Imperial College London

COURSEWORK

IMPERIAL COLLEGE LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

Placement Interim Report: Schlumberger

Author:
Keidi Kapllani (CID: 00946701)

Supervisor:
Thomas Clarke

Date: July 6, 2018

Contents

1 Introduction

1.1 The company

As part of my 3^{rd} Year Placement I am working as a Software Engineer for WesternGeco, a subsidiary of Schlumberger LTD group, a company which specializes in oilfield services with offices in Gatwick. I am assigned to the REMS (Research, Engineering, Manufacturing, and Sustaining) team, composed of software developers and geophysicists responsible for the development and maintenance of a Geophysics Data Processing software called Omega.

Schlumberger is the world's leading provider of technology for reservoir characterization, drilling, production, and processing to the oil and gas industry. Working in more than 85 countries and employing approximately 100,000 people who represent over 140 nationalities, Schlumberger supplies the industry's most comprehensive range of products and services. Some notable clients include British Petroleum, Exxon Mobil and Qatar Petroleum. [1]

1.2 The team

The WesternGeco REMS group aside from publishing breakthrough internal and external papers, are responsible for implementing most of the research papers in OMEGA, a geophysics data processing platform which integrates comprehensive work-flows and advanced algorithms with leading science and scalable processing. The algorithms and components of Omega are then used in conjunction with the PETREL software, which is geophysical interpretation and characterization platform, by the SIS (Schlumberger Integrated Solutions) team for reservoir characterization and modeling.[2] My assigned project focuses on working with SFMs (Seismic Function Modules) which are essential components of the Omega platform.

Omega is divided into three main builds; a development build used by researches and software developers, an internal build used by the SIS team to process client data and a commercial build which is sold by the company to contractor for independent use. After research has been completed on new and innovative methods, they are initially implemented as an Omega SFM using the MATLAB language to serve as proof-of-concept and test the viability and the advantages of the method. The SFM then goes to an approval process which analysis the new SFMs usage in the market and the whether it would be deemed profitable to be released in the commercial version.

If approved then work begins to port the SFM to C++ version which provides stability and performance needed for the commercial build. My work at the time of writing consists of porting an experimental SFM, researched and developed by my supervisor Dr. Daniele Boiero, to C++ and prepare it for inclusion in the commercial build.

1.3 Splitting Intensity Analysis and Inversion

Seismic acquisitions are data recordings of shocks waves emitted by a source which propagate through the earth and are reflected by the subsurface layers. These waves are recorded using an array of receivers. There are two types of waves generally used in acquisitions: compressional P-waves and shear S-waves. These waves are then used in multicomponent acquisitions. By analyzing the wave velocity and angle of polarization geophysicists can characterize the anomalies and formations of the subsurfaces.

