the first synthetic data produced, identified problems with the observed data and source wavelet frequencies. Designed a plan to correct it;

### Week 3:

- Continued developing the customised python package, adding extra manipulation features and optional parameters to functions in order to increase their flexibility and functionality;
- Filtered the observed dataset and the source wavelet in attempt to produce better synthetic data on the initial model prior inversion;
- Performed quality checking on the synthetic data to ensure correct geometry set up, enough absorbing boundaries and minimal cycle-skipping;
- Learned how to submit jobs on the local computer cluster;
- With all FULLWAVE3D parameters tested and checked on synthetic runs, submitted the first inversion job on the local cluster;
- Increased the resolution of the inversion model in attempt to produce better results;
- Added an animation feature to the python package to visualise the evolution of the inversion models as they iterate through FULLWAVE3D. This will help to identify spurious behaviour on the models and assess the validity of the chosen inversion parameters.

### **Independence and Initiative**

Independence was shown on my part from the very start of this project. I familiarised myself with the FULLWAVE3D code, I was given guidance by two PhD students who had restricted time. Most of the familiarisation and understanding of the software architecture was done by testing the outcomes for a range of inputs and extended reading of the documentation.

In my first day of work at Total, I held a meeting to present to the GRC team what the aim of the project was, what I was able to offer and how they could help me achieve my goal. This was essential to design an effective plan for the completion of this project. Every member of team was able to give ideas and identify which parts of the project they could contribute to.

Apart from the extended support from the GRC team, all the work was carried out independently. I would only seek for help and second opinions on topics that I am not so acquainted with, such as seismic data resampling and filtering, bash and linux environments, etc.

Whilst performing the first inversion, we identified a problem with the source wavelet, to which I proposed implementing a matching filter technique used in current FWI research.

The customised SEG-Y file manipulation package I developed was also very important in the quality checking of the produced models, allowing us to identify unwanted behaviours during the runs and tracing back their origins. The package was developed using some tools from the *Segyio* [1] library, which I learned about at the EAGE conference earlier this month.

## References

- [1] Equinor. *Segyio* 1.5.3, 2018.
- [2] M. Maharramov and B. Biondi. Robust Simultaneous Time-lapse Full-waveform Inversion with Total-variation Regularization of Model Difference. *77th EAGE Conference and Exhibition 2015*, (June):2–7, 2015.
- [3] Musa Maharramov and Biondo Biondi. Joint full-waveform inversion of time-lapse seismic data sets. (October):954–959, 2014.
- [4] Musa Maharramov, Biondo Biondi, and Mark Meadows. Time-lapse inverse theory with applications. *Geophysics*, 81(6), 2016.
- [5] Andrew Ratcliffe, Caroline Win, Vetle Vinje, Graham Conroy, Mike Warner, Adrian Umpleby, Ivan Stekl, Tenice Nangoo, and Alexandre Bertrand. Full waveform inversion: A North Sea OBC Case Study. In *SEG Technical Program Expanded Abstracts 2011*, pages 2384–2388. Society of Exploration Geophysicists, 2011.
- [6] L. Sirgue, O. I. Barkved, J. P. Van Gestel, O. J. Askim, and J. H. Kommedal. 3D Waveform Inversion on Valhall Wide-azimuth OBC. In *71st EAGE Conference and Exhibition incorporating SPE EUROPEC 2009*, 2009.
- [7] Albert Tarantola. Inversion of seismic reflection data in the acoustic approximation. *Geophysics*, 49(8):1259–1266, 1984.
- [8] J. Virieux and S. Operto. An overview of full-waveform inversion in exploration geophysics. *Geophysics*, 74(6):WCC1–WCC26, 11 2009.
- [9] Michael Warner, Andrew Ratcliffe, Tenice Nangoo, Joanna Morgan, Adrian Umpleby, Nikhil Shah, Vetle Vinje, Ivan Štekl, Llus Guasch, Caroline Win, Graham Conroy, and Alexandre Bertrand. Anisotropic 3D full-waveform inversion. *Geophysics*, 78, 2013.
- [10] Di Yang, Mark Meadows, Phill Inderwiesen, Jorge Landa, Alison Malcolm, and Michael Fehler. Double-difference waveform inversion: Feasibility and robustness study with pressure data. *Geophysics*, 80(6), 2015.

