

BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA  
FACULTAD DE INGENIERÍA



REPRODUCIBILIDAD COMPUTACIONAL PARA CONVERSIÓN TIEMPO  
A PROFUNDIDAD DE DATOS SÍSMICOS

TESIS

Que para obtener el título de  
INGENIERO GEOFÍSICO

Presenta:

Javier Nieto Baltazar

Director de tesis:

Sergio Chávez Pérez

Asesor interno:

José Castillo Román

PUEBLA, PUE.

MAYO 2018



*"Lo que nos hace grandes es el hecho de que  
podamos ver lo pequeños que somos"*



# Resumen

Convertir datos sísmicos de tiempo a profundidad es un proceso recurrente en el ámbito petrolero, debido a la importancia de conocer la localización de estructuras y zonas de interés en profundidad. Normalmente la conversión es llevada a cabo en paqueterías con licencias comerciales que conlleva altos costos, llegando a ser este uno de los principales factores de accesibilidad. Existe una creciente tendencia al código abierto, que ofrece una alternativa para realizar conversión tiempo a profundidad con herramientas libres. Con ello promover beneficios como disminución de costos, independencia de licencias comerciales, innovación tecnológica, trasmisión de conocimiento y que cualquier profesional de la práctica pueda llevarlo a cabo. Este trabajo ilustra una manera de ejecutar conversión tiempo a profundidad, con documentos reproducibles y en diferentes herramientas computacionales basadas en paquetería libre.



# Abstract

The time-to-depth conversion of seismic data is a recurrent process in the industry of oil and gas, because is necessary to know the location of structures and interest areas on depth. Usually time-to-depth conversion is done with software that needs a commercial licensed but it brings high costs, becoming this one of the main factors for the accessibility to software. Today there is upward trend in open source, which offers an alternative to realize time-to-depth conversion with free tools. Using that kind of tools to bring benefits such as low cost and decreasing of dependence of commercial licenses, technological innovation, transmission of knowledge and where any professional can carry it out. This work illustrates a way to execute time-to-depth conversion, with reproducible documents and software that are based on open source.



# Contenido

<b>Resumen</b>	<b>I</b>
<b>Abstract</b>	<b>III</b>
<b>Lista de figuras</b>	<b>VII</b>
<b>Lista de tablas</b>	<b>VIII</b>
<b>Introducción</b>	<b>1</b>
<b>1. Reproducibilidad computacional en representación sísmica</b>	<b>3</b>
1.1. Relevancia del código abierto . . . . .	4
1.2. Papel del código abierto en geociencias . . . . .	5
<b>2. Conversión tiempo a profundidad de datos sísmicos</b>	<b>11</b>
2.1. Importancia del modelo de velocidad . . . . .	13
2.2. Velocidades . . . . .	13
2.3. Herramientas computacionales . . . . .	17
2.3.1. Opendtect . . . . .	21
2.3.2. Madagascar . . . . .	24

2.3.3. Seismic Unix . . . . .	31
<b>3. Estimación de velocidad de intervalo</b>	<b>33</b>
3.1. Estimación de velocidad de intervalo utilizando ecuación de Dix .	34
3.2. Estimación de velocidad de intervalo utilizando teoría de rayos paraxiales . . . . .	35
3.3. Caso I . . . . .	40
3.4. Caso II . . . . .	44
<b>4. Discusión</b>	<b>47</b>
<b>Conclusiones y recomendaciones</b>	<b>49</b>
<b>Agradecimientos</b>	<b>51</b>
<b>Referencias</b>	<b>53</b>
<b>Apéndice A</b>	<b>57</b>
<b>Apéndice B</b>	<b>65</b>

# Índice de figuras

1.1. <i>Paqueterías de código abierto más populares para el análisis, visualización, manipulación y procesamiento de datos sísmicos (Vargas-Meleza, Nieto-Baltazar, et al., 2017)</i> . . . . .	7
2.1. <i>a) Sección sísmica longitudinal en tiempo del volumen sísmico. b) Sección longitudinal del volumen de velocidad.</i> . . . . .	20
2.2. <i>a) Sección sísmica longitudinal convertida de tiempo a profundidad utilizando Opendtect. b) Sección sísmica longitudinal convertida de profundidad a tiempo utilizando Opendtect.</i> . . . . .	23
2.3. <i>Modelo de velocidad de intervalo en 1D generado por interpolación polinomial a partir de datos puntuales.</i> . . . . .	26
2.4. <i>a) Sección sísmica longitudinal convertida de tiempo a profundidad utilizando el modelo 1D sintético de velocidad de intervalo con Madagascar. b) Sección sísmica longitudinal convertida de profundidad a tiempo utilizando el modelo sintético 1D de velocidad de intervalo con Madagascar.</i> . . . . .	27
2.5. <i>Modelo sintético 2D de velocidad de intervalo (Valores de cada capa en orden de más superficial a más profunda; 1500 m/s, 1936 m/s, 1977 m/s, 2003 m/s, 2167 m/s.</i> . . . . .	29

2.6. a) Sección sísmica longitudinal convertida de tiempo a profundidad utilizando el modelo sintético 2D de velocidad de intervalo con Madagascar.	
b) Sección sísmica longitudinal convertida de profundidad a tiempo utilizando el modelo sintético 2D de velocidad de intervalo con Madagascar.	30
2.7. a) Sección sísmica longitudinal obtenida de tiempo a profundidad utilizando Seismic Unix. b) Sección sísmica longitudinal de profundidad a tiempo utilizando Seismic Unix.	32
3.1. Representación gráfica del concepto de rayo central, para localizar y evaluar dos puntos representados por planos perpendiculares, en su variación respecto al tiempo.	37
3.2. Ilustración del rayo paraxial, donde surge una familia de rayos en la superficie y son propagados hacia abajo hasta el punto $(x_o, t_o)$ , donde son evaluados para conocer la dispersión geométrica de la familia de rayos.	39
3.3. a) Modelo de velocidad de intervalo utilizando Dix. b) Modelo de velocidad de intervalo utilizando rayos paraxiales. c) Datos sísmicos en profundidad para modelo de velocidad por Dix. d) Datos sísmicos en profundidad para modelo de velocidad por rayos paraxiales.	43
3.4. a) Modelo de velocidad de intervalo utilizando Dix. b) Modelo de velocidad de intervalo utilizando rayos paraxiales. c) Datos sísmicos en profundidad para modelo de velocidad por Dix. d) Datos sísmicos en profundidad para modelo de velocidad por rayos paraxiales.	45

# Índice de tablas

2.1. Comparación entre principales características de migración en profundidad y conversión a profundidad (Tomada y modificada de <a href="https://www.linkedin.com/pulse/imaging-vs-depth-conversion-rajeshwaran-dandapani/">https://www.linkedin.com/pulse/imaging-vs-depth-conversion-rajeshwaran-dandapani/</a> ?trk=hp-feed-article-title-publish).	12
--	----



# Introducción

Los datos sísmicos de reflexión para exploración petrolera tienen su origen al medir el tiempo de propagación de las ondas en el subsuelo. Por lo tanto, la información está en unidades de tiempo. Pero si se requiere conocer la profundidad en la que están ubicadas las estructuras geológicas, es necesario convertir a unidades de distancia. Para ello se recurre a metodologías de conversión tiempo a profundidad.

Los datos sísmicos en profundidad son necesarios para evaluar la viabilidad del estudio de perforación, ubicación de estructuras en profundidad, zona de interés o conocer la distancia del yacimiento. Esto hace que la conversión tiempo a profundidad para datos sísmicos sea un proceso relevante en el ámbito petrolero. El mercado ofrece diferentes opciones para llevarla a cabo, pero en la mayoría es necesario contar con una licencia comercial. Por lo tanto, implica altos costos y limita el acceso a estudiantes e investigadores.

Hay una tendencia creciente en diferentes áreas sobre implementación de código abierto, tanto para la industria como para la academia. En exploración sísmica existen diversos proyectos que tienen como objetivo el desarrollo de herramientas

con código abierto para representación sísmica y reproducibilidad computacional. El acceso a este tipo de herramientas conlleva beneficios como: reducción de costos, promover colaboración entre profesionales e investigadores, transmisión de conocimiento, y disminución del rezago tecnológico, permitiendo adaptar y desarrollar tecnología con base en conocimiento y/o herramientas previas.

El objetivo de realizar este trabajo es discutir la conversión de datos sísmicos de tiempo a profundidad, con herramientas computacionales basadas en paquetería libre. Y demostrar que, con un equipo de cómputo al alcance de cualquier estudiante o profesional, es factible la conversión tiempo a profundidad de una forma económica y eficiente.

# Capítulo 1

## Reproducibilidad computacional en representación sísmica

La investigación reproducible es una forma de trabajo que exige incluir código y datos a publicaciones, con el fin de permitir la verificación independiente y replicación de experimentos computacionales (Claerbout y Fomel, 2009). Hace poco más de cuatro décadas, tuvo como principales pioneros a Jon Claerbout, profesor de geofísica en la universidad de Standord, junto con su equipo de investigadores del Proyecto de Exploración de Stanford (SEP, por sus siglas en inglés), un consorcio académico patrocinado por la industria petrolera. Creando SEPlib, una de las paquetes pioneras en fomentar la reproducibilidad en sismología de exploración, una plataforma para procesamiento de datos sísmicos, basada en la filosofía de reproducibilidad donde el usuario tiene la posibilidad de realizar tres actividades importantes: 1) estudiar y verificar los programas; 2) reproducir de manera automática los resultados; 3) variar los parámetros originales para analizar diferentes escenarios.

Actualmente, la reproducibilidad resulta más accesible para el usuario, gracias al uso de lenguajes de programación como Python (Hall, 2016), herramientas para compilación de programas como SCons, y ambientes para cálculo numérico como MATLAB (Donoho et al., 2009).

Llevar a cabo investigación reproducible implica tiempo y esfuerzo. Además de considerar que los programas deben estar en constante mantenimiento para evitar su obsolescencia. Las paqueterías utilizadas para este trabajo están encaminadas a la publicación de documentos reproducibles, y el aprendizaje apoyado en este tipo de documentos permite una mejor comprensión del cómputo implicado.

## 1.1. Relevancia del código abierto

Los recientes avances de tecnologías en información y comunicación han dado lugar al surgimiento de políticas de acceso libre encaminadas a la difusión. Por lo tanto, las herramientas computacionales de código abierto han comenzado a ganar relevancia.

Cabe aclarar que estrictamente código abierto (*open source*, en inglés) no es lo mismo que paquetería libre (*free software*, en inglés). Su principal diferencia está en los diferentes puntos de vista a los valores fundamentales (<https://www.gnu.org/philosophy/open-source-misses-the-point.en.html>). Pero ambos van encaminados a la libertad de ejecutar, estudiar, modificar y redistribuir el código. Los programas computacionales con estas características, comúnmente se les llama de “código abierto”.

Con código abierto uno como usuario tiene acceso al código fuente, adaptar el código a nuestras necesidades y conveniencias, usar los proyectos de otras comunidades, desarrollar diferentes planteamientos y soluciones a los problemas, auditar el código que estamos usando, ver exactamente lo que está sucediendo y compartir con otros usuarios para colaborar y/o mejorarlo.

Este tipo de proyectos, basados en esta filosofía han venido siendo utilizados desde hace tiempo y comenzado a tener relevancia en diversos proyectos de diferentes áreas. Por ejemplo, *Android* que es la plataforma móvil más distribuida para móviles, y basada sobre código abierto. Quizás el proyecto más conocido es *Linux*, justamente nace como un proyecto de código abierto y se ha mantenido así en los años. *Linux* es uno de los sistemas operativos más usados, tiene miles de colaboradores que lo mejoran día a día y que ha sido muy bien recibido por la comunidad debido a esa flexibilidad de crear soluciones a casos específicos. También existen comunidades como es *Github* en donde proyectos de diferentes disciplinas, están disponibles para colaborar y ser usados (disponible en: <https://github.com>).

## 1.2. Papel del código abierto en geociencias

La existencia de programas de código abierto representa una gran herramienta en investigación y docencia. Al investigador le permite llevar a cabo su trabajo libremente sin comprar licencias de paquetes comerciales con altos precios, que en caso de representación sísmica puede ascender a miles de dólares (González-Verdejo y Chávez-Pérez, 2012). El beneficio se extiende a profesores y estudiantes, quienes así pueden trabajar con herramientas a las que no podrían acceder de otra manera.

Han existido diversos proyectos de difusión sobre código abierto, aplicados a solucionar problemas en geociencias. Por ejemplo, la *Society of Exploration Geophysicist* (SEG), que es la principal organización sin fines de lucro para promover y divulgar la geofísica de exploración, en su revista *Geophysics*, una de las más importantes para la divulgación de investigación y educación en geofísica aplicada, ha creado una sección *Geophysical Software and Algorithms* para la divulgación de artículos que cumplan con los lineamientos de reproducibilidad (disponible en: <http://software.seg.org/>). *The Leading Edge*, otra revista, cuenta con su serie de tutoriales con código sobre algún tema de exploración geofísica, y realizan la publicación a través de la sección especial del sitio en internet *SEGwiki* (disponible en: [http://wiki.seg.org/wiki/Geophysical\\_tutorials](http://wiki.seg.org/wiki/Geophysical_tutorials)).

Creación de talleres y sesiones especiales para divulgación de este tipo de herramientas, en las reuniones anuales de la Sociedad Europea de Geocientíficos e Ingenieros (EAGE, por sus siglas en inglés). Sitios en internet que proporcionan información de diferentes paquetes para procesamiento e interpretación: Comparison of free geophysics software (disponible en: [https://en.wikipedia.org/wiki/Comparison\\_of\\_free\\_geophysics\\_software](https://en.wikipedia.org/wiki/Comparison_of_free_geophysics_software)).