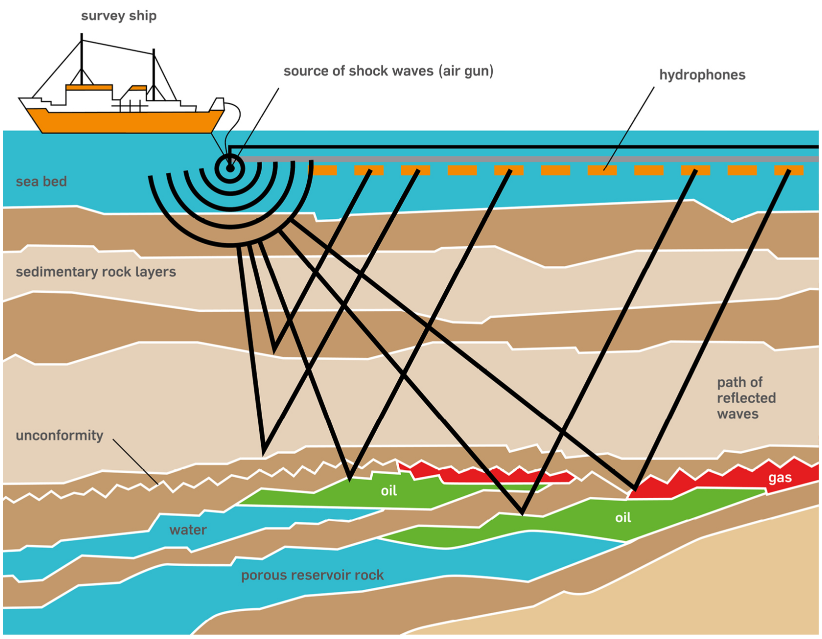

Figure 1: Illustration of seismic acquisition methods[3]

In azimuthally anisotropic media, the propagation velocity of S-waves varies with azimuth. In these cases, S-waves naturally polarizes into a fast wave (S1) parallel to the fractures (or stress) and a slow wave (S2) perpendicular to them. This phenomenon, known as shear-wave splitting (SWS), can cause difficulties in obtaining good PS images and must be corrected for mapping the reservoir. Dr. D. Boiero proposed a different approach which consists in inverting the interval values of splitting intensity to obtain a model of anisotropic parameters that varies with time or depth. This make it possible to estimate anisotropic properties within a geological formation by analyzing the differences of SI measured at the top and and the bottom of a layer making SWS simpler to apply. [4].

1.4 Theoretical Background

Split Intensity Inversion

For anisotropic media, the information about the structural properties and anomalies of the subsurfaces is contained in the change of time between the fast and slow

S-waves and the angle with the each layer of the subsurface. For a small δt compared with the dominant period of the signal it is possible to write the transverse component $T(t,\varphi)$, which is a function of time (t) and azimuth (φ), as a scaled time derivative of the radial component $R(t,\varphi)$:

$$R(t,\varphi) = w(t,\varphi) \tag{1}$$

$$T(t,\varphi) = \frac{1}{2}SI(\varphi)w'(t,\varphi) \tag{2}$$

In the simple case of a plane S-wave propagating vertically in a homogeneous horizontal transverse isotropic (HTI) medium; studying the variations of SI (Split Intensity) as a function of φ is a powerful technique to determine δt and φ_0. In the particular case of a weakly anisotropic layer, if the scale of lateral variations is much larger than the wavelength , SI is simply

$$SI_n(\varphi) = \sum_{i=1}^{n} \delta t_i \times \sin 2(\varphi - \varphi_i) \tag{3}$$

where n is the number of layers. Ultimately by inverting SI a model of splitting parameters can be obtained varying with time or depth.

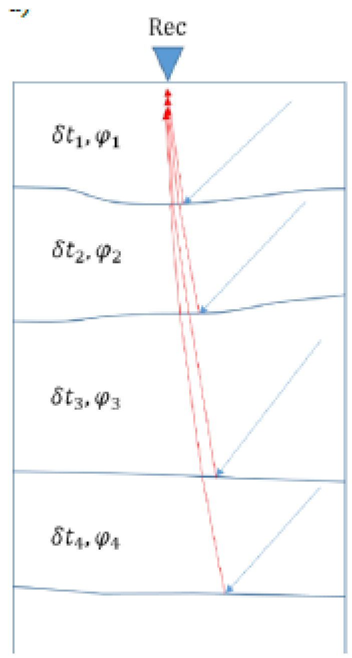

Figure 2: Converted waves at layer boundaries in an anisotropic medium

In order to compute the inversion of SI for a given dataset of seismic acquisition, firstly vertical and horizontal regularization matrices are computed with chosen input parameters according to the situation. In my perspective we can formulate the problem as an iterative process which involves firstly assembling the regularization matrices and then inverting the value of SI and checking the misfit of the computed model.