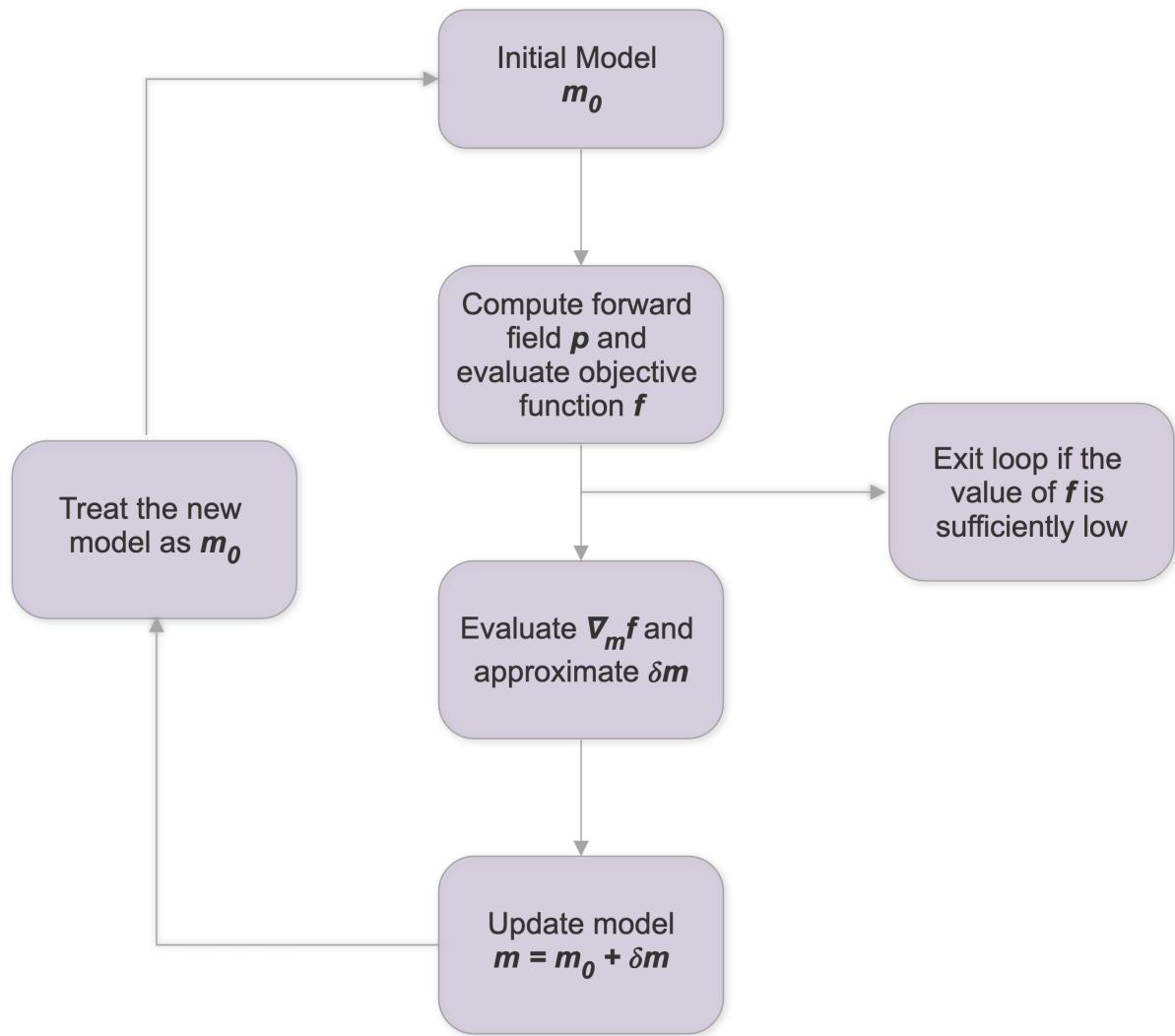


Figure 1: Diagram of a local inversion workflow.

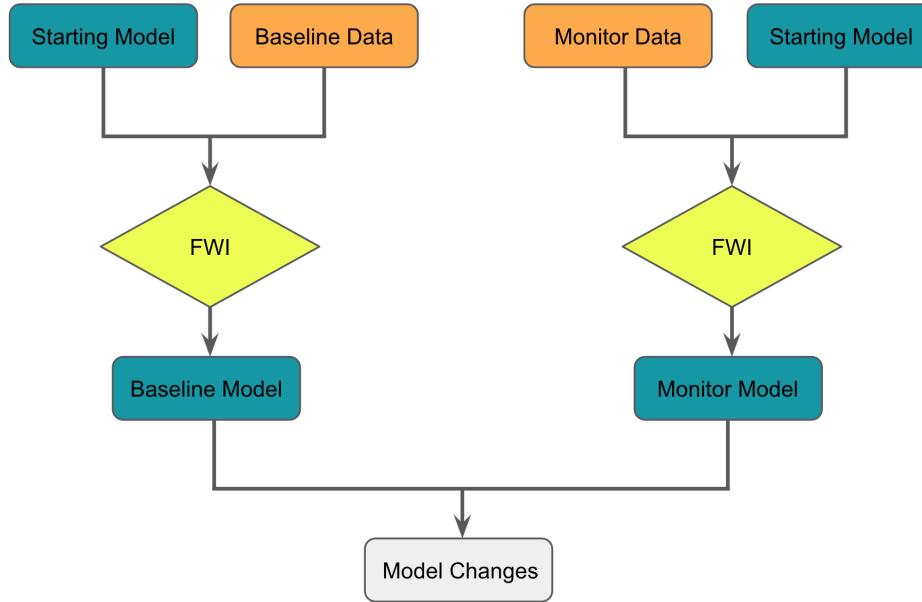


Figure 2: Schematic workflow of Parallel Inversion, adapted from Yang et. al (2015) [10]. Here the baseline and monitor data are inverted independently from the same starting model.

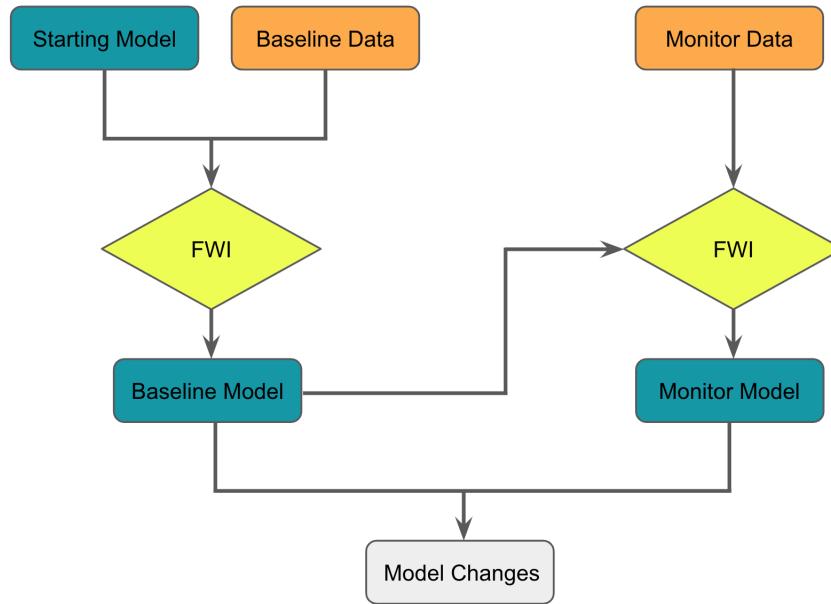


Figure 3: Schematic workflow of Sequential Inversion, adapted from Yang et. al (2015) [10]. Here the baseline and monitor data are inverted independently, but the starting model for the monitor inversion is the final model from the baseline inversion.