En sismología de exploración, las técnicas de procesamiento de datos están en constante innovación, tanto a nivel académico como industrial e implican un alto grado de complejidad computacional. En la industria petrolera, estas innovaciones ocurren con la finalidad de desarrollar programas comerciales o mejorar los sistemas computacionales internos de las compañías petroleras y de servicios. Por motivos de confidencialidad se restringe el acceso a sus códigos fuente. Por lo tan-



Figura 1.1: *Paqueterías de código abierto más populares para el análisis, visualización, manipulación y procesamiento de datos sísmicos (Vargas-Meleza, Nieto-Baltazar, et al., 2017)*

to, una estrategia económica para reducir el rezago tecnológico y la dependencia de herramientas comerciales en la industria e investigación petroleras es aprovechar el potencial de las interfaces de programas de aplicación de código abierto mencionadas en (Fig. 1.1), ilustrando algunas de las paqueterías más relevantes en sismología de exploración.

*Madagascar* provee un ambiente computacional que permite manipular, procesar y visualizar datos sísmicos. Contiene una colección de módulos de cálculo, secuencias de procesamiento de datos y publicaciones científicas reproducibles.

Es una de las principales plataformas para la reproducibilidad computacional en sismología de exploración para procesar datos sísmicos, antes y después de apilar (disponible en: [http://www.ahay.org/wiki/Main\\_Page](http://www.ahay.org/wiki/Main_Page)).

*Attribute Assisted Seismic Processing and Interpretation* (AASPI, por sus siglas en inglés) desarrollada por el consorcio académico de investigación de la Universidad de Oklahoma. Esta paquetería es de código abierto pero no es gratuita, permite el cálculo de atributos sísmicos (disponible en: <http://mcee.ou.edu/aaspi/about.html>).

*Jupyter* permite crear cuadernos de trabajo, que es un entorno interactivo en el cual se puede combinar código, texto y gráficas. Con la principal característica que nos permite editar y ejecutar desde un navegador (disponible en: <http://jupyter.org/>).

*Seismic Unix* es una paquetería de código abierto enfocada la visualización y procesamiento de datos sísmicos, fue desarrollado por el consorcio de *Center of Wave Phenomena* que pertenece a la Escuela de Minas de Colorado (disponible en: <https://cwp.mines.edu/>).

*OpenDtect* es una paquetería con orientación hacia la visualización e interpretación de datos sísmicos. Está constituido en dos partes, la primera es código abierto que ofrece herramientas para visualizar, procesar e interpretar datos sísmicos, mientras que la parte privativa ofrece herramientas avanzadas y complementos compatibles con otras paqueterías comerciales (disponible en: <https://opendtect.org/>).

[www.dgbes.com/](http://www.dgbes.com/)).

Hay que tener en cuenta ciertos aspectos para poder usar estas paqueterías y en consecuencia obtener ventaja de sus beneficios:

- a) El esfuerzo intelectual realizado por el usuario es mayor, debido a que, si no cuenta con conocimientos básicos en algún lenguaje de programación, la creación de documentos reproducibles será complicada.
- b) La curva de aprendizaje es mayor, gran cantidad de paqueterías libres se ejecutan en entornos Linux, la mayoría de los usuarios no está familiarizado con este sistema.
- c) Ejecutar herramientas de la paquetería no es trivial, pues se deben ejecutar desde una interfaz gráfica (terminal) o en secuencia de instrucciones (script, en inglés).
- d) El entorno de las paqueterías no es amigable al usuario, a diferencia como sucede en la mayoría de las paqueterías con licencia comercial.
- e) Conlleva implicaciones éticas, debido a que el usuario tiene total responsabilidad de sus propósitos y uso que le dé al código.



# **Capítulo 2**

## **Conversión tiempo a profundidad de datos sísmicos**

Los datos sísmicos de reflexión son la herramienta más importante en exploración petrolera y con el avance de los años se han ido perfeccionado en calidad y precisión las imágenes que representan el subsuelo. El método mide el tiempo de viaje de las ondas, y la información está en unidades de tiempo. La importancia de tener nuestros datos sísmicos en unidades de distancia es conocer la profundidad de los horizontes, prospectos y pozos en el sitio de interés. Ya que las estructuras geológicas son medidas en estas últimas unidades.

Conversión tiempo a profundidad es un proceso complejo, que comienza desde un adecuado procesamiento sísmico, análisis de velocidad, obtención de velocidades y el estudio de datos de pozos para refinar la conversión (Hosken y Deregowski., 1985).

La conversión de datos sísmicos de tiempo a profundidad puede considerarse trivial porque es comprendida como una simple relación tiempo, velocidad y distancia, pero esto depende del método que sea utilizado. Algunos de los métodos de conversión tiempo a profundidad son: por escalamiento (Robein, 2003), trazado de rayos de imagen (Hubral, 1980; Filpo et al., 2016) o trazado de rayos paraxiales (Cameron et al., 2007). La conversión es un tema relevante en la actualidad, la búsqueda de la mejora de técnicas y el avance en computación ha hecho que sigan investigando nuevas metodologías, como utilización de sistemas de ecuaciones diferenciales parciales para la obtención de la velocidad de intervalo en presencia de variaciones laterales de velocidad (Li y Fomel, 2015) o realizar la conversión mediante frentes de onda de imagen (Sadala at al., 2017).

<b>Migración en profundidad contra conversión a profundidad</b>	
<ul style="list-style-type: none"> <li>- Repositionamiento lateral de reflectores.</li> <li>- Colapsamiento de difracciones.</li> <li>- Produce una imagen estructural correcta X, Y.</li> <li>- Mejora la resolución vertical y horizontal.</li> <li>- Posicionamiento lateral de amplitudes.</li> <li>- Se hace antes de la conversión a profundidad.</li> <li>- Es principalmente una corrección lateral.</li> <li>- Utiliza velocidades de migración.</li> </ul>	<ul style="list-style-type: none"> <li>- Convierte tiempo de arriba a profundidades.</li> <li>- Produce predicciones de profundidades fuera de los pozos.</li> <li>- Calibración vertical absoluta.</li> <li>- La conversión a profundidad se hace después de la migración.</li> <li>- La conversión a profundidad es estrictamente una corrección vertical.</li> <li>- Conversión a profundidad utiliza velocidades de propagación vertical.</li> </ul>

Tabla 2.1: Comparación entre principales características de migración en profundidad y conversión a profundidad (Tomada y modificada de <https://www.linkedin.com/pulse/imaging-vs-depth-conversion-rajeshwaran-dandapani/?trk=hp-feed-article-title-publish>).

Es común confundir entre migración en profundidad y conversión tiempo a profundidad. Pero son procesos independientes y diferente propósito. Pueden compartir modelo de velocidad, aunque los objetivos de cada proceso son diferentes como se puede apreciar en (Tabla 2.1).

## 2.1. Importancia del modelo de velocidad

Uno de los datos esenciales para realizar conversión tiempo a profundidad es la velocidad. La adecuada obtención de las velocidades es un factor crucial en sismología de exploración. En el caso de conversión tiempo a profundidad, el modelo de velocidad tiene un papel relevante en el reposicionamiento de reflectores. Debe haber especial cuidado en el manejo de las velocidades ya que llegan a distorsionar: valores en profundidad, tamaño y formas de las estructuras (Robein, 2003).

En sísmica de exploración son indispensables los modelos geológicos y de velocidad, el primero representa el estudio geológico para la caracterización del yacimiento mientras que el segundo es una distribución espacial de la velocidad, a menudo estos últimos son aplicados para realizar migración y/o conversión tiempo a profundidad.

## 2.2. Velocidades

Velocidad es definida como la distancia que recorre un objeto entre el tiempo invertido en ello. En sísmica registramos los tiempos de arribo de la energía que se propaga en el substituto, que varía dependiendo de las velocidades de las rocas, y se rige por la ecuación de onda (2.1), donde  $v(x,y,z)$  es la velocidad del medio,

$U(x,y,z,t)$  la función de onda respecto al tiempo.

$$\nabla U(x, y, z, t) = \frac{1}{v^2(x, y, z)} \frac{\partial^2}{\partial^2 t} U(x, y, z, t). \quad (2.1)$$

A menudo se habla de la “velocidad” que tiene una capa geológica, esto para los geofísicos representa la velocidad de intervalo (2.4). En exploración sísmica se manejan diferentes conceptos de velocidad, dependiendo de los estudios a realizar, la información que se tenga y de las metodologías a desarrollar. Esto nos sirve para un mejor entendimiento de los fenómenos físicos. La velocidad varía en función de la litología, profundidad, porosidad del material, compactación, litificación, contenido de fluidos, entre otros (Robein, 2003).

## Velocidad instantánea

Es definida como el límite de la velocidad media cuando el intervalo de tiempo considerado tiende a 0. Este término se usa para la velocidad calculada de los registros de impedancia acústica (Sheriff, 2002). Donde  $V_i$  es velocidad instantánea,  $\Delta x$  es variación de distancia y  $\Delta t$  variación respecto al tiempo.

$$V_i = \lim_{\Delta t \rightarrow 0} \frac{\Delta x}{\Delta t} = \frac{dx}{dt}. \quad (2.2)$$

## Velocidad promedio

Es el cociente de la suma de los espesores recorridos entre el tiempo total de viaje de la onda en dichos espesores. Se calcula considerando una trayectoria vertical y estratificación paralela (Yilmaz, 1987). Donde  $V_{prom}$  es velocidad promedio,  $V_{int}$  velocidad de intervalo y  $t$  es tiempo.

$$V_{prom} = \frac{\sum_{i=1}^n V_{int} t_i}{\sum_{i=1}^n t_i}. \quad (2.3)$$

## Velocidad de intervalo

Velocidad que caracteriza una capa o intervalo, conocida también como velocidad de formación en el ámbito geológico (Robein, 2003).

$$V_{int} = \frac{Z_{n-1} - Z_n}{t_{n-1} - t_n}, \quad (2.4)$$

donde  $Z_{n-1}$  es la profundidad de la cima para la capa y  $Z_n$  profundidad de la base de la capa,  $t_{n-1}$  tiempo inicial y  $t_n$  tiempo total de recorrido. También es frecuente calcular una aproximación utilizando ecuación de Dix, sísmica de pozo o registros geofísicos.

## Velocidad cuadrática media

La velocidad cuadrática media (RMS, por sus siglas en inglés), en ocasiones también llamada velocidad de Dix (Byun et al., 1988), es un promedio ponderado de la velocidad con la que viajan o propagan las ondas a través de un subsuelo

estratificado o intervalos a lo largo de una trayectoria específica.

$$V_{rms}^2 = \frac{\sum_{i=1}^n V^2 int_i \Delta t_i}{\sum_{i=1}^n t_i}, \quad (2.5)$$

donde  $t_i$  es tiempo de viaje en una dirección a través de la capa,  $V_{int}$  es velocidad de intervalo en cada capa.

## **Velocidad de apilamiento**

La velocidad de apilamiento se utiliza para corregir los tiempos de arribo de las reflexiones sísmicas por sobretiempo normal (Normal Moveout, en inglés), antes de sumar o apilar las trazas. Normalmente es obtenida realizando un análisis de velocidad y determina la relación tiempo a distancia a partir del análisis de las mediciones del sobretiempo normal por distancia.

En la práctica, es considerado como iguales: las velocidades de apilamiento y velocidad cuadrática media, debido a que sus valores de velocidad son similares (Byun et al., 1988).

## **Fuentes de velocidades sísmicas**

Existen diferentes fuentes donde se pueden obtener valores de velocidad de las formaciones. Los más comunes son los siguientes.

### Tiros de prueba

Es una de las formas más comunes en exploración sísmica, para conocer la velocidad de propagación de la onda. El método consiste en medir el tiempo de recorrido de la onda sísmica desde la fuente en superficie hasta el receptor dentro de un pozo, este detector se va colocando a distintas posiciones conocidas. Y el receptor registrará los tiempos de las primeras llegadas, con el cual es obtenida la velocidad.

### Registro sónico

Uno de los métodos más recurridos para mediar la velocidad, es mediante una herramienta sónica de pozo. Consiste en emitir una onda acústica que se propaga desde la fuente hasta la formación y de regreso hasta un receptor. Este es conocido como tiempo de tránsito (Schlumberger, 1989).

### Análisis de velocidad por semblanza

Es un proceso de cálculo de la velocidad de apilamiento, mediante la utilización de datos de punto medio común. Tiene como base el sobretiempo normal por distancia fuente receptor para determinar velocidades a partir de datos sísmicos (Sheriff, 2002). La precisión del método depende de la relación señal-ruido, la cual afecta la selección de los valores en el espectro de velocidad (Taner y Koehler, 1969).

## **2.3. Herramientas computacionales**

El objetivo de este trabajo es demostrar que, contando con datos sísmicos e información de velocidad, uno puede realizar conversión tiempo a profundidad

mediante el uso de paquetería libre y con recursos computacionales que estén al alcance de cualquier profesional. Las paqueterías libres que utilicé para realizar la conversión tiempo a profundidad son: OpendTect, Madagascar y Seismic Unix. Fueron elegidas debido a que son las más conocidas y de fácil acceso.

Con respecto a los datos utilizados, pertenecen al repositorio de datos libres de OpendTect. La elección de estos datos fue debido a que cuentan con un volumen de datos sísmicos en tiempo y con su respectiva información de velocidades, además de atributos sísmicos e información de horizontes (disponible en: <https://www.opendtect.org/osr/>). La información sísmica pertenece a un bloque ubicado en el sector holandés del Mar del Norte. El bloque fue cubierto con sísmica 3D con fines comerciales. Corresponde a la parte del Mioceno al Pleistoceno, geológicamente representa un paquete deltaico conformado por arenas y lutitas.

Con respecto a la velocidad, los datos son proporcionados dentro de un archivo de texto, que consta de una colección de puntos en varias posiciones, para después generar el volumen de velocidad de intervalo mediante interpolación lineal.

Para realizar conversión tiempo a profundidad en las diferentes paqueterías, es mediante la siguiente forma. Una vez teniendo los datos de entrada que son datos sísmicos en tiempo (Fig. 2.1a) y los valores de velocidad de intervalo (Fig. 2.1b), entonces convierto de tiempo a profundidad para obtener datos sísmicos en profundidad. Una vez hecho lo anterior, convierto de profundidad a tiempo para obtener nuevamente datos sísmicos en tiempo y observar el reposicionamiento que han tenido los reflectores.