2 The project

My assigned project is divided into 3 main phases: remove the MATLAB wrapped C++ code and recode the SFM using C++ only, expand the inversion class for TTI(Tilted transverse isotropic) media, update documentation and present completed project to the company. Furthermore after recoding the SFM a multi-threading loop will be added to speed up the processing times.
I am assisted by Dr. Daniele Boiero, my supervisor on this project and also author of the paper on which the SFM is based, Dr. Lee West and Dr. Pradeep Loganathan, both part of the Signal Processing Software Engineering team. Dr. Boiero has provided invaluable support and training on Omega and the theory behind the SFM whereas both Dr. West and Dr. Loganathan have provided feedback and support on both the coding and testing of the SFM as well as various tools used for debugging the code. They also are part of the team which performs the code review before the revised SFM is submitted for commercial build.
At the moment of writing this report I have completed the porting to C++ and added multi threading and am in the process of preparing the revised SFM for an audit by the Omega Development Team. It is worth noting that the addition of multi-threading has decreased the computational time of the SFM for a particular set of data, from around 10 hours to just above 1 hour.

2.1 The structure of the SFM

The SFM in question is called Split Intensity Inverse. The module has two inputs, which are the measured data and the starting model, and two outputs, namely the calculated SI values and the output models.

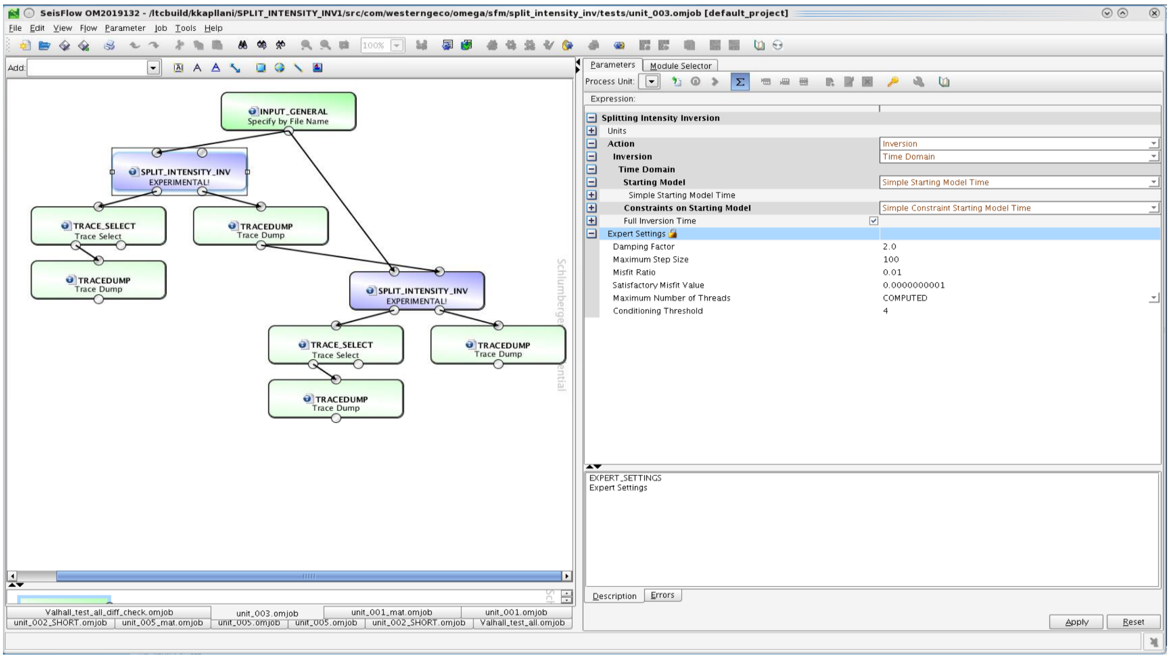

Figure 3: An Omega job using the Split Intensity Inv. SFM

A user has a selection of parameters to choose when running a job using this SFM. The structure of these options offered by the SFM is detailed in Figure 4. The two main options to chose from are whether to perform full inversion or conditioning. The latter one is an inversion of the splitting intensity parameters only without calculating the model values of δt and φ. If the full inversion is chosen the user then needs to specify the domain of the inversion, namely in time or depth. Although the inversion is always performed in the time domain, if the depth domain is chosen the parameters are converted to time before the inversion and then converted back in depth afterwards. This however is an unpopular usage due to acquisition data almost always being available in time domain.