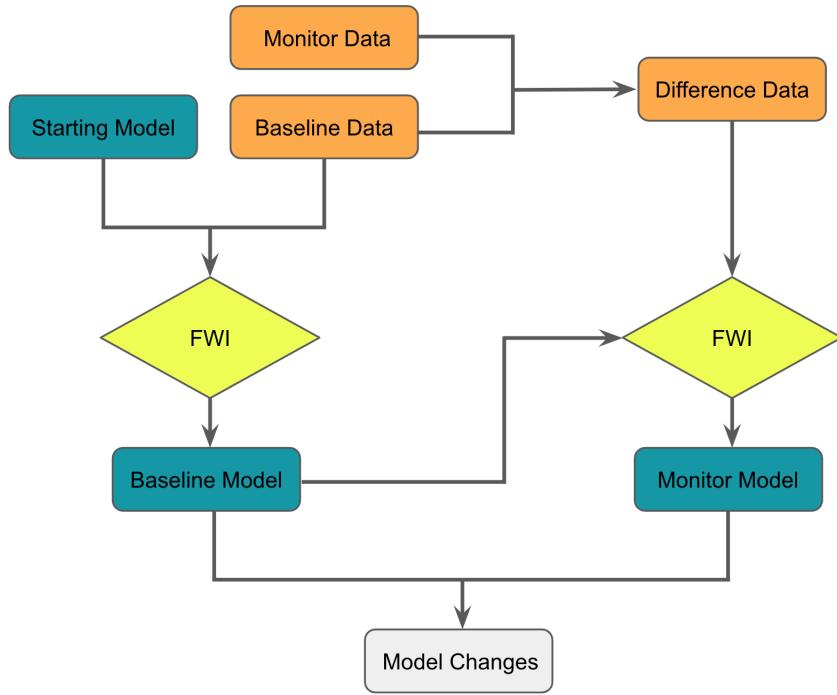


Figure 4: Schematic workflow of Double Difference Waveform Inversion, adapted from Yang et. al (2015) [10]. Here the difference between the monitor and baseline datasets is inverted with the baseline model as the starting model.

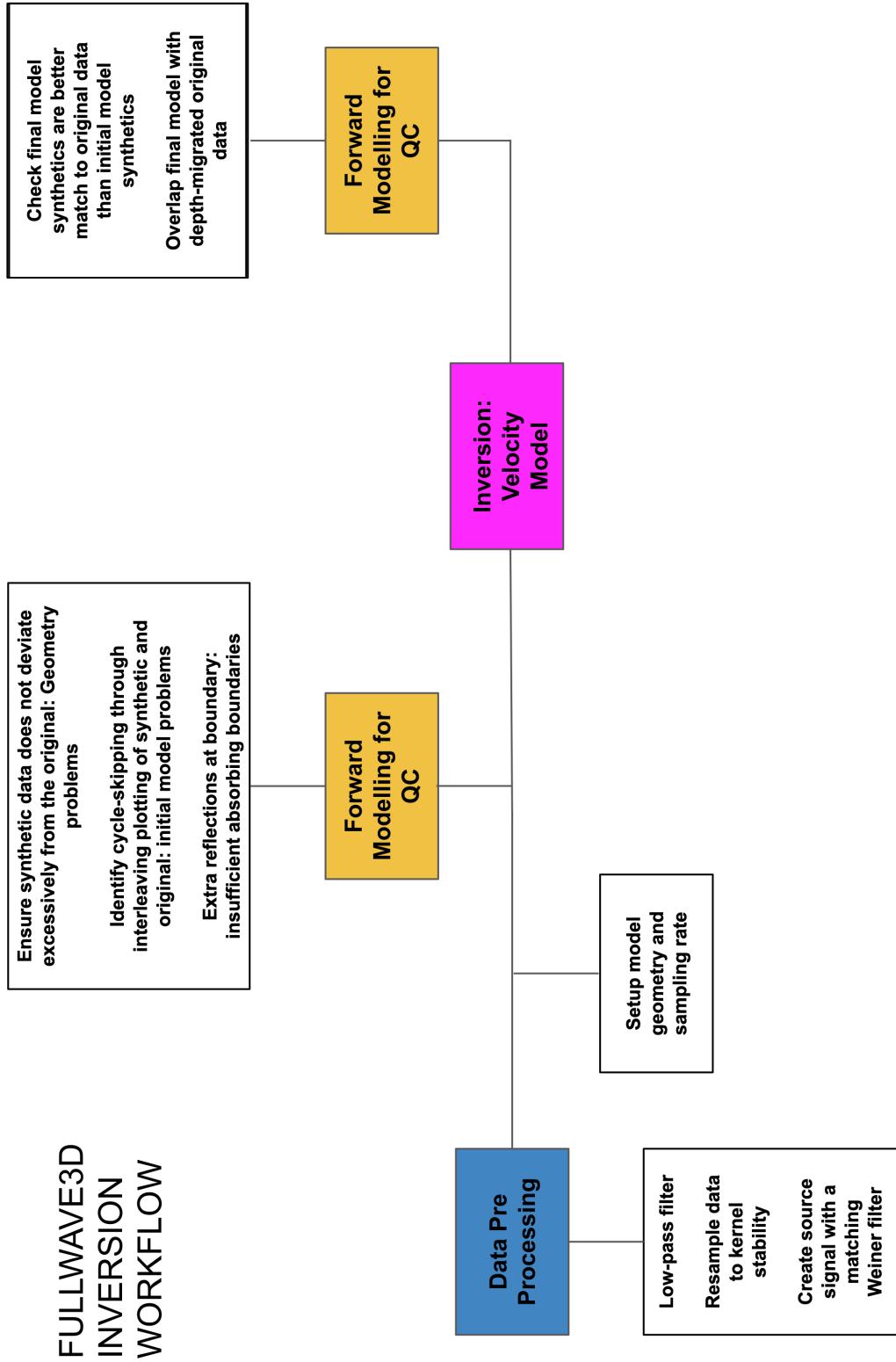


Figure 5: Schematic workflow for running a sustainable FWI inversion on FULLWAVE3D. Forward modelling is performed prior and after inversion for quality checking of the models and geometry setup. The inversion workflow (pink box) is the one described in Figure 1, which include a series of forward and inversion steps for the optimisation of an initial model.

SECTION / WEEK START	10/06	17/06	24/06	01/07	08/07	15/07	22/07	29/07	05/08	12/08	19/08	26/08
FWI / FULLWAVE3D Familiarisation	1	2	3	4	5	6	7	8	9	10	11	12
Familiarisation with Total's Internal Tools	✓											
Data Correction / Manipulation	✓	✓										
Implement Parallel and Sequential FWI		□										
Implement Double Difference FWI			□									
Design Joint FWI Implementation				□								
Implement Joint FWI					□	□						
Test Methods on Different Models						□	□					
Report Writing: Project Plan			✓									
Report Writing: Literature Review	✓	✓	□									
Report Writing: Data			□									
Report Writing: Fullwave3D implementation				□								
Report Writing: Results Discussion					□	□	□					
Report Writing: Conclusion and Limitations										□		
Report Writing: Abstract										□		
Report Writing: Formatting, Appendix and References										□		
Report Writing: Revision and Grammar/Vocab Check										□		
AD HOC/BUFFER									□	□	□	

Table 1: Overview of project plan over summer term weeks.