Cabe mencionar que el método usado en cada paquetería para realizar conversión tiempo a profundidad es por escalamiento, que está basada en la siguiente ecuación (2.6) que describe la relación entre velocidad y tiempo para calcular la profundidad de las capas (Clearboult, 1985). Donde  $v_i$  es valor de velocidad de intervalo,  $t$  es tiempo y multiplicado por 1/2 debido al doble tiempo de viaje.

$$Z = \frac{1}{2} \sum_{i=0}^n V_i(t_{i-1} - t_i). \quad (2.6)$$

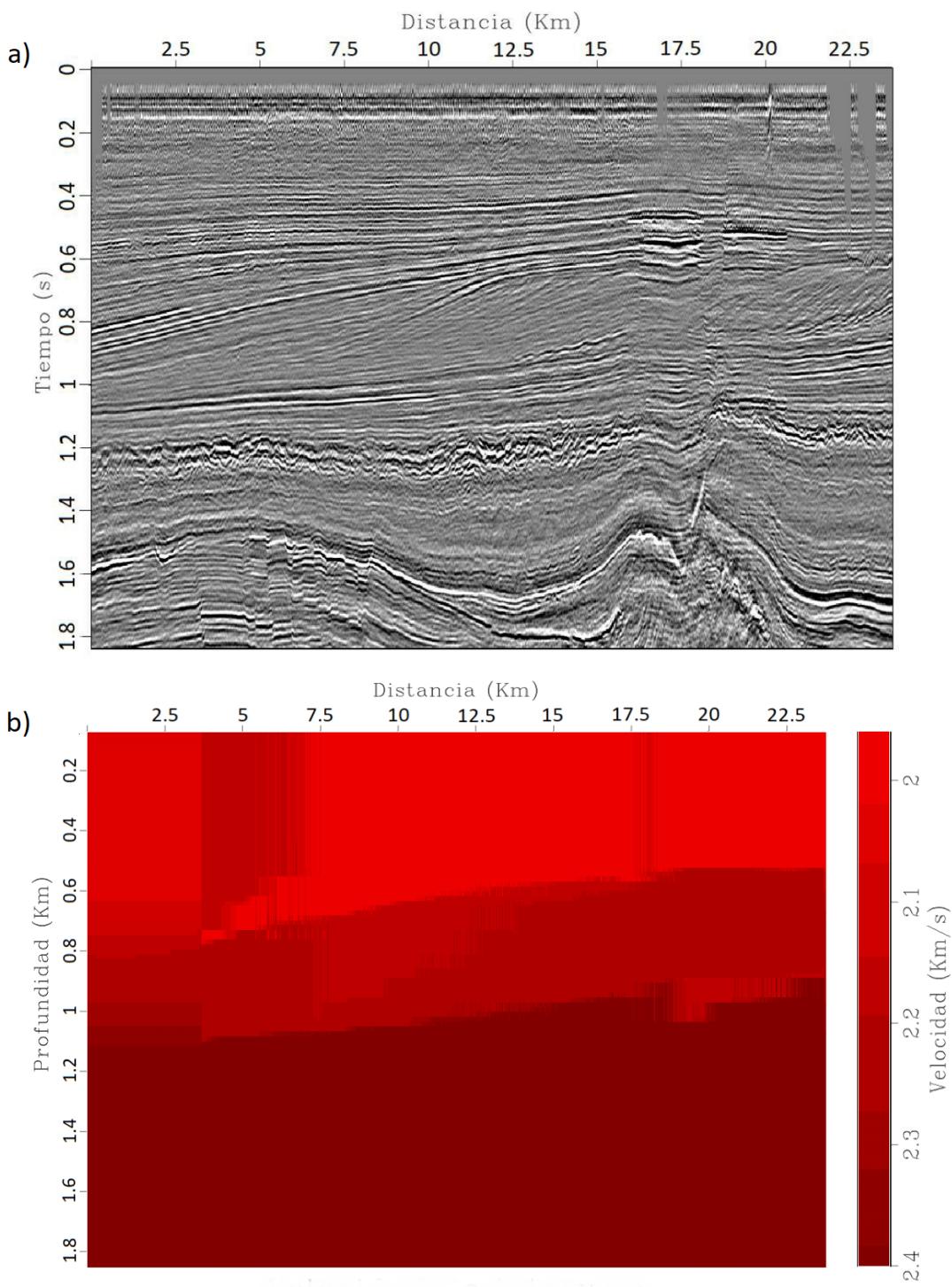


Figura 2.1: a) Sección sísmica longitudinal en tiempo del volumen sísmico. b) Sección longitudinal del volumen de velocidad.

### 2.3.1. Opendtect

OpendTect es una paquetería libre con un enfoque para visualización e integración de datos sísmicos, principalmente para interpretación. La versión libre sólo cuenta con herramientas básicas, debido a que si uno requiere herramientas más avanzadas hay que adquirir una licencia. Sin embargo, para este propósito no existe inconveniente, la herramienta para realizar la conversión tiempo a profundidad es libre.

El entorno de la paquetería tiene un parecido a otras en el mercado que requieren licencia comercial, esto hace que la manipulación de los datos sea más cómoda en comparación con otras paqueterías libres. No cuenta con muchas opciones de procesamiento sísmico. Sin embargo, la versión para *Linux* es compatible con *Madagascar*. Esto permite ejecutar algoritmos de esta última paquetería dentro del entorno OpendTect. Ofrece visualización para datos sísmicos que puede ser 2D o 3D.

Una vez teniendo datos de entrada para realizar la conversión, que son sección sísmica en tiempo (Fig. 2.1a) y velocidad de intervalo (Fig. 2.1b), realizo la conversión a profundidad. Opendtect la ofrece mediante la opción de *Time to Depth Conversion*, alojada en el menú de procesamiento. El comando está basado en la ecuación (2.6), (<http://doc.opendtect.org/5.0.0/doc/>).

El resultado obtenido representa datos sísmicos en profundidad (Fig. 2.2a), en esta figura se observar un ligero reposicionamiento de los reflectores y un cambio de valores en el eje Y. En los 3.5 Km de distancia se aprecia una discontinuidad

de los reflectores debido a el modelo de velocidad que los presenta (recuadro azul).

OpendTect nos ofrece realizar conversión de profundidad a tiempo mediante la opción *Time to Depth Conversion*, que es la misma del anterior caso, sólo hay que indicar que tipo de conversión se desea. La figura (Fig. 2.2b) es resultado de realizar conversión profundidad a tiempo, a comparación de los datos sísmicos en tiempo originales (Fig. 2.1a), se observa que los reflectores sísmicos no tienen la misma posición y que están a un ligero mayor tiempo que la sección original, esto es debido a que estrictamente no es un cambio de dominio sino un reajuste de coordenadas que tienen un error acumulativo.

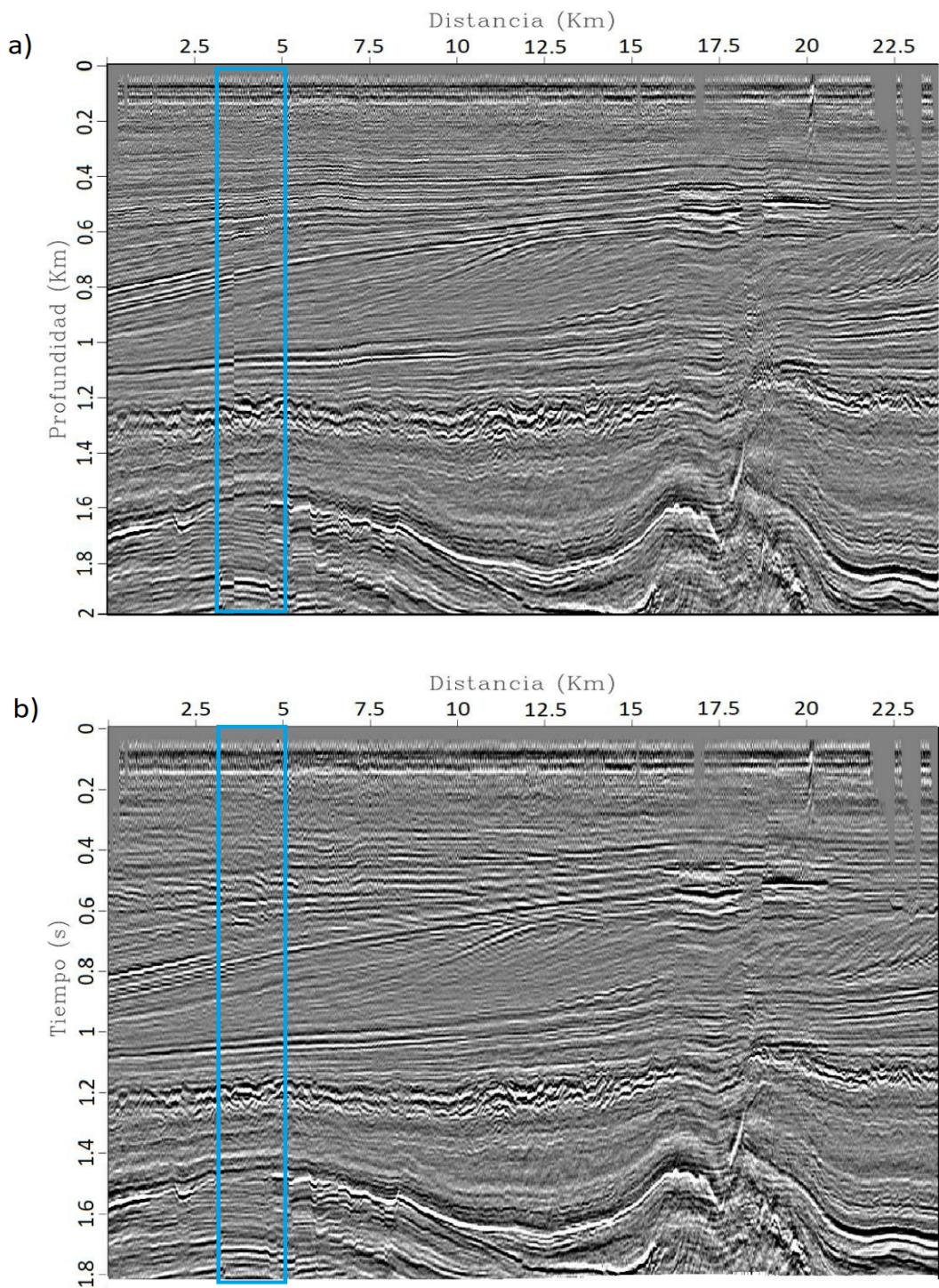


Figura 2.2: a) Sección sísmica longitudinal convertida de tiempo a profundidad utilizando Opendtect. b) Sección sísmica longitudinal convertida de profundidad a tiempo utilizando Opendtect.

### 2.3.2. Madagascar

La paquetería de Madagascar es de código abierto enfocada al análisis de datos multidimensionales y experimentos computacionales reproducibles. Su misión es proporcionar un entorno conveniente y potente para el desarrollo de proyectos sísmicos (Fomel et al., 2013).

El objetivo es el mismo, convertir datos sísmicos de tiempo a profundidad, además de generar modelos de velocidad. Algo a considerar es que graficar y manipular los datos en Madagascar requiere un mayor esfuerzo intelectual, debido a que para configurar varios aspectos es necesario tener conocimientos en programación basada en Python y comandos de la paquetería.

Los datos para realizar conversión tiempo a profundidad son la sección sísmica en tiempo (Fig. 2.1a) y la velocidad de intervalo (Fig. 2.1b), pero la sección en profundidad como resultado de estos datos tiene distorsión en la imagen, debido a las anomalías antes mencionadas en la sección de velocidad. Para solucionar ese problema genero dos modelos de velocidad, en una dimensión y en dos dimensiones.

Para generar el modelo 1D de velocidad de intervalo (Fig. 2.3) es necesario contar con los valores puntuales de velocidad con su respectiva profundidad. Hay que tener cuidado con la escala en los valores ingresados, debido a que Madagascar los considera “Km” y “s”, por lo tanto, realizo un cambio de escala para los datos.

Madagascar ofrece conversión de tiempo a profundidad mediante el comando

*sftimetodepth*. Está basado en la ecuación (2.6), (<http://ahay.org/blog/2013/07/01/program-of-the-month-sftime2depth/>), que relaciona velocidad, profundidad y tiempo. Una vez realizada la conversión tiempo a profundidad con el modelo de velocidad en una dimensión, obtengo datos sísmicos en profundidad (Fig. 2.4a). Los reflectores tienen un reacomodo ligeramente con mayor profundidad a diferencia de (Fig. 2.2a)

La paquetería también ofrece conversión de profundidad a tiempo mediante el comando *sfdepth2time*. El resultado obtenido es la sección en tiempo a partir de datos en profundidad (Fig. 2.4b). En comparación con datos sísmicos en tiempo originales (Fig. 2.1a), presenta un reacomodo de los reflectores en tiempos mayores, debido a la poca información de velocidad con la que cuenta el modelo.