Lastly the SFM contains a section for expert parameters, parameters set by default and inadvisable to change for general usage. One parameter which was added by me is the Maximum Number of Threads. This controls the use of Multi-Threading in the SFM. There are two options to choose from for this parameters: Computed or Not Used. When computed is chosen the SFM retrieves the number of processor cores available from the machine and sets that as the Maximum Number of Threads to use. The default values is computed.

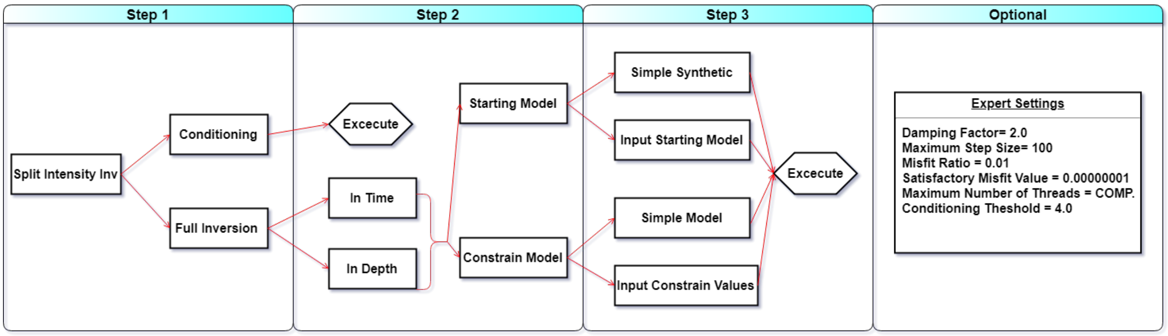

Figure 4: Flowchart of SFM options and parameters

2.2 The C++ code

The SFM is composed of five different classes which are all connected by the main class Split Intensity Inverse.

SplitIntensityInv class

The main class which interfaces with Omega work flow and prepares the data for processing. The class is responsible for reading the parameters specified by the user and also reading the input data and checking the validity of it. This class is converted to Java by Omega to be then executed accordingly. Hence this class includes all the other classes in the SFM and calls each of the according to the specified function in the job.

AnisotropicModelInfo

This class contains the model parameters and methods used to write and read parameters from the model. It also contains two methods which are used to convert the model from time domain to depth domain and vice versa.

AnisotropicModelInv

This class is the inversion class. When called by SplitIntensityInv it performs the full inversion of the inputted models contained in AnisotropicModelInfo and returns the update models and a misfit between the output models and the starting model. All the methods for generating the vertical and horizontal regularization matrices are contained within this class.

SemblancePicks

This class contains the SI picks measured as well as the methods to read and write update picks.

SIconditioning

This class performs the conditioning/inversion of the SI picks contained in the SemblancePicks class.

2.3 Skills acquired

The project presents some engaging challenges however I am pleased with my progress so far. I have had to apply a wide range of skills and in some cases new approaches to problems and issues faced until now. Team coordination and progress tracking have been some of the new skills which I have had to acquire. Additionally, I have used my EEE major knowledge in linear algebra to implement algorithms such as SVD (Singular Value Decomposition) for matrix inversion efficiently and bug-free in C++.
Moreover coding in the industrial environment is very different from coding assignments in education. During the first tow months of my placement I significantly improved my knowledge on both C++ and MATLAB which I believe will be beneficial in the long-term. Lastly I have been introduced to new memory diagnostic tools, such as valgrind, and acquainted myself with multi-threading C++ libraries(OpenMP, TBB, ...).