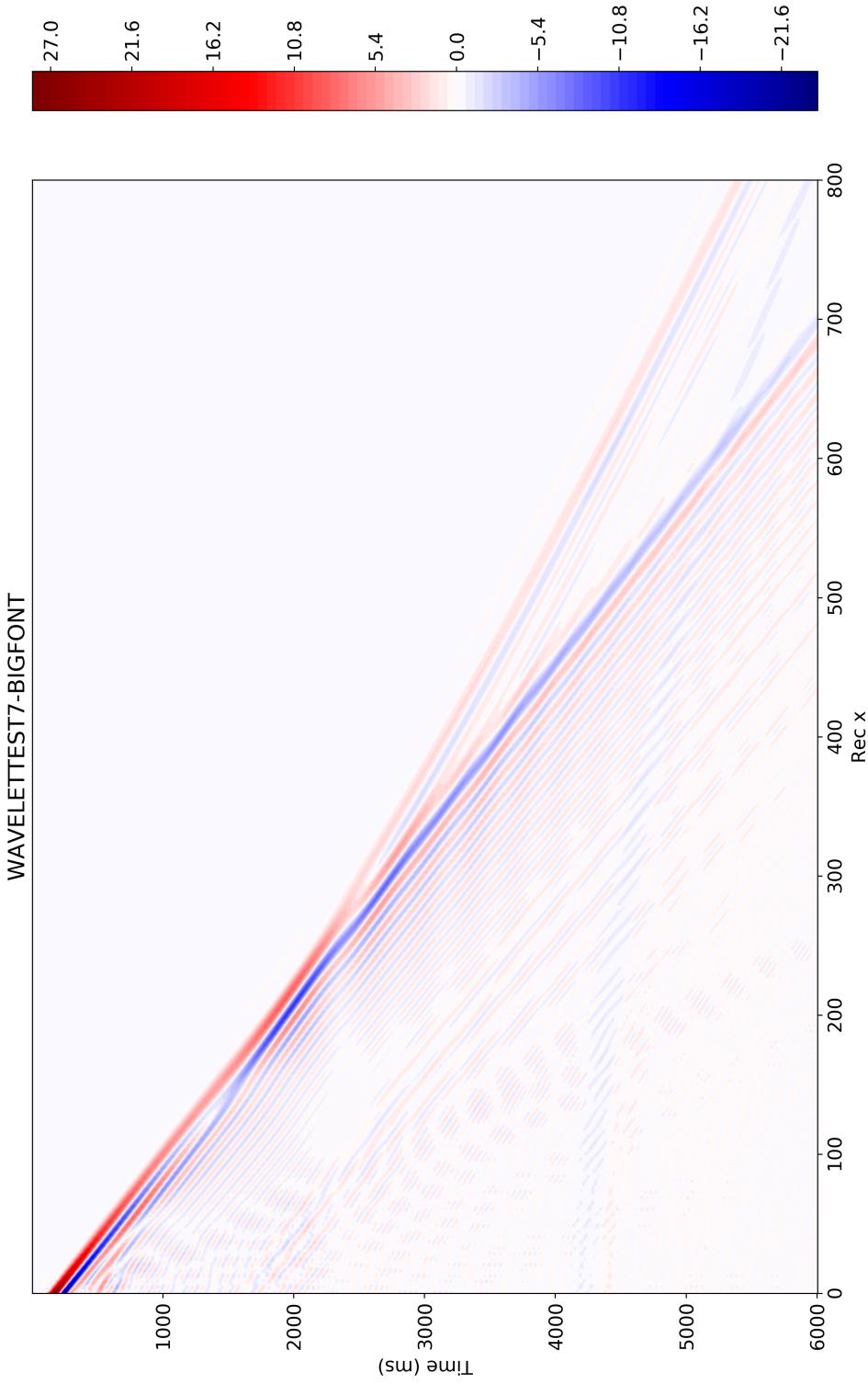


Figure 6: Amplitude plot of the synthetic seismic data produced from the FULLWAVE3D algorithm on an initial model. The data refers to the first shot of the set geometry and the x-axis represents each receiver. Amplitudes are normalised to mean of 0 and standard deviation of 1.