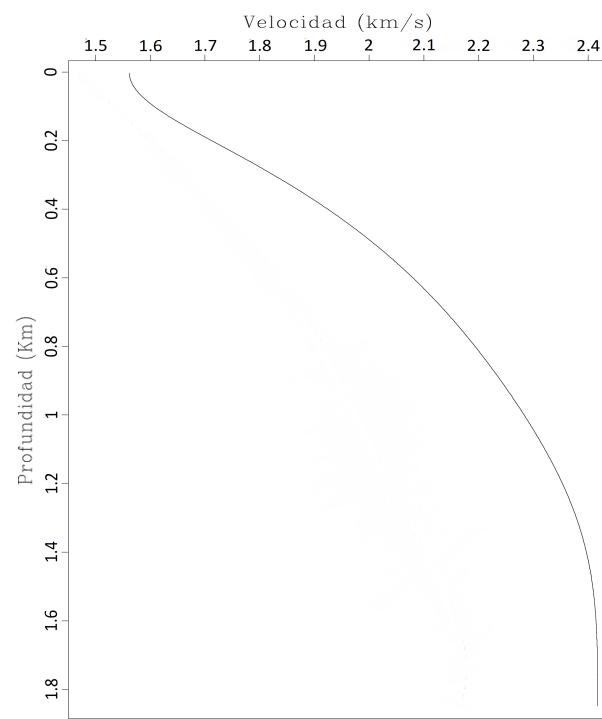


Figura 2.3: *Modelo de velocidad de intervalo en 1D generado por interpolación polinomial a partir de datos puntuales.*

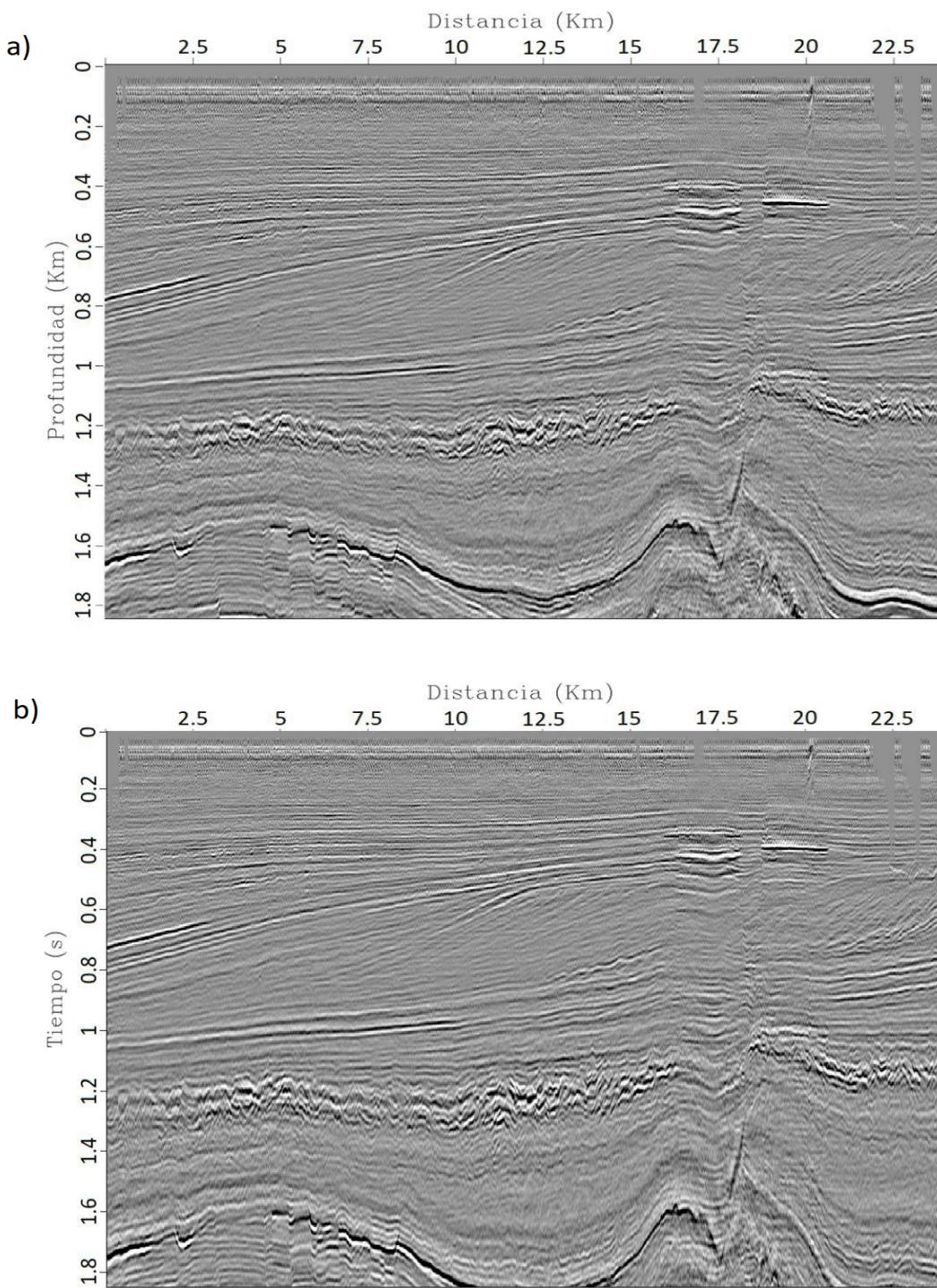


Figura 2.4: a) Sección sísmica longitudinal convertida de tiempo a profundidad utilizando el modelo 1D sintético de velocidad de intervalo con Madagascar. b) Sección sísmica longitudinal convertida de profundidad a tiempo utilizando el modelo sintético 1D de velocidad de intervalo con Madagascar.

La segunda propuesta de solución fue generar un modelo sintético de velocidad de intervalo en dos dimensiones (Fig. 2.5). Esto porque es más realista generar un modelo de velocidad que tome en cuenta la variación de la forma en las estructuras.

Una vez teniendo los datos de la sección sísmica en tiempo (Fig. 2.1a) y el modelo de velocidad de intervalo en dos dimensiones (Fig. 2.1), convierto los datos a profundidad (Fig. 2.6a). A diferencia de la sección en profundidad utilizando el modelo en una dimensión, en esta última los reflectores en la parte inferior tienen un reacomodo ligeramente con menor profundidad, y es más notable que entre los 0.8 km y 1.2 km que los reflectores tienen un diferente acomodo horizontal debido a que se considera la forma de la estructura para el modelo de velocidad. Posteriormente, realizo la conversión de profundidad a tiempo (Fig. 2.6b), a diferencia de la sección sísmica obtenida de profundidad a tiempo con el modelo de una dimensión. Aquí los reflectores están reacomodados a menor tiempo, con valores similares a la sección sísmica original en tiempo. Esto indica que utilizar este tipo de modelos de velocidad permite obtener una disminución de error del reposicionamiento de reflectores.

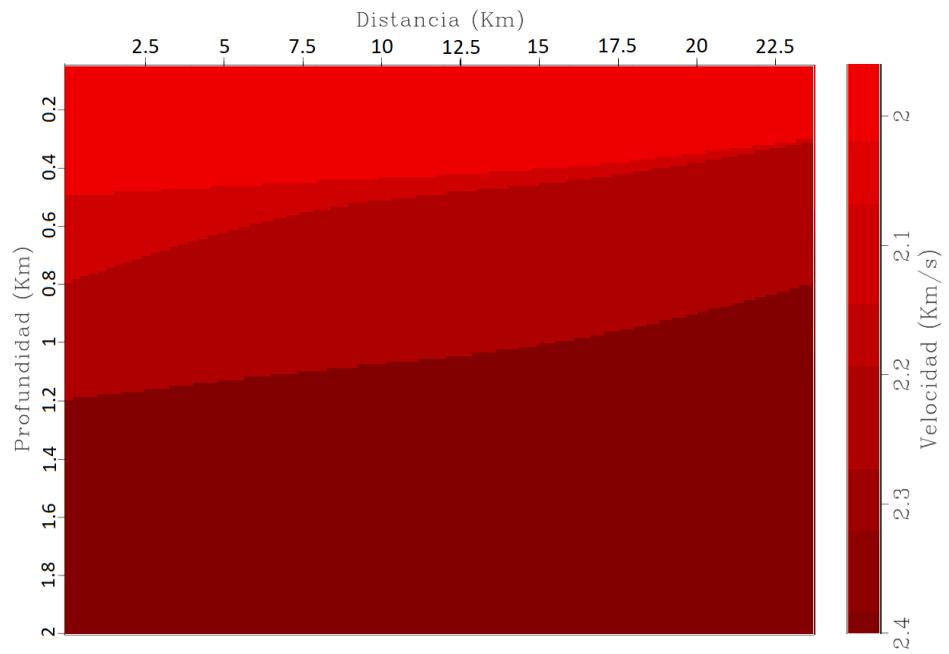


Figura 2.5: *Modelo sintético 2D de velocidad de intervalo (Valores de cada capa en orden de más superficial a más profunda; 1500 m/s, 1936 m/s, 1977 m/s, 2003 m/s, 2167 m/s).*

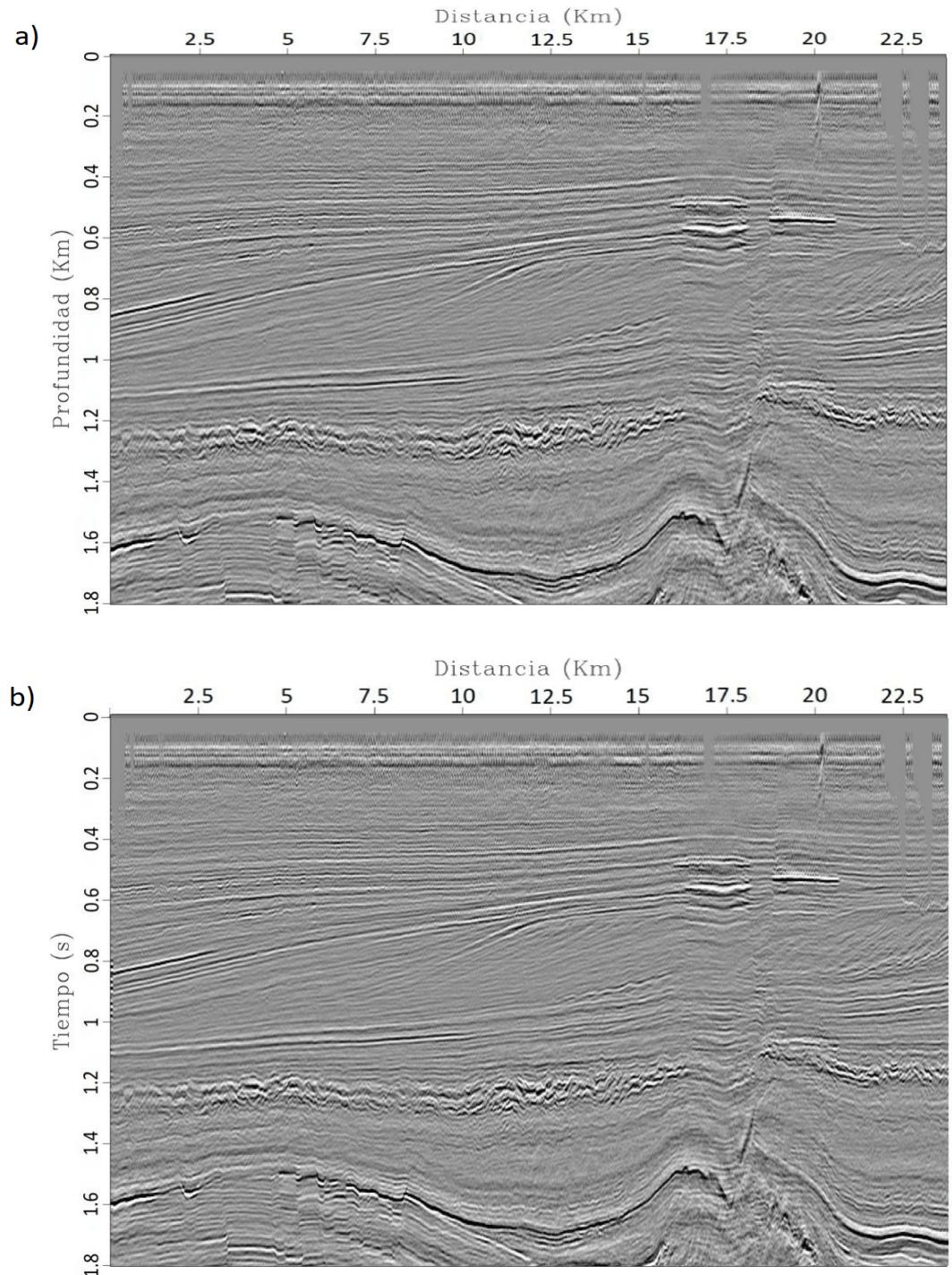


Figura 2.6: a) Sección sísmica longitudinal convertida de tiempo a profundidad utilizando el modelo sintético 2D de velocidad de intervalo con Madagascar. b) Sección sísmica longitudinal convertida de profundidad a tiempo utilizando el modelo sintético 2D de velocidad de intervalo con Madagascar.

### 2.3.3. Seismic Unix

Seismic Unix es una paquetería desarrollada por la Escuela de Minas de Colorado, con la meta de efectuar investigación y desarrollo sobre temas de sísmica. Seismic Unix el entorno es parecido a Madagascar. Sin embargo, es más sencillo de manipularlo para procesamiento y visualización de datos sísmicos. A diferencia de Madagascar, Seismic Unix no cuenta con una amplia gama de herramientas para realizar un procesamiento sísmico avanzado de datos. Seismic Unix puede considerar buena opción para personas que no están involucradas en el desarrollo de secuencias de comando y quieran contar con herramientas básicas para procesamiento de datos sísmicos.

La paquetería ofrece conversión tiempo a profundidad mediante el comando *suttod*, basado en la ecuación (2.6) (Stockwell y Cohen, 2002). Para realizar conversión tiempo a profundidad, los datos de entrada son la sección sísmica en tiempo (Fig. 2.1a), y para velocidad acepta dos formas para ingresar datos: mediante un archivo binario con los datos de velocidad o es ingresar muestras en profundidad con sus respectivos valores de velocidad de intervalo que son interpolados con interpolación sinusoidal. El resultado obtenido en profundidad (Fig. 2.7a) muestra que los reflectores tienden a repositionarse a menor tiempo, pero se puede considerar que es muy similar a los casos anteriores. Por último, realizo conversión de profundidad a tiempo con *sudtot*. como resultado obtengo (Fig. 2.7b), El reacomodo de los reflectores con respecto a la sección en tiempo original tiene un mayor desplazamiento, igual que sucedió en Madagascar con el modelo en una dimensión.