3 Project Management and Planning

An important aspect which has to be considered before undertaking a project is that it has to be carefully managed. The reasons behind this lie with the significant impact this can have both on the quality of the project as well as the timescale of

completion. Initially the main deliverables need to be established so that work is prioritized correctly. After that a suitable schedule is devised, underlining intermediary milestones and contingency considerations for the project.
In this particular project a very important aspect will be the testing of the revised SFM due to the strict rules and guidelines surrounding SFMs included in the Commercial Build of Omega. A series of unit tests, tests with synthetic data, need to be generated and ran on the new SFM and the results of these tests need to be compared to the results from the previous version. For Split Intensity Inversion there are 5 unit tests which asses the performance and the output of the SFM in all the operating modes. Furthermore after successful completion of the unit tests the SFM is tested on a set of real data in order to determine the improvement in performance and compatibility in results. Finally the code is submitted for a code review, where senior Software Developers analyze the added code and check for any inconsistencies or bugs. Lastly the accompanying documentation of the SFM is updated to reflect any changes to parameters or modes of operation.

3.1 Progress to Date

The main deliverables and my progress to date will be discussed in this section.

1. **Familiarization with developing environment**: The initial part of the project consisted of the introduction to Omega and its various set of tools as well as reading and understanding the theory behind the SFM. Additionally, the set up of my own development space in the Omega platform and the build area was conducted. This was completed in the first two weeks of the placement.

2. **Completing the Porting to C++**: Upon beginning my project the first deliverable was removing the MATLAB code and recoding in C++. A fixed timescale for this deliverable was not given by my supervisor as completion depends on the number of issues which might be encountered along the way. However this was completed at the end of Week 6.

3. **Testing and submission for revision**: After the C++ porting was completed I proceeded to the start the testing phase. The SFM was tested for all five unit tests as well together with real data as required from the guidelines. I am currently in the process of scheduling a code review and preparing the code accordingly. I estimate the submission will occur within the next two weeks.

4. **Expansion to include TTI media**: During the code review and submission stage I will begin the implementation of the expansions for TTI media. This deliverable will constitute most of the rest of my placement as in this case no MATLAB code is available and the expansion will need to be implemented directly from the research paper. Furthermore, new unit tests will be created and code coverage test will have to be run on these unit tests, to determine how much of the code they cover. Only then the unit tests can be executed and finally the results evaluated.

I have kept my supervisor continuously updated on my progress and weekly meetings have been conducted to discuss progress, possible issues and an eventual extension of the project in case of completion before the deadline.
Moreover, I have been involved in monthly informal meetings with the software engineers where new software and interesting projects have been presented and discussed.

3.2 Future Work - Gantt Chart

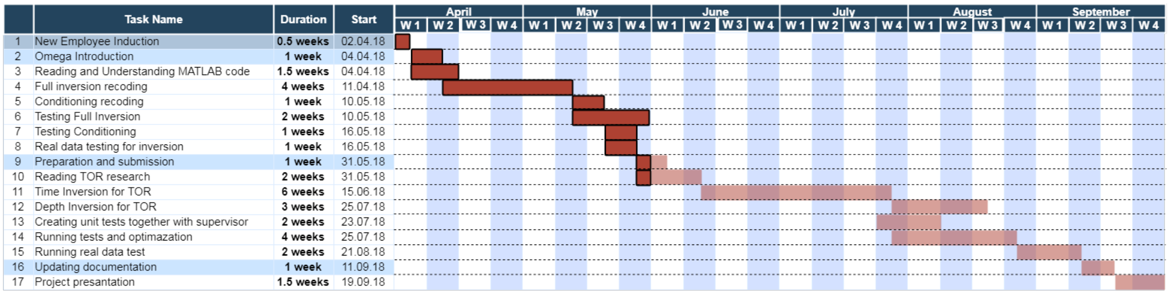

Figure 5: Gantt chart

With the removal of MATLAB files from the SFM and submission for a new revision nearing its completion the focus of the project will now focus on the TOR media expansion of the SFM. This will most probably prove to be quite challenging considering the fact that neither MATLAB files or unit tests exist for it. Furthermore, during this task I will work more closely with my supervisor to be able to comprehend the extensive geophysics background needed to complete the task.
The last few weeks of the placement will also include a presentation in front of WesternGeco team, where I will present a summary of my completed project and the advantages of it.