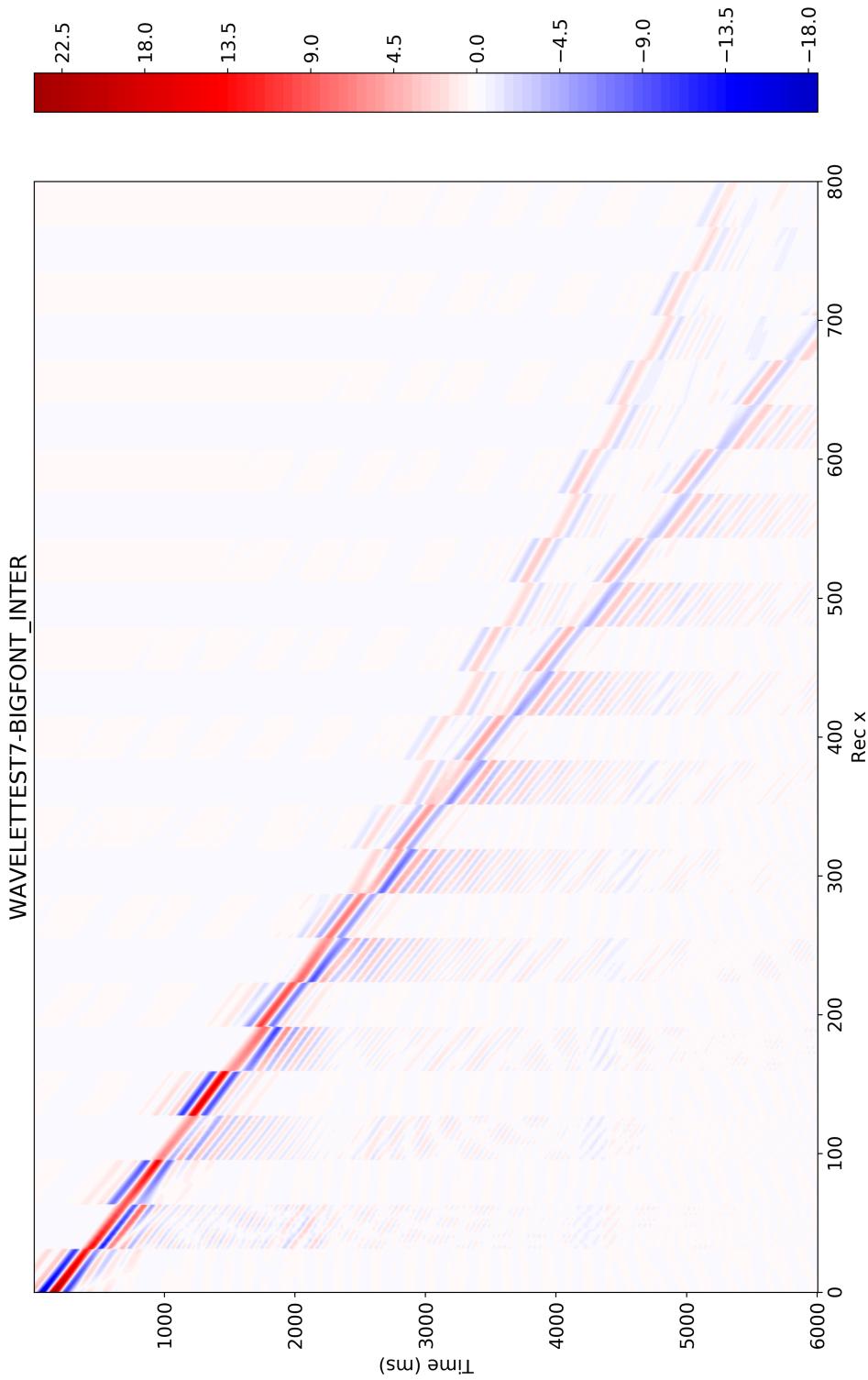


Figure 7: Interleaving amplitude plot of 25 blocks showing the synthetic data from Figure 6 on the right and its correspondent true data on the left.  
We see here that the first arrivals are within half-a-cycle of one another up until receiver 300.

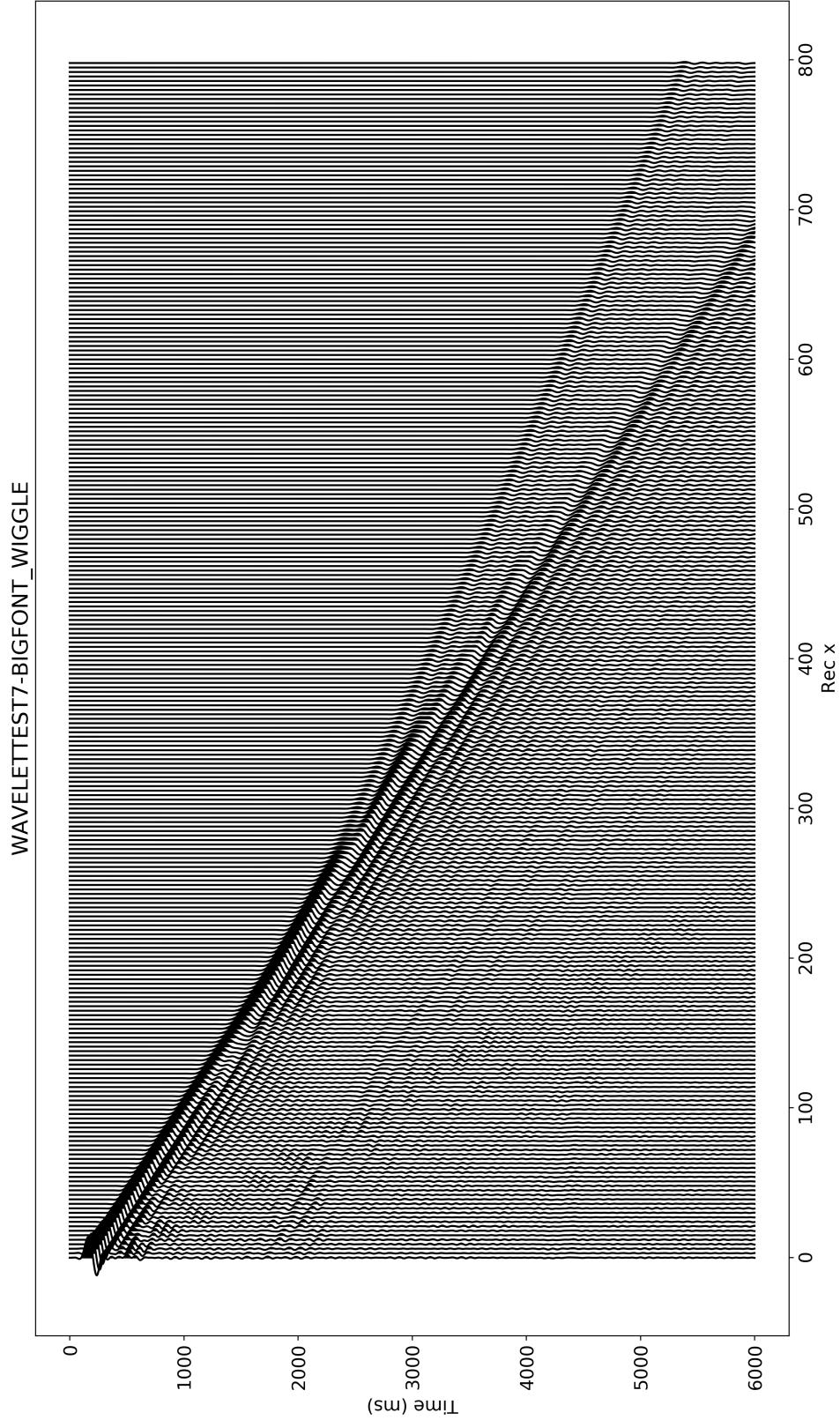


Figure 8: Wiggle trace plot of the synthetic data from Figure 6 with every 2 traces skipped in the x-direction. Amplitudes are scaled by a factor of 0.5.