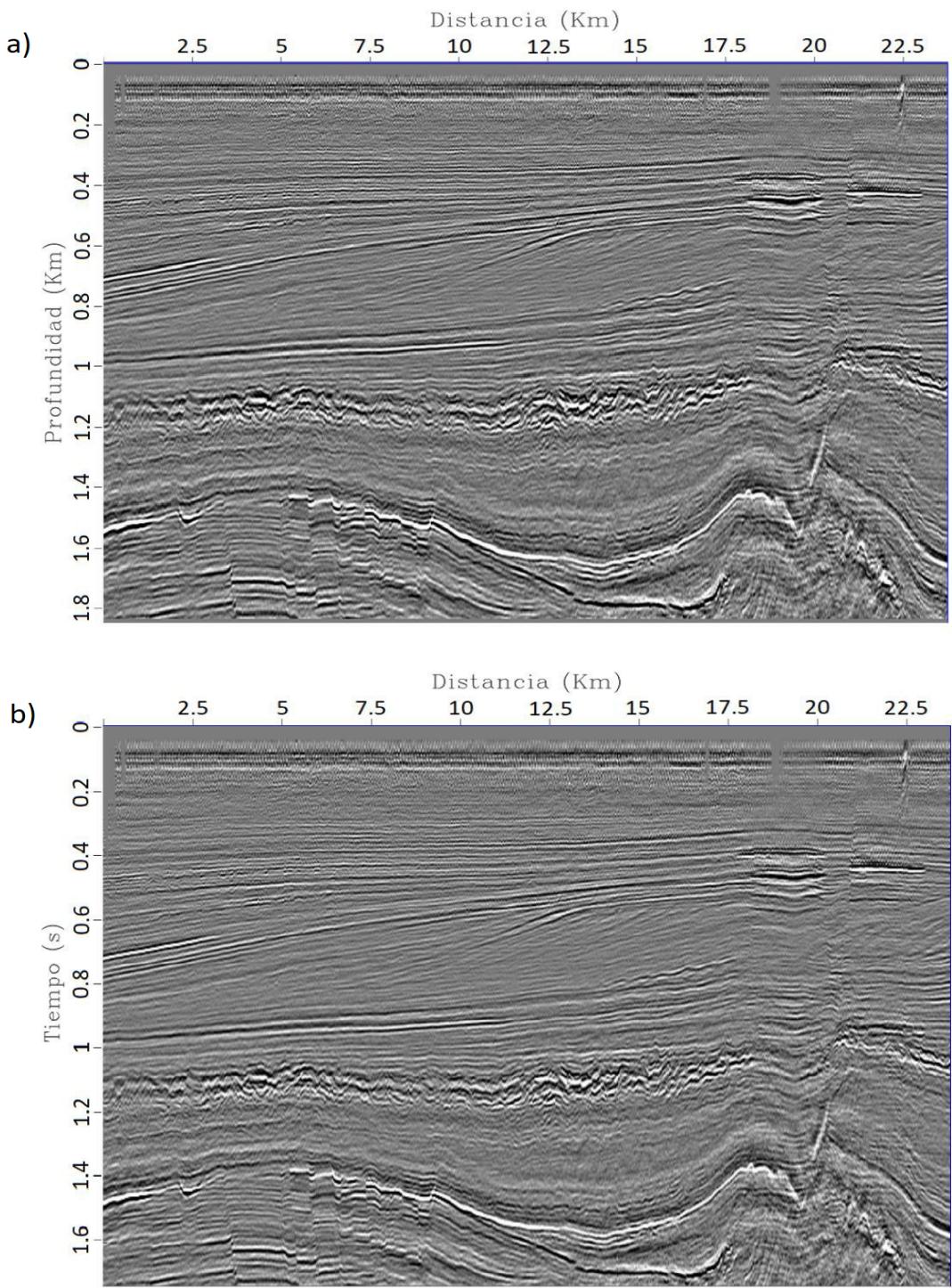


Figura 2.7: a) Sección sísmica longitudinal obtenida de tiempo a profundidad utilizando Seismic Unix. b) Sección sísmica longitudinal de profundidad a tiempo utilizando Seismic Unix.

# Capítulo 3

## Estimación de velocidad de intervalo

Algo fundamental para realizar la conversión tiempo a profundidad, es tener el modelo de velocidad de intervalo. Como se mencionó uno puede estimar la velocidad de intervalo a partir de los datos sísmicos, mediante ecuación de Dix y/o por rayos paraxiales.

El acceso a documentos reproducibles nos permite la trasmisión de conocimiento. Por ejemplo, estimar la velocidad de intervalo mediante rayos paraxiales (Cameron et al., 2007), es un trabajo de investigación que fue creado bajo la filosofía reproducible. Nos pone a disposición un algoritmo robusto y eficiente para obtener velocidad de intervalo y conversión tiempo a profundidad cuando existen cambios bruscos de velocidad. Por ejemplo, presencia de domos salinos, donde la sal debido a su alta velocidad da problemas con metodologías tradicionales para estimar velocidad de intervalo. A diferencia de la ecuación de Dix que en su teoría

considera capas horizontales y velocidades homogéneas. Y que es constantemente utilizada en la actualidad para estimar velocidad de intervalo a partir de datos sísmicos.

El propósito es estimar la velocidad de intervalo con cada una de las metodologías, para después observar cuales son las diferencias entre cada uno de los modelos de velocidad obtenidos por cada metodología. Posteriormente realizar conversión tiempo a profundidad, y ver cuáles son los efectos en el reposicionamiento de los reflectores utilizando cada uno de los modelos de velocidad de intervalo. Y demostrar que con documentos reproducibles es posible mejorar los resultados del modelo de velocidad de intervalo. Para exemplificar genero dos casos que son mostrados al final del capítulo.

### **3.1. Estimación de velocidad de intervalo utilizando ecuación de Dix**

Debe su nombre al geofísico norteamericano C. Hewitt Dix, que la desarrolló con el objetivo de determinar velocidad sísmica de intervalo a partir de datos sísmicos. Surge de por necesidad de tomar en cuenta la trayectoria de tiempo mínimo que recorre la onda al propagarse.

La ecuación de Dix (3.1) toma en cuenta la trayectoria de tiempo mínimo de viaje que tiene una onda al propagarse en el subsuelo hacia el reflector, a diferencia de la distancia mínima entre fuente y reflector. Para ello Dix propone el concepto  $V_{rms}$  (2.5) que es un promedio ponderado en vez de velocidad promedio

---

(2.3).

$$V_{int_n}^2 = \frac{V_{rms_n}^2 t_n - V_{rms_{n-1}}^2 t_{n-1}}{t_n - t_{n-1}}, \quad (3.1)$$

donde  $V_{int}$  es velocidad de intervalo de la capa,  $V_{rms_{n-1}}$  es velocidad cuadrática media hasta el reflector  $n-1$ ,  $V_{rms_n}$  es velocidad cuadrática media del siguiente reflector,  $t_{n-1}$  es tiempo de viaje hasta el  $n-1$  y  $t_n$  es tiempo de viaje hasta el reflector  $n$ .

### **3.2. Estimación de velocidad de intervalo utilizando teoría de rayos paraxiales**

Esta metodología parte del trabajo conjunto de Maria Cameron, Sergey Fomel y James Sethian (Cameron et al., 2007). Donde establecen una relación teórica entre velocidad de Dix y velocidad de intervalo, utilizando la teoría de rayos de imagen y teoría de rayos paraxiales (Popov y Psencik, 1978; Cerveny, 2001; Popov, 2002). Cuyo objetivo es construir un modelo de velocidad de intervalo más preciso a partir de la velocidad de Dix, y que el modelo de velocidad resultante, pueda ocuparse para realizar migración en profundidad y/o conversión tiempo a profundidad, con ello mejorar la imagen sísmica.

La técnica puede ser vista como una extensión de la fórmula de Dix (3.1) para medios lateralmente no homogéneos. Por mucho tiempo para obtener imágenes sísmicas del subsuelo se consideró que la velocidad dependía solamente de la

profundidad y que las estructuras eran horizontales o casi planas. Para estudiar estructuras más complejas y buzentas, Peter Hubral (1977) introdujo el concepto de rayo de imagen, que es definido como el rayo de tiempo mínimo desde un punto difractor en el subsuelo que llega perpendicular a la superficie.

Esta metodología establece una relación matemática entre la velocidad de intervalo, velocidad de Dix y dispersión geométrica. Para conceptualizar el problema, propongamos un rayo paraxial y en su centro un rayo central que está definido con un sistema de coordenadas  $(t, q_1, q_2)$ , y que para cada momento del tiempo podemos generar un plano perpendicular desde el rayo central, como se muestra en (Fig. 3.1). Debido a que los planos generados son perpendiculares al rayo central,  $e_n$  puede entenderse como el vector ortogonal del plano. Por lo tanto, localizar un punto en el espacio y observar la evolución del rayo central, queda expresado por  $r_m = r_0(t) + q_1 e_1 + q_2 e_2$ . Donde  $r_0$  es el punto alcanzado por el rayo central en el tiempo  $t$ .

Hasta este punto podemos localizar el rayo central respecto a la variación del tiempo. Pero para darle una mejor solución a este tipo de problemas, plantémoslo en formalismo Hamiltoniano, que es un sistema de coordenadas generalizadas en términos de momentos Q y P, respecto al tiempo. Ahora para comprender el flujo del rayo central y la evolución de su trayectoria queda expresada de la siguiente forma:

$$\frac{\partial e_1}{\partial t} = \frac{\partial v(t, q_1, q_2)}{\partial q_1} \quad \frac{\partial e_2}{\partial t} = \frac{\partial v(t, q_1, q_2)}{\partial q_2}. \quad (3.2)$$

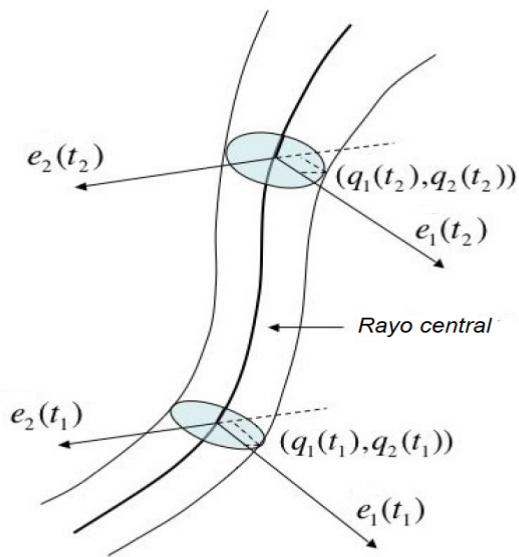


Figura 3.1: Representación gráfica del concepto de rayo central, para localizar y evaluar dos puntos representados por planos perpendiculares, en su variación respecto al tiempo.

De acuerdo con Popov y Psencik (1978), Cerveny (2001). La ecuación del rayo y su evolución en el tiempo para Q y P (3.2) es expresada de la siguiente manera:

$$\frac{\partial}{\partial t} \begin{pmatrix} q \\ p \end{pmatrix} = \begin{pmatrix} 0 & v_0^2 I_2 \\ \frac{1}{v_0} V & 0 \end{pmatrix} \begin{pmatrix} q \\ p \end{pmatrix}, \quad (3.3)$$

donde I es la matriz identidad 2x2,  $v_0$  es la velocidad a lo largo del rayo central y V es la matriz 2x2 de la velocidad. Y de acuerdo con Popov (2002), el valor absoluto del determinante de la matriz Q tiene un buen sentido geométrico. Por lo tanto, el valor absoluto de Q es la dispersión geométrica de la familia de rayos.

La relación para estimar la velocidad de intervalo esta dada por Ec. 3.4, que

surge del siguiente teorema: *Permitáse que un rayo de imagen llegue al punto en superficie de la Tierra  $x_0$  en el tiempo  $t_0$  desde un punto en el subsuelo. Supongamos que hay una familia telescopica de rayos alrededor del rayo de imagen que comienza perpendicular a la superficie de la Tierra que trazamos hacia atrás. Sea  $Q(x_0, t_0)$  la cantidad de dispersión geométrica de la familia telescopica de rayos evaluados en el momento  $t$ . Entonces, la velocidad de Dix es la relación de la velocidad sísmica y el valor absoluto de  $Q$  (3.4).*

$$VDix(x, t) = \frac{V(x, t)}{|Q(x, t)|}. \quad (3.4)$$

Para comprender el concepto de dispersión geométrica del rayo de imagen, supongamos que tenemos un rayo paraxial, que es considerado como un conjunto de rayos cercanos a el rayo central (Fig. 3.2). Trazar un rayo de imagen  $(x_0, t)$ ,  $x_0$  es el punto inicial en superficie,  $t$  es el tiempo total de viaje, llamemos este como rayo central. Considere un tubo de rayos a su alrededor, todos estos rayos comienzan desde un pequeño vecindario  $dx_0$ , del punto  $x_0$  perpendicular a la superficie de la Tierra. Por lo tanto, representan un fragmento de una onda plana que se propaga hacia abajo.

Considere el fragmento del frente de onda limitado por este tubo de rayos en el tiempo  $t_0$ . Dejar  $dq$  ser el fragmento de la tangente al frente de onda en el punto  $(x_0, t_0)$ , entonces  $Q$  es la razón de cambio expresado como  $dq/dx_{ij}$ .

Haciendo algunas consideraciones, la expresión de Ec. 3.3 del rayo y su evolución en el tiempo para  $Q$  y  $P$ , y la relación con la velocidad de Dix en dos

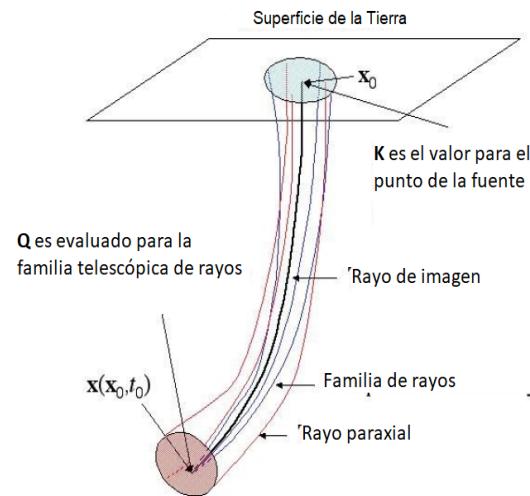


Figura 3.2: Ilustración del rayo paraxial, donde surge una familia de rayos en la superficie y son propagados hacia abajo hasta el punto  $(x_o, t_o)$ , donde son evaluados para conocer la dispersión geométrica de la familia de rayos.

dimensiones. Puede ser expresada de la siguiente manera.

$$\frac{Q}{f^2 Q^2} = -\frac{1}{fQ} \frac{fQ}{Q}. \quad (3.5)$$

Por lo tanto, la evolución en el tiempo de la dispersión geométrica de los rayos de imagen para dos dimensiones es expresada mediante una ecuación diferencial parcial (3.5). Donde  $f$  representa la velocidad de Dix y  $Q$  la dispersión geométrica. También se puede consultar la relación matemática para el caso de tres dimensiones (Cameron et al,2007).