3.3 Contingency Considerations

As it is clear in the Gantt chart (See Section 3.2), the project has been roughly divided in three main phases. The contingency planning will now therefore concern the second and third phase. An alternative plan has been discussed with my supervisor in the eventuality of the second phase of the project will not be able to completed in time. In such a case a revised TOR inversion will be implemented which is simplified for time domain only.
Moreover the code reviews for both phases will have a significant impact on the timely completion. In case of negative feedback from the reviews, more time will have to be allocated for eventual fixes and modification to the code. This will lead to the plan being amended accordingly. However this has also been taken into account and will lead to the implementation of the simplified expansion.
Other unexpected delays and setbacks will be addressed as they present.

References

[1] Background schlumberger. *https://www.slb.com/about/who/backgrounder.aspx*, Accessed May 2018. pages 3

[2] Omega geophysical data processing platform. *https://www.software.slb.com/products/omega*, Accessed May 2018. pages 3

[3] Exploration krisenergy 2018. *https://krisenergy.com/company/about-oil-and-gas/exploration/*, Accessed May 2018. pages 4

[4] D. Boiero and C. Bagaini. Horizon-based splitting intensity analysis and inversion for anisotropic characterization. *78th EAGE Conference and Exhibition 2016*, 2016. pages 4

[5] D. Boiero and C. Bagaini. Shear-wave splitting intensity for tilted orthorhombic media. *79th EAGE Conference and Exhibition 2017*, Dec 2017. pages 12

4 Appendix

4.1 TTI and TOR media

This section introduces the theory behind the Inverse Analysis for Tilted Orthorhombic Media, on which the expansion of the SFM will be based. (cited at [5])
An upward-propagating S-wave with displacement $\boldsymbol{u}_S^0(z,t)$ that propagate through a single TOR(Tilted Orthorhombic Media) creates a displacement $\boldsymbol{u}_S^1(z,t)$ at the top which can be modeled as:

$$\begin{bmatrix} u_R^1 \\ u_T^1 \\ u_Z^1 \end{bmatrix} = e^{i\omega t}\boldsymbol{R}(-\varphi,-\theta,-\psi)\boldsymbol{H}\boldsymbol{R}(\varphi,\theta,\psi)\begin{bmatrix} u_R^0 \\ u_T^0 \\ u_Z^0 \end{bmatrix} \tag{4}$$

We want to express u_T^1 as a function of u_R^0 and u_T^0. If we consider the initial amplitude of $u_Z^1 = -u_R^0 \tan I$, where I is the incident angle, we can obtain from equation (4):

$$u_T^1 = -\frac{1}{2}[(E\sin 2(\varphi_w-\varphi)) + F\cos 2(\varphi_w-\varphi) + \tan I(G\sin(\varphi_w-\varphi) + H\cos(\varphi_w-\varphi))]u_{R_0} \tag{5}$$

Therefore SI is equal to:

$$SI = E\sin 2(\varphi_w-\varphi) + F\cos 2(\varphi_w-\varphi) + \tan I(G\sin(\varphi_w-\varphi) + H\cos(\varphi_w-\varphi)) \tag{6}$$

From equation (5) we notice that this is an expanded expression for SI when compared to equation (3) and therefore the implementation will require the addition of few terms for both SI and regularization matrices as they depend on the length of the SI matrix.