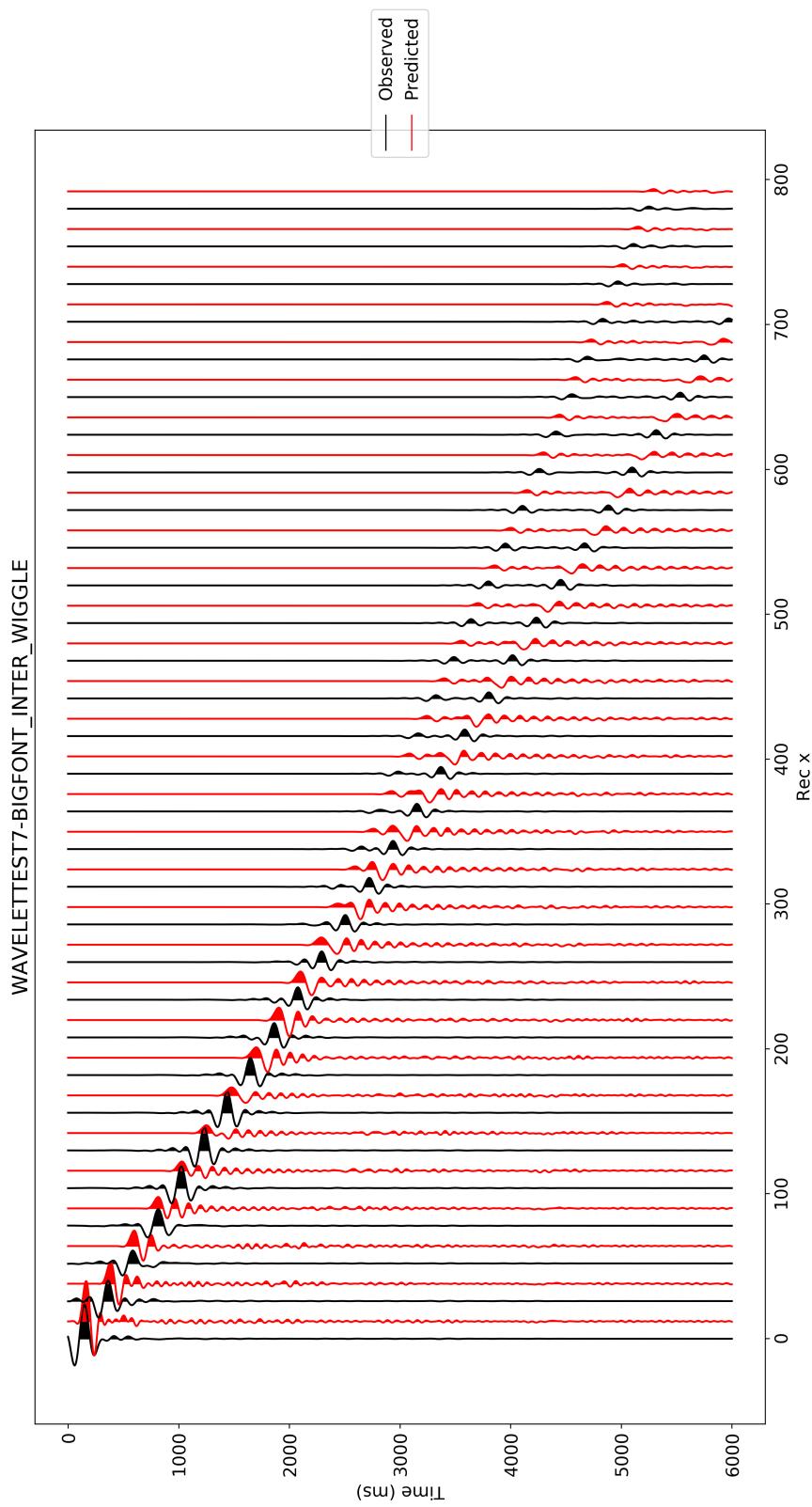


Figure 9: Interleaving wiggle trace plot showing the synthetic data from Figure 6 in red and its correspondent true data in black. Plot skips every other 25 traces in the x-direction and amplitude is not scaled. We observe more closely here the incoherence between the datasets around receiver 300, also shown in Figure 7

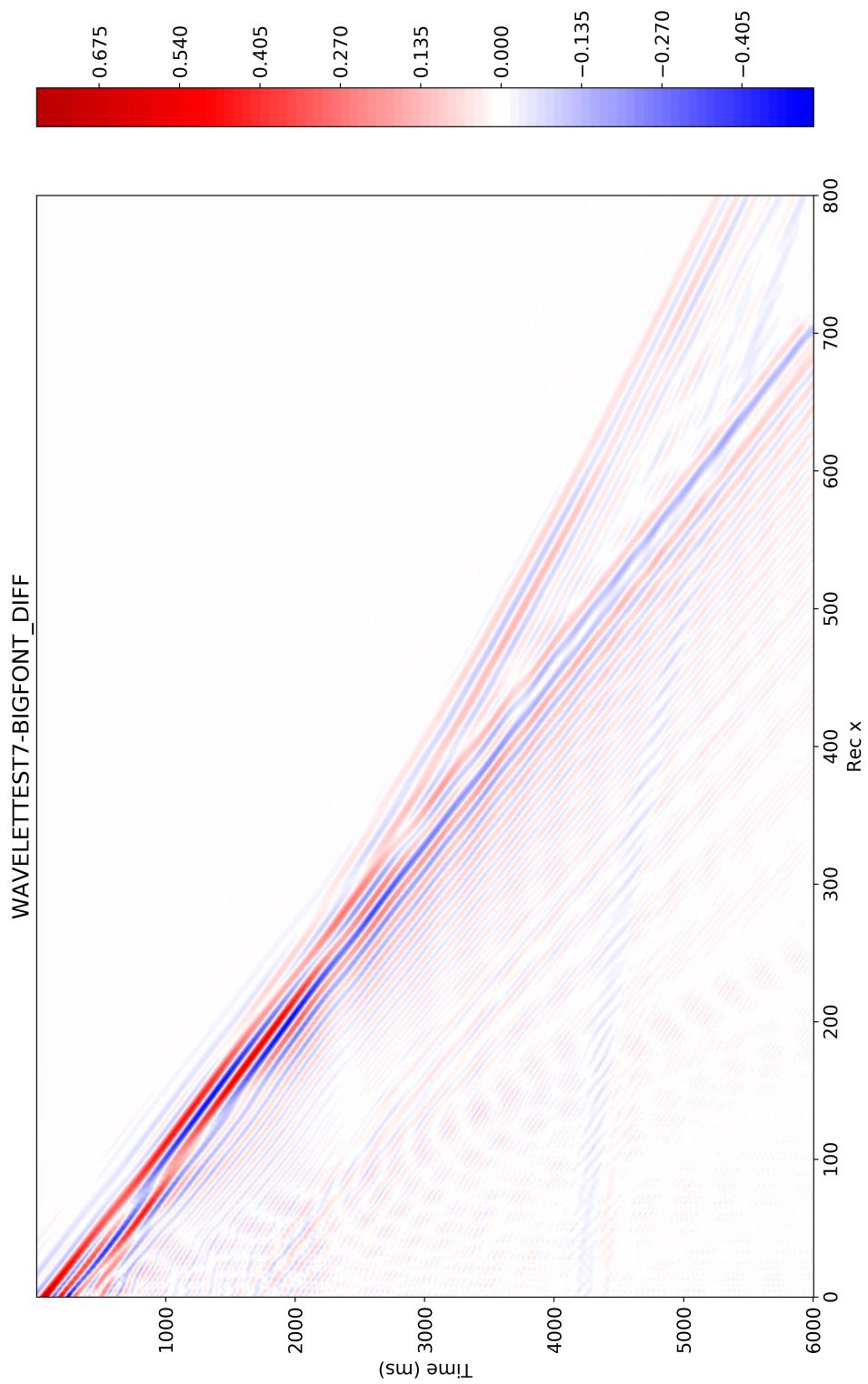


Figure 10: Amplitude plot of the OBSERVED - PREDICTED datasets difference, normalised to values between 0 and 1.