Para resolver Ec. 3.5 se puede realizar mediante dos maneras: A) Por Método de Lax-Friedrichs, que es un método numérico para la solución de ecuaciones hiperbólicas en derivadas parciales basado en diferencias finitas. B) Por método espectral de Chebyshev que es un método numérico en el cual las condiciones de frontera son obtenidas desde un espectro (Cameron et al., 2008).

Una vez resuelta la Ec. 3.5 y obtener los valores de velocidad de intervalo  $V(x,t)$ , entonces para pasar los valores a  $V(x,z)$ . Proponen un método para conversión tiempo a profundidad, basado en una clase de algoritmo de Dijkstra, también conocido como algoritmo de trayectorias mínimas, que determina la trayectoria más corta dado un vértice origen al resto de los vértices de una rejilla. Para el algoritmo de conversión tiempo a profundidad, es aplicado un caso particular, conocido como “método de marcha rápida” (FMM, por sus siglas en inglés) propuesto por James Sethian (1996). El cual resuelve en forma discreta la ecuación de Eikonal. Una característica clave del *Fast marching method* es que lleva un cuidadoso orden de la evaluación para el tiempo de viaje. También algo a favor del FMM es la estabilidad al cálculo a pesar de cambios bruscos de velocidad en los modelos, incluso proporciona una optimización de los recursos computacionales (Alkhalifah y Fomel, 1997).

### 3.3. Caso I

Los datos utilizados para este caso pertenecen a un documento reproducible, alojado en el repositorio de Madagascar con el nombre “cmp-tp”. La información necesaria para obtener la velocidad de intervalo con las metodologías es la velocidad de Dix, que de acuerdo con (Byun et al., 1988) la velocidad de apilamiento

y velocidad RMS o de Dix son equivalentes. Por lo tanto, la velocidad obtenida para realizar el apilamiento es la que utilizo de entrada para obtener la velocidad de intervalo mediante ecuación de Dix y por rayos paraxiales.

Primero se obtiene velocidad de intervalo utilizando ecuación de Dix, mediante el comando *sfdix* que está basado en la ecuación de Dix (2.6). Mientras que, para obtener la velocidad de intervalo mediante rayos paraxiales, utilzo el comando *sfcameron2d*, que está basado en el algoritmo de (Cameron et al, 2007), donde estiman velocidad de intervalo desde velocidad de Dix, mediante dispersión geométrica de rayos. Cabe mencionar que esta metodología está disponible sólo en la paquetería de Madagascar.

Como resultado tengo la velocidad de intervalo mediante ecuación de Dix (Fig. 3.3a) y rayos paraxiales (Fig. 3.3b). Lo más destacado es que con ecuación de Dix hay un mayor contraste en valores de velocidad, están muy marcadas las altas y bajas velocidades, no se distinguen ligeros cambios de velocidad y las altas velocidades tienen mayor presencia. Mientras que la velocidad de intervalo estimada con rayos paraxiales tiene una distribución más uniforme de los valores de velocidad y es apreciable que respeta los ligeros cambios de los valores de velocidad.

Posteriormente, ejecuto conversión tiempo a profundidad de los datos sísmicos, utilizando cada uno de los modelos de velocidad. Como resultado, los datos sísmicos en profundidad para el modelo de velocidad obtenido con ecuación de Dix (Fig. 3.3c), los reflectores tienen un reposicionamiento ligeramente con mayor profundidad, y es más apreciable en la parte inferior, esto debido a los valores

más altos de velocidad que tiene el modelo. En comparación los datos sísmicos en profundidad para el modelo de velocidad generado con rayos paraxiales (Fig. 3.3d), los reflectores tienen un reposicionamiento más uniforme conforme incrementa la profundidad, y ligeramente a menor profundidad en comparación con el anterior. Esto se debe a una distribución más uniforme de velocidad que presenta el modelo.

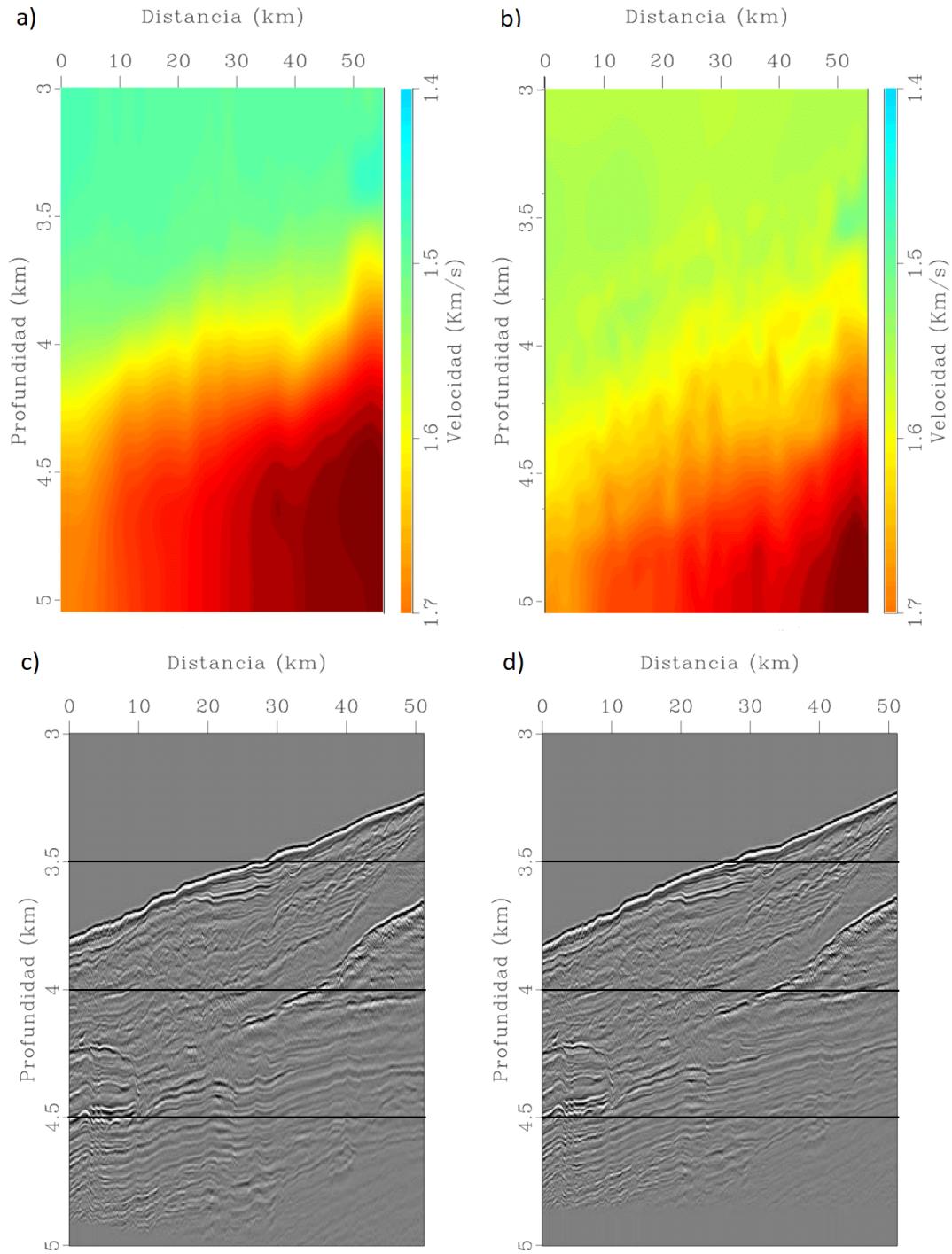


Figura 3.3: a) *Modelo de velocidad de intervalo utilizando Dix.* b) *Modelo de velocidad de intervalo utilizando rayos paraxiales.* c) *Datos sísmicos en profundidad para modelo de velocidad por Dix.* d) *Datos sísmicos en profundidad para modelo de velocidad por rayos paraxiales.*

### 3.4. Caso II

Nuevamente replico lo anterior, pero con diferentes datos, en esta ocasión los datos sísmicos llevan por nombre “beinew” y están alojados en el repositorio de Madagascar. Una vez obteniendo los valores de velocidad de apilamiento, convierto esos valores a velocidad de intervalo con cada una de las metodologías propuestas.

Para el modelo de velocidad de intervalo generado mediante ecuación de Dix (Fig. 3.4a), es notable la presencia de altas velocidades principalmente en la parte inferior y sin respetar los ligeros cambios de velocidad. En comparación con el modelo de velocidad de intervalo obtenido por la metodología de rayos paraxiales (Fig. 3.4b), donde los valores de velocidad están mejor distribuidos y las altas velocidades están concentradas en áreas más definidas, así como los ligeros cambios de velocidad son apreciables.

Para la sección sísmica en profundidad utilizando el modelo de velocidad estimado por ecuación de Dix (Fig. 3.4c), el reposicionamiento de los reflectores tiende a tener mayor profundidad e incrementa conforme va aumentando la profundidad. A diferencia de los datos sísmicos en profundidad para modelo de velocidad de intervalo utilizando rayos paraxiales (Fig. 3.4d), aquí los reflectores tienen un reposicionamiento más uniforme conforme incrementa la profundidad, y ligeramente a menor profundidad en comparación con el anterior, debido a la distribución más uniforme de velocidad del modelo.

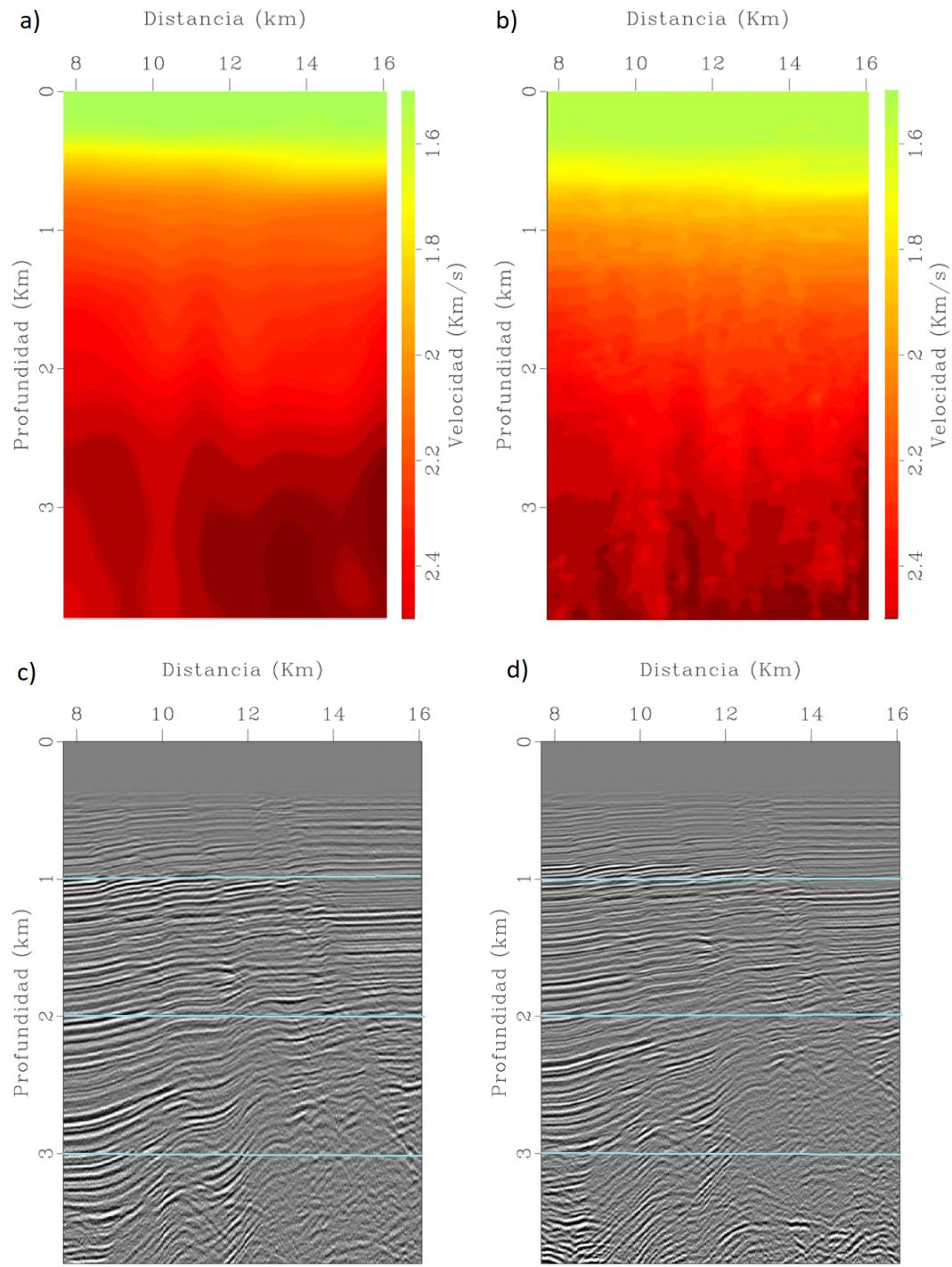


Figura 3.4: a) *Modelo de velocidad de intervalo utilizando Dix.* b) *Modelo de velocidad de intervalo utilizando rayos paraxiales.* c) *Datos sísmicos en profundidad para modelo de velocidad por Dix.* d) *Datos sísmicos en profundidad para modelo de velocidad por rayos paraxiales.*



# Capítulo 4

## Discusión

Los resultados presentados muestran que utilizar herramientas basadas en código abierto y reproducibles puede considerarse una alternativa para dar solución a la necesidad de conversión tiempo a profundidad para datos sísmicos.

Ejemplifco que se puede realizar conversión tiempo a profundidad con recursos computacionales que están al alcance de cualquier profesional, cabe resaltar que para este caso el trabajo fue ejecutado en un sistema de cómputo con procesador Intel Atom y 2 GB de RAM. Demostrando que no es necesario tener un sistema de cómputo sofisticado para obtener datos sísmicos en profundidad.

Con respecto a las paqueterías usadas (Opendtect, Madagascar, Seismic Unix), los resultados obtenidos para conversión de datos sísmicos en profundidad son adecuados como se mostró en los resultados. Aunque cada paquetería tiene particularidades; en cuestión de facilidad para el usuario Opendtect tiene ventaja y la interfaz gráfica es amigable. Seismic Unix requiere crear secuencias de comando

y tiene acceso a más herramientas para representación sísmica. Sin embargo, en valor técnico Madagascar tiene una gran ventaja, debido al acceso que se tiene a diferentes documentos reproducibles, ofrece gran variedad de soluciones técnicas para representación sísmica. Hay que tomar en cuenta que requiere un mayor esfuerzo intelectual y tener disciplina autodidacta para aprender lo necesario y poder manipular correctamente comandos y datos.

Utilizar este tipo de herramientas de código abierto nos permite acceder a trabajos e investigaciones actuales y que lleven el estado del arte para ofrecer una mejora en conversión tiempo a profundidad. Por ejemplo, los documentos reproducibles ponen a disposición técnicas como estimación de velocidad de intervalo mediante rayos paraxiales para mejorar el modelo de velocidad, que tiene un papel fundamental para convertir datos sísmicos de tiempo a profundidad. Recurrir a este tipo de herramientas libres es útil para actividades de docencia, investigación y práctica profesional. Sus beneficios académicos, técnicos y económicos son importantes, así como conocer el computo implicado. Por ello debemos difundir el uso y generación de herramientas de este tipo.

# Conclusiones

- Herramientas reproducibles y de código abierto nos permite ejecutar conversión tiempo a profundidad de manera confiable y profesional, así como acceder a trabajos reproducibles que permiten tener mejores resultados.
- Permiten desarrollar, adaptar y transferir tecnología de interés en la industria petrolera a bajo costo, no sustituyen a herramientas comerciales (reducen su dependencia). Así como a investigadores, profesores, estudiantes y profesionales, les permite trabajar con herramientas a las que no podrían tener acceso de otra manera.
- Puede advertirse que este tipo de herramientas irán cobrando mayor relevancia en el desarrollo de trabajos en representación sísmica y otras áreas, es importante ser consciente de la tendencia creciente de estas herramientas para obtener el mayor beneficio.
- Debido a las características propias de estas herramientas pueden considerarse una opción para capacitación de profesionales y de estudiantes en centros de investigación.



# Agradecimientos

A Sergio Chávez Pérez y a Liliana Vargas Meleza por su constante apoyo, críticas y sugerencias para la realización de este trabajo, que han sido relevantes en mi formación profesional.

A Douglas Gómez y Edgar Espinoza, cuyas observaciones fueron importantes para completar este trabajo.

Al Instituto Mexicano del Petróleo, por el apoyo brindado durante mi estancia y por el conocimiento adquirido en esa institución.

A la Benemérita Universidad Autónoma de Puebla por el compromiso que brinda a la educación, formación de profesional y la oportunidad de realizar mis estudios.

A mis sinodales Mtro. Luis Fernando Gómez Ceballos y Mtro. José Serrano Ortíz, así como a mi asesor interno Mtro. José Castillo Román. Muchas gracias por sus consejos y tiempo brindado en el desarrollo de esta tesis.

A mi madre y mi familia, así como a mis amigos con los que conviví y me hicieron más grata esta etapa de vida.

# Referencias

- Alkhalifah, T., y S. Fomel, 1997, Implementing the fast marching eikonal solver: Spherical versus cartesian coordinates: Stanford Exploration Project, 149–171.
- Byun, B. S., Tran, H. V., y Carter, M. D., 1988, An analysis of statistical average velocities in a horizontally layered Earth: Geophysics, v. 53, No. 11, 1488-1489.
- Cameron, M., 2007, Seismic velocity estimation from time migration, University of California, Berkeley.
- Cameron, M., Fomel, S., y Sethian, J, 2008, Time-to-depth conversion and seismic velocity estimation using time-migration velocity: Geophysics, v.73, No. 5, 205-210.
- Claerbout J.F., 1985, Imaging the Earth's Interior: Stanford University, Palo Alto, CA, EEUU.
- Clearboust, J. F., y Fomel, S., 2009, Reproducible research, Computing in science and engineering, IEEE, No. 9, 5-7.
- Dix, C. H., 1955, Seismic velocities from surface measurements: Geophysics, v. 10, No.20, 68-86.
- Donoho, D.L., Maleki, A., Morteza, S., Ur Rahman, I. Stodden, V., 2009, Reproducible research in computational harmonic analysis: Computing in Science and Engineering, 11, 8-18.
- Filpo, E., Portugal, R., Zago, N., Cunha, P. M., Vicentini, A., y Carbonevi, J. L., 2016, Image-ray concept as the key to 20 years of success of time-to-depth conversion

- in Petrobras: The Leading Edge, v. 4, No.35, 323-328.
- Fomel, S., Sava, P., Vlad, I., Liu, Y., y Bashkardin.,2013, Madagascar: Open-source software project for multidimensional data analysis and reproducible computational experiments: Journal of Open Research Software, v.1, No.8.
- González, Verdejo., y Chávez, Pérez., 2012, Reproducibilidad en representación sísmica. Ingeniería, investigación y tecnología, v.13, No.2, 185-197.
- Hall, M.,2016, Eight Strategies for Revolutionizing the Open Geoscience Ecosystem. In 78th EAGE Conference and Exhibition: 2016-Workshops.
- Hosken, J. W. J., y Deregowski, S. M., 1985, Tutorial: Migration strategy. Geophysical prospecting, v.33, No.1, 1-33.
- Hubral, P.,1977, Time migration—Some ray theoretical aspects. Geophysical Prospecting, v. 25, No.4, 738-745.
- Hubral, P., y Krey, T.,1980, Interval velocities from seismic reflection time measurements: Society of Exploration Geophysicists.
- Li, S., y Fomel, S.,2015, A robust approach to time-to-depth conversion and interval velocity estimation from time migration in the presence of lateral velocity variations: Geophysical Prospecting, v.63, No.2, 315-337.
- Robein. E., 2003, Velocities, Time-Imaging and Depth-imaging in refraction seismic principles and methods: EAGE Publications.
- Sadala Valente, L. D. S., Santos, H. B., Costa, J. C., y Schleicher, J.,2017, Time-to-depth conversion and velocity estimation by image-wavefront propagation: Geophysics, v.82, No.82.
- Schlumberger,1989, Principios y Aplicaciones de la Interpretación de Registros: MCA Marketing.
- Sethian, J. A.,1996, A fast marching level set method for monotonically advancing fronts: Proceedings of the National Academy of Sciences, v.93, No.4, 1591-1595.
- Sheriff, R. E.,2002, Encyclopedic dictionary of applied geophysics: Society of expl-

ration geophysicists.

Stockwell, J. W., y Cohen, J. K., 2002, The new SU user's manual. Center for Wave Phenomena, Colorado School of Mines, Golden, USA, 3.

Taner, M. T., y Koehler, F., 1969, Velocity spectra—digital computer derivation applications of velocity functions: Geophysics, v.34, No.6, 859-881.

Vargas, M.L., Nieto-B.J. et al., 2017, Reproducibilidad computacional y herramientas de código abierto en sismología de exploración y producción, Congreso Mexicano del Petróleo, 2017, Puebla.

Yilmaz, O., 1987, Seismic Data Processing: Society of Exploration Geophysicists, Tulsa.



# Apéndice A

```
1
2 #####CONVERSION TIEMPO A PROFUNDIDAD CON MODELO EN UNA DIMENSION#####
3
4 from rsf.proj import *
5
6 #Seccion sismica
7 Flow( 'seis',
8       'sismica.sgy',
9       'segyread tfile=tfile1.rsf hfile=seis.asc bfile=seis.bin endian
=y')
10
11 Flow( 'subseis', "seis" ,
12      '',
13      put n2=950 label2="Inline 2D"
14      '')
15 #window min1=0.004 max1=1.8
16
17 Result( 'subseis' ,
18      '',
19      byte gainpanel=all |
20      grey title="Seccion en tiempo original" label1="Tiempo" unit1="s"
label2="Seccion longitudinal" labelsz=6
```

```
21   ''')
22
23 # Velocidad RMS 1D
24 Flow( 'vrms.asc' ,None ,
25       '',
26       echo
27       0.05 1.500
28       0.858 1.936
29       1.026 1.977
30       1.125 2.003
31       1.848 2.167
32
33 n1=2 n2=6 in=$TARGET
34 data_format=ascii_float
35 ''')
36
37 Flow( 'vrms' , 'vrms.asc subseis' ,
38       '',
39       dd form=native |
40       spline pattern=${SOURCES[1]}
41       '')
42
43 # Velocidad de intervalo
44 Flow( 'vintdix' , 'vrms' , 'dix rect1=50')
45
46 Result( 'vintdix' , 'vrms vintdix' ,
47         '',
48         cat axis=2 ${SOURCES[1]} |
49         graph wanttitle=n dash=1,0
50         label1=Tiempo unit1=s
51         label2=Velocidad unit2=km/s
```

```

52     transp=n yreverse=n labelsz=6
53     ''')
54
55 # Conversion tiempo a profundidad
56 Flow( 'ttoddix ', 'subseis vintdix ',
57     '',
58     time2depth velocity=${SOURCES[1]} intime=n |
59     put label1=Profundidad unit1=km
60     ''')
61
62 Result( 'imgttoddix ', 'ttoddix ', 'grey scalebar=n label1=Profundidad
63     unit1="Km" label2="Seccion longitudinal" title="Seccion en
64     profundidad" labelsz=6 ')
65
66 #Conversion profundidad a tiempo
67
68 Flow( 'dtotdix ', 'ttoddix vintdix ',
69     '',
70     time2depth velocity=${SOURCES[1]} intime=n |
71     put label1=Tiempo unit1=s
72     ''')
73
74 Result( 'imgdtotdix ', 'dtotdix ', 'grey scalebar=n label1=Tiempo unit1="
75     s" label2="Seccion longitudinal" title="Seccion en tiempo
76     reconstruida" labelsz=6 ')
77
78 from rsf.proj import *

```

```
79
80 xmax = 23.75
81 zmax = 2
82
83 layers = ((0.05,0.05,0.05,0.05),
84             (0.5,0.45,0.40,0.3),
85             (0.8,0.55,0.45,0.31),
86             (1.20,1.1,1.0,.80))
87
88 velocities = (1.500,
89                 1.936,
90                 1.997,
91                 2.050,
92                 2.167)
93
94 def arr2str(array,sep=' '):
95     return string.join(map(str,array),sep)
96
97 vstr = arr2str(velocities,' ')
98
99 n1 = len(layers[0])
100 n2 = len(layers)
101
102 Flow('layers.asc',None,
103       '',
104       echo %$,
105       n1=%d n2=%d o1=0 d1=%g
106       data_format=ascii_float in=$TARGET
107       '' % (string.join(map(arr2str,layers),' '),
108             n1,n2,xmax/(n1-1)))
109 Flow('layers','layers.asc',dd form=native)
```

```

110
111 d = 0.0101 # non-round for reproducibility
112
113 Flow( 'refs' , 'layers' ,
114     'spline o1=0 d1=%g n1=%d' % (d,int(1.5+xmax/d)))
115 Flow( 'dips' , 'refs' , 'deriv scale=y')
116
117 Flow( 'mod2D' , 'refs' ,
118     '',
119     'unif2 d1=%g n1=%d v00=%s
120     '' % (d,int(1.5+zmax/d),vstr))
121
122 Result( 'imgmod2D' ,
123     'mod2D' ,
124     '',
125     window min1=.05 |
126     grey color=j title="Modelo de velocidad de intervalo
127     sintetico 2D"
128     titlesz=8 labelsz=6 scalarbar=y labelsz=6 bias=1
129     label1="Profundidad" unit1="Km"
130     label2="Distancia" unit2="Km"
131     '' )
132 #Configuracion y Visualizar Seccion sismica en tiempo
133
134 Flow( 'seis' ,
135     'sismica.sgy' ,
136     'segypread tfile=tfile1.rsf hfile=seis.asc bfile=seis.bin endian
137     =y')
138 Flow( 'subseis' , "seis" ,

```

```

139   '',
140 put n2=950  label2="Inline 2D"
141 '')
142
143 Result('subseis',
144 '',
145 byte gainpanel=all |
146 grey title="Seccion en tiempo original" label1="Tiempo" label2="
147     Seccion longitudinal" unit1="s" labelsz=6
148 ')
149 #Conversion tiempo a profundidad
150
151 Flow('ttod','subseis mod2D',
152       '',
153       time2depth velocity=${SOURCES[1]} intime=n |
154       put label1=Profundidad unit1=km
155       ')
156
157 Result('imgttod','ttod',
158       'window max1=1.8 | grey scalebar=n labelsz=6 label1=
159       Profundidad label2="Seccion longitudinal" unit1="Km" title="
160       Seccion en profundidad"')
161
162 #Conversion profundidad a tiempo
163
164 Flow('dtot','ttod mod2D',
165       '',
166       depth2time velocity=${SOURCES[1]} intime=n |
167       put label1=Tiempo unit1=s
168       ')

```

```
167
168 Result( 'imgdtot' , 'dtot' , 'window max1=1.8 | grey scalebar=n labelsz=6
    label1="Tiempo" label2="Seccion longitudinal" unit1="s" title="
    Seccion en tiempo reconstruida" ')
169
170 End()
171
172 #####CONVERSION TIEMPO A PROFUNDIDAD CON SEISMIC UNIX#####
173
174 #!/bin/bash
175
176 suwind < sismica.su key=tracr min=1 max=850 > inline.su
177
178 sximage < inline.su title="Seccion en tiempo"
179
180 #Conversion tiempo–profundidad
181
182 suttoz < inline.su t=0.05,0.858,1.026,1.125 v=1.5,1.960,2.175,2.260
    > inlined.su
183
184 sximage < inlined.su title="Seccion de tiempo a profundidad"
185
186 #Conversion profundidad – tiempo
187
188 suztot < inlined.su z=.0037,.0830,1.012,1.5 v=1.5,1.960,2.175,2.260
    > inliner.su
189
190 sximage <inliner.su title="Seccion de profundidad a tiempo"
```



# Apéndice B

```
1
2 #####ESTIMACION DE VELOCIDAD INTERVALICA UTILIZANDO RAYOS PARAXIALES I
3 ######
4 from rsf.proj import *
5
6 # Obtener datos
7 Fetch('cmps-tp.HH', 'blake')
8
9 # CMP (common midpoint) gathers
10 Flow('cmps', 'cmps-tp.HH',
11       'dd form=native | reverse which=2')
12 # one CMP
13 Flow('cmp', 'cmps',
14       '',
15       window f3=950 n3=1 max1=6 |
16       put o2=0.0 d2=1
17       '')
18 Plot('cmp',
19       '',
20       grey title="CMP gather"
21       unit1=s label1=Offset unit2=km label2=km labelsz=12
```

```

22      ''')
23
24 # Near-offset
25 Result( 'noff' , 'cmps' ,
26          '',
27          window n2=1 n3=1024 |
28          grey title="Near Offset Section"
29          unit1=s label2=Distancia unit2=km label1=Tiempo labelsz=12
30          ''')
31
32 # offset maps
33 Flow( 'off1' ,None , 'math n1=24 o1=0.4 d1=0.1 output=x1' )
34 Flow( 'off2' ,None , 'math n1=24 o1=2.8 d1=0.05 output=x1' )
35 Flow( 'off' , 'off1 off2' , 'cat axis=1 ${SOURCES[1]}' )
36 Flow( 'offs' , 'off' , 'spray n=111' )

37
38 # Analisis de velocidad
39
40 vscan = ''
41 vscan offset=${SOURCES[1]}
42 v0=1.4 nv=61 dv=0.005 half=n semblance=y
43 ''
44 pick = 'pick rect1=20 rect2=3 vel0=1.5'

45
46 # Semblanza para una CMP
47 Flow( 'vscan' , 'cmp off' ,vscan )
48 Plot( 'vscan' ,
49        '',
50        grey color=j allpos=y title="Velocity Scan"
51        unit1=s label2=Velocity unit2=km/s pclip=100 labelsz=12
52        ''')

```

```

53
54 # pick maximum semblance for one CMP
55 Flow( 'pick' , 'vscan' , pick)
56 Plot( 'pick0' , 'pick' ,
57      '',''
58      graph transp=y yreverse=y min2=1.4 max2=1.7
59      plotcol=7 plotfat=10 pad=n wanttitle=n wantaxis=n labelsz=12
60      '','')
61 Plot( 'pick1' , 'pick' ,
62      '',''
63      graph transp=y yreverse=y min2=1.4 max2=1.7
64      plotcol=0 plotfat=1 pad=n wanttitle=n wantaxis=n labelsz=12
65      '','')
66 Plot( 'vscan2' , 'vscan pick0 pick1' , 'Overlay' )
67
68 # Semblanza cada 10th CMP
69 Flow( 'vscans' , 'cmps offs' , 'window j3=10 | ' + vscan)
70
71 # Seleccion de semblanza para cada 10th CMP
72 Flow( 'picks0' , 'vscans' , pick)
73
74 # interpolar
75 Flow( 'picks' , 'picks0' ,
76      'transp | remap1 n1=1105 d1=0.05 o1=0 | transp')
77 Result( 'picks' ,
78      '',''
79      grey color=j scalebar=y barreverse=y
80      allpos=n bias=1.47
81      barlabel=Velocidad barunit=Km/s
82      minval=1.5 maxval=1.7 labelsz=12
83      title="Velocidad de apilamiento"

```

```

84     label1=Tiempo unit1=s label2=Distancia unit2=km
85     ''')
86
87 # Normal moveout and apilamiento
88 nmo = ''
89 nmo offset=${SOURCES[1]}
90 velocity=${SOURCES[2]} half=n
91 ''
92
93 Flow( 'nmo' , 'cmp off pick' ,nmo)
94 Plot( 'nmo' ,
95       '',
96       grey title="Normal Moveout"
97       unit1=s label2=Offset unit2=km labelsz=12
98       ''')
99 Result( 'nmo' , 'cmp vscan2 nmo' , 'SideBySideAniso' )
100
101 Flow( 'nmos' , 'cmps off picks' ,nmo)
102 Flow( 'stack' , 'nmos' , 'stack' )
103 Result( 'stack' ,
104       '',
105
106       grey title="Seccion apilada" labelsz=12
107       label1=Tiempo unit1=s label2=Distancia unit2=km
108       ''')
109
110 # Convertir tiempo a profundidad
111 ######
112
113 # Dix
114 Flow( 'semb' , 'vscans picks0' , 'slice pick=${SOURCES[1]} ')
```

```

115 Flow( 'vel0' , 'picks0 semb' ,
116      'dix rect1=20 rect2=2 weight=${SOURCES[1]} ')
117
118 Flow( 'vel' , 'vel0' ,
119       'transp | remap1 n1=1105 d1=0.05 o1=0 | transp')
120 Plot( 'vel' ,
121       '',
122       grey color=j scalebar=y barreverse=y
123       allpos=n bias=1.52
124       barlabel=Velocidad barunit=Km/s
125       minval=1.4 maxval=1.7 labelsz=12
126       title="Velocidad de intervalo por DIX"
127       label1=Profundidad unit1=km label2=Distancia unit2=km
128       ''')
129
130 Flow( 'image' , 'stack vel' ,
131       '',
132       time2depth velocity=${SOURCES[1]}
133       intime=y dz=0.005 nz=1001
134       '')
135 Plot( 'image' ,
136       '',
137       window n2=1024 min1=3 | grey title="Image with Dix" labelsz
138       =12
139       label1=Profundidad unit1=km label2=Distancia unit2=km
140       ''')
141 #Cameron
142
143 Flow( 'vcam' , 'picks' , ' transp | cameron2d method=cheb nz=11000 dz
144       =.005 ')

```

```

144 Flow( 'vcam2' , 'picks' , 'cameron2d method=cheb nz=11000 dz=.005 ' )
145
146 Plot( 'vcam' ,
147     ' ' ,
148     window max2=1.8 | grey color=j scalebar=y barreverse=y transp
149     =n
150     allpos=n bias=1.47
151     barlabel=Velocidad barunit=Km/s
152     minval=1.4 maxval=1.7 labelsz=12
153     title="Velocidad de intervalo por Cameron"
154     label1=Distancia unit1=km label2=Profundidad unit2=km
155     ' ' )
156
157
158 Flow( 'vcamt' , 'vcam' , 'transp' )
159
160
161 Flow( 'vcam0' , 'vcamt' ,
162     'remap1 n1=625 d1=.004 o1=4' )
163
164 Flow( 'vcam00' , 'vcam0' ,
165     'transp | remap1 n1=11000 d1=0.005 o1=0 | transp' )
166
167
168 Result( 'vcam00' ,
169     ' ' ,
170     grey color=j scalebar=y barreverse=y transp=n
171     bias=1.4 allpos=n labelsz=12
172     title="Interval Velocity by Cameron"
173     label1=Profundidad unit1=km label2=Distancia unit2=km
174     ' ' )
175
176
177 Flow( 'imagecam' , 'stack vcam2' ,
178     ' ' ,
179     ' ' )

```

```

174     time2depth velocity=${SOURCES[1]}
175     intime=y dz=0.005 nz=1001
176     ,,,)
177 Plot('imagecam',
178     ,,
179     window n2=1024 min1=3 | grey title="Seccion en profundidad
180     con Cameron" labelsz=12
181     label1=Profundidad unit1=km label2=Distancia unit2=km
182     ,,,)
183 #
184 # vintdix vs vintcam
185 Result('Fint','vel vcam','SideBySideAniso')
186
187 # Seccion en profundidad con Vint y con Vintcam
188
189 Result('Fsis','image imagecam','SideBySideAniso')
190
191 End()
192
193 #####ESTIMACION DE VELOCIDAD INTERVALICA UTILIZANDO RAYOS PARAXIALES
194 II#####
195 from rsf.proj import *
196 def Grey(data,other):
197     Result(data,'grey label2=Midpoint unit2="km" clip=1.84785e
198 +06 label1=Time unit1="s" title="" wherexlabel=t wanttitle=y
199     wheretitle=b screenratio=1.4 max1=2.5 % %other)
200

```

```

201 Flow( 'gulf' , 'beinew.HH' ,
202   ' ,
203   dd form=native | put
204   label1=Time unit1=s
205   label2=Half-Offset unit2=km
206   label3=Midpoint unit3=km | window f2=4
207   ' ,
208   ' )
209
210 # Display
211 Result( 'gulf' ,
212   ' ,
213   window min1=0.5 max1=3.8 max2=1.5 | byte |
214   transp plane=23 |
215   grey3 flat=n frame1=500 frame2=160 frame3=3
216   title="Gulf of Mexico Data" point1=0.8 point2=0.8
217   ' ,
218   ' )
219 # Escaneo de velocidad
220 Flow( 'vscan-gulf' , 'gulf' ,
221   'vscan v0=1.5 dv=0.02 nv=51 semblance=y' ,
222   split=[3,250], reduce='cat')
223
224 Result( 'vscan-gulf' ,
225   ' ,
226   byte allpos=y gainpanel=all pclip=100 |
227   transp plane=23 |
228   grey3 flat=n frame1=750 frame2=100 frame3=25
229   label1=Time unit1=s color=j labelsz=12
230   label3=Velocity unit3=km/s
231   label2=Midpoint unit2=km

```

```

232     title="Velocity Scan (traditional)" point1=0.8 point2=0.8
233     ''')
234
235 # Seleccion de velocidad
236 Flow( 'vnmo-gulf' , 'vscan-gulf' , 'pick rect1=5 rect2=5' )
237
238 Result( 'vnmo-gulf' ,
239     '',
240     grey color=j allpos=n bias=1.4
241     scalebar=y barreverse=y barunit=km/s barlabel=Velocidad
242     minval=1.5 maxval=2.5 labelsz=12
243     label2=Distancia unit2=km label1=Tiempo unit1=s
244     title="Velocidad NMO"
245     ''')
246
247 # Apilamiento
248 Flow( 'nmo-gulf' , 'gulf vnmo-gulf' , 'nmo velocity=${SOURCES[1]} ')
249 Flow( 'stack-gulf' , 'nmo-gulf' , 'stack' )
250
251 Result( 'stack-gulf' ,
252     '',
253     grey labelsz=12
254     label2=Distancia unit2=km label1=Tiempo unit1=s
255     title="Gulf of Mexico Data" point1=0.8 point2=0.8
256     ''')
257
258 #DIX
259 Flow( 'vintdix' , 'vnmo-gulf' , 'dix rect1=10 rect2=10' )
260
261 Flow( 'vintdixd' , 'vintdix vintdix' , 'time2depth velocity=${SOURCES
[1]} intime=y' )

```

```

262
263 Plot( 'vintdixd' ,
264   '',
265 window max1=3.8 | grey color=j allpos=n bias=1.4
266 scalebar=y barreverse=y barunit="Km/s" labelsz=12
267 minval=1.5 maxval=2.5 barlabel=Velocidad
268 label2=Distancia label1=Profundidad unit1=Km
269 title="Velocidad e intervalo en tiempo"
270 '')
271
272 Flow( 'ttoddix' ,
273   'stack-gulf vintdixd' ,
274   '',
275 time2depth velocity=${SOURCES[1]} intime=n
276 ')
277
278 #Grafica
279 Plot( 'ttoddix' ,
280   '',
281   '',
282 window max1=3.8 | grey
283 label2=Distancia unit2=Km label1=Profundidad unit1=km labelsz=12
284 title="Seccion sismica en profundidad utilizando Vintdix"
285 ')
286
287 #####CAMERON
288 Flow( 'vintcam x0 t0' , 'vnmo-gulf' , 'transp | cameron2d method=cheb dz
289 =0.005 nz=1001 | put label2=Profundidad unit2=Km ')
290
291 #Grafica
292 Plot( 'vintcam' ,

```

```

292
293   ''
294 transp | window max1=3.8 | grey color=j allpos=n bias=1.4
295 scalebar=y barreverse=y barunit=Km/s
296 minval=1.5 maxval=2.5 barlabel=Velocidad labelsz=12
297 label2=Distancia unit2=Km label1=Profundidad unit1=km
298 title="Seccion sismica en profundidad utilizando Vintcam"
299 ''
300
301 Flow( 'vintcam2' , 'vintcam' , 'transp' )
302
303 Flow( 'ttodcam' ,
304       'stack-gulf vintcam2' ,
305     '' )
306 time2depth velocity=${SOURCES[1]} intime=n
307 ''
308
309 #Grafica sismica
310 Plot( 'ttodcam' ,
311
312   ''
313 window max1=3.8 | grey scalebar=n
314 label2=Distancia unit2=Km label1=Profundidad unit1=km labelsz=12
315 title="Seccion sismica en profundidad utilizando Vintcam"
316 ''
317
318 # Rayos
319 Flow( 'vd x z' , 'vintcam' , 've2d nt=1000 dt=.004 x=${TARGETS[1]} z=${
320           TARGETS[2]} ')
321 Plot( 'rayos' , 'x z' ,

```

```
322      ''
323      cmplx ${SOURCES[1]} | transp | window j2=10 |
324      graph wanttitle=n yreverse=y labelsz=12
325      plotcol=7 scalebar=y wantaxis=n
326      bartype=v
327      ''')
328
329 Result('vintrayos','vintcam rayos','Overlay')
330
331 # Comparacion de velocidades
332
333 Result('Fint','vintdixd vintcam','SideBySideAniso')
334
335 # Comparacion de secciones en profundidad
336
337 Result('Fsis','ttoddix ttodcam','SideBySideAniso')
338
339 End()
```

Los códigos también están disponibles en <https://github.com/javibash> o puede comunicarse a nieto.jb@gmail.com para alguna información.

Texto generado en LATEX.