



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA
EXPLORACIÓN Y EXPLOTACIÓN EN RECURSOS NATURALES-
EXPLORACIÓN

**DESARROLLO DE UN ALGORITMO EN PARALELO DE INVERSIÓN REVERSA
EN TIEMPO PARA LA EXPLORACIÓN PETROLERA**

TESIS
QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN INGENIERÍA

PRESENTA:
JUAN ARTURO JIMÉNEZ REQUENA

TUTOR PRINCIPAL
DR. JOSÉ CARLOS ORTIZ ALEMÁN
INSTITUTO MEXICANO DEL PETRÓLEO

MÉXICO, D. F. ABRIL 2018

JURADO ASIGNADO:

Presidente: Dr. Aguayo Camargo Joaquin Eduardo

Secretario: Dr. Cárdenas Soto Martín

Vocal: Dr. Ortiz Alemán José Carlos

1^{er}. Suplente: M. C. Escobedo Zenil David

2^{do}. Suplente: M. I. Castrejón Pineda Héctor Ricardo

Lugar o lugares donde se realizó la tesis: INSTITUTO MEXICANO DEL PETRÓLEO

TUTOR DE TESIS:

DR. JOSÉ CARLOS ORTIZ ALEMÁN



FIRMA

(Segunda hoja)

Índice general

Índice	i
1. Introducci[Pleaseinsertintopreamble]n	1
1.1. Hipótesis	3
1.2. Objetivos	4
1.3. Antecedentes	5
2. Marco teórico	6
2.1. Ecuación de onda	6
2.2. Migración	8
2.3. Migración RTM	10
2.3.1. Principio de la Migración Reversa en Tiempo	11
2.3.2. Aplicaciones y desafíos en la RTM	12
2.4. Kernels de sensibilidad	14
2.5. Modelo de velocidades	25
2.6. Problema Inverso	28
2.7. Discretización 3D	29
2.8. Condición de estabilidad	30

2.9. Diferencias finitas	32
2.10. Cómputo paralelo	35
2.10.1. Programación en CUDA C	36
2.10.2. Programación en Fortran MPI	39
3. Esquema numérico	40
3.1. Definición de parámetros	40
3.2. Kernels de sensibilidad	43
3.3. Fuentes (s) y Receptores (r)	45
3.4. Fronteras CPML	48
4. Resultados	55
4.1. Resultado de la propagación	55
4.2. Comparación de tiempos	60
5. Conclusión	63
Referencias	65

Capítulo 1

Introducción

Actualmente, en la industria petrolera la exploración geofísica es de gran relevancia por la creciente necesidad de encontrar más reservas de hidrocarburos que garanticen las necesidades energéticas de la sociedad en los años venideros. De entre los diversos métodos geofísicos que existen, la mayoría de las compañías petroleras utilizan la prospección sísmica como