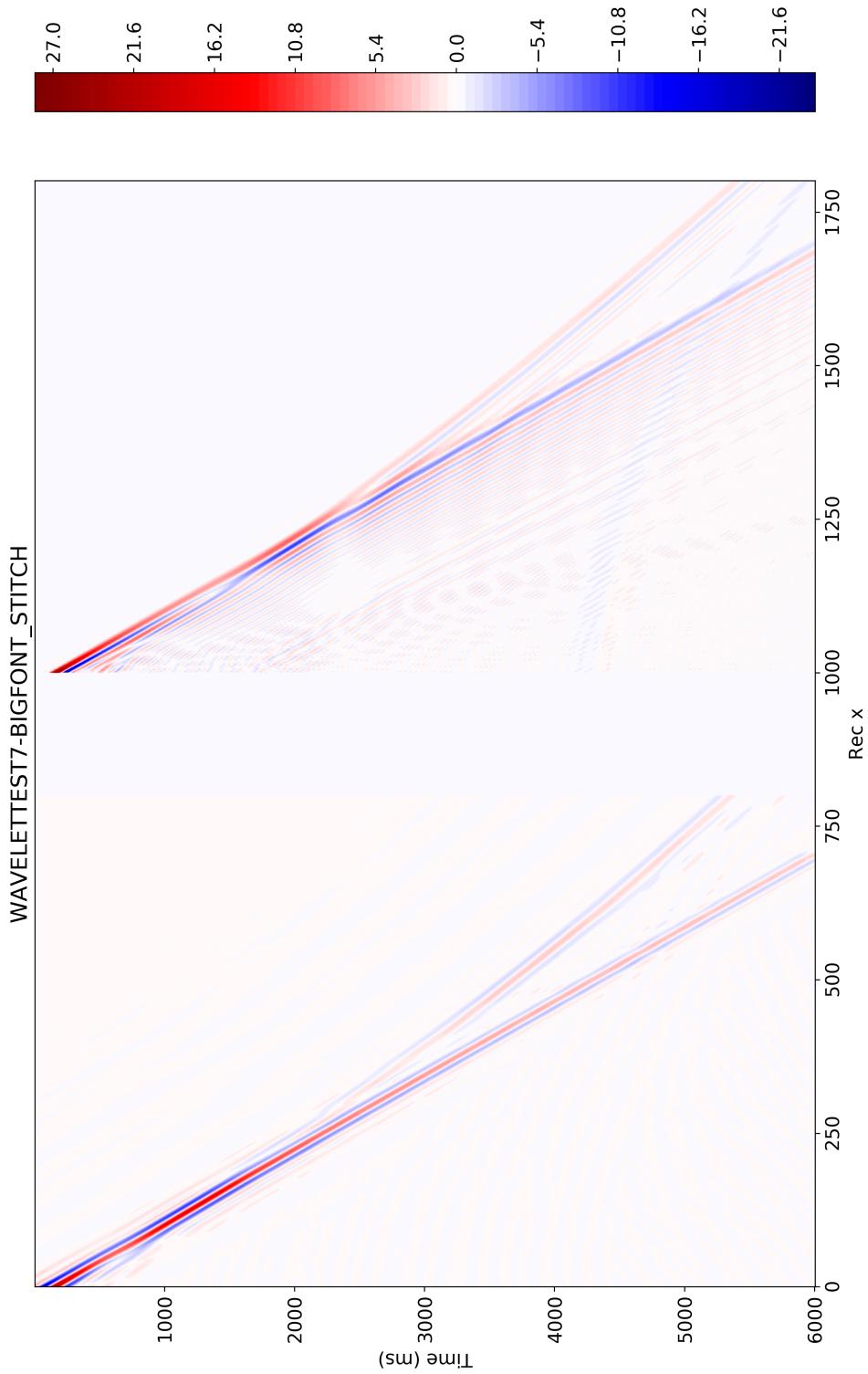


Figure 11: Amplitude plot of the stitching of the observed dataset (left) and the correspondent synthetic shown in Figure 6 (right). Inner padding of 200 receivers with null data.

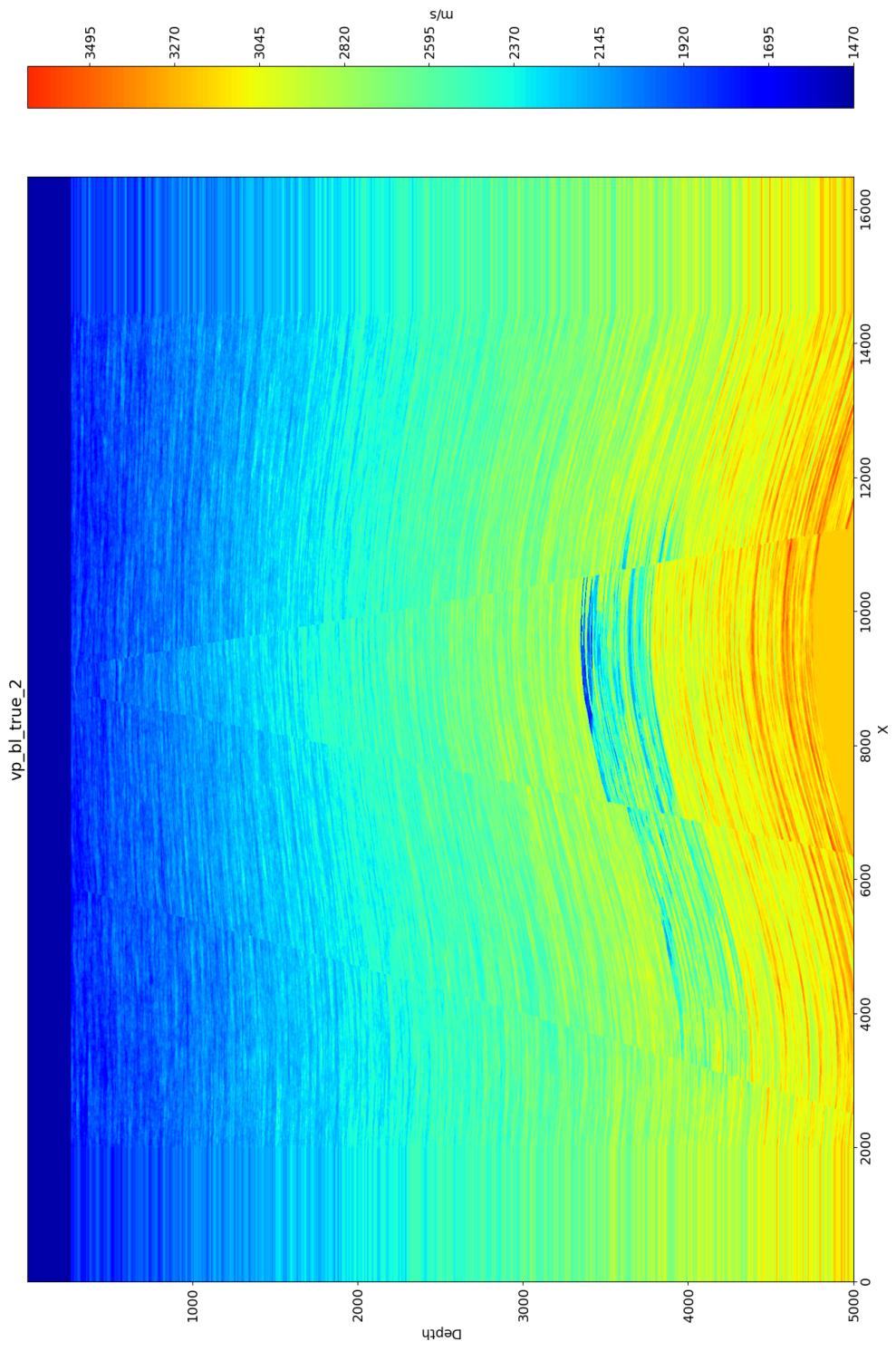


Figure 12: True P-wave velocity model, courtesy of Total. Both axes are in meters and are resolved for a grid composed of 2m x 2m cells.

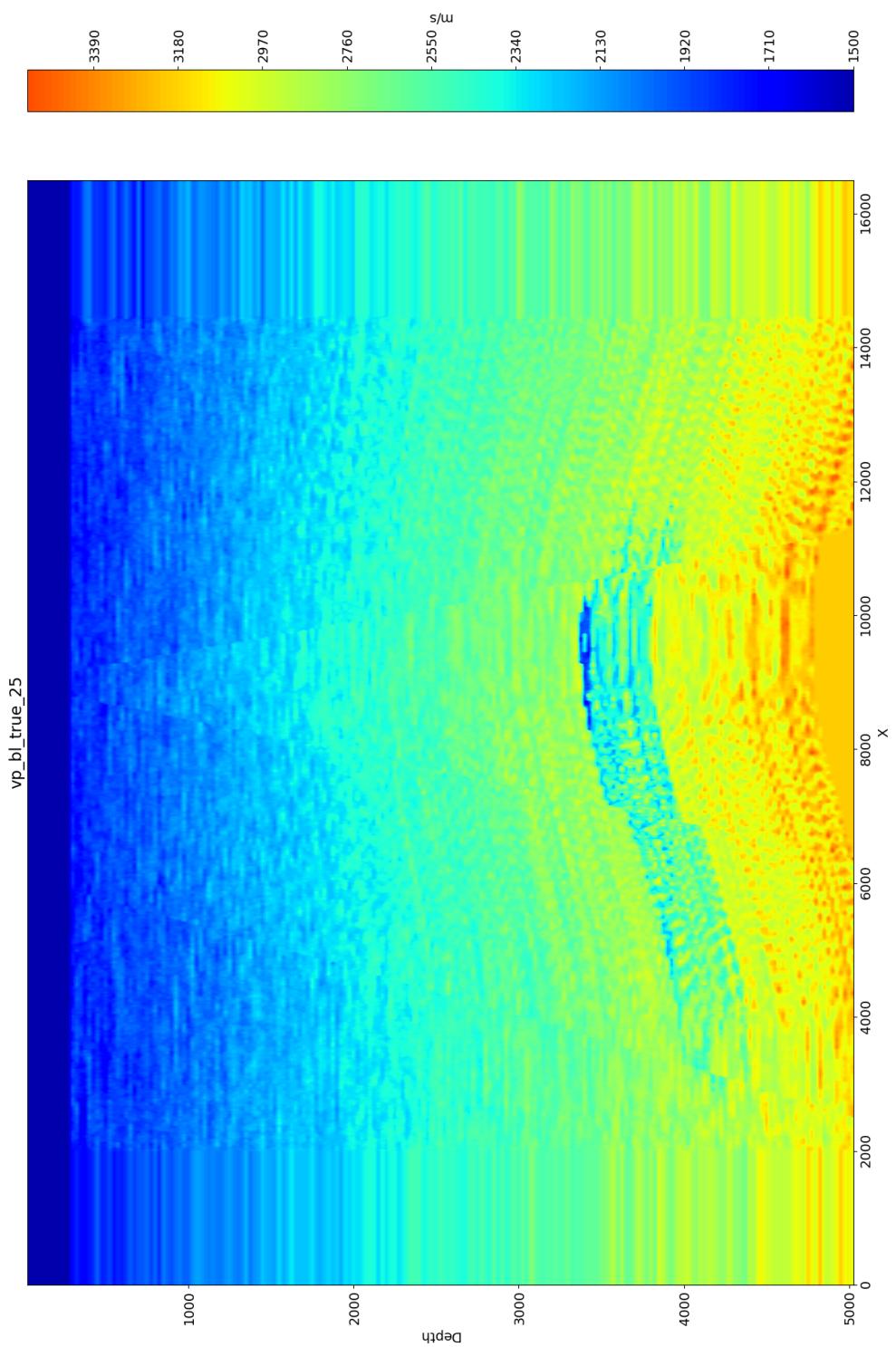


Figure 13: True model from Figure 12 downsampled to  $25\text{m} \times 25\text{m}$  cells using Total's internal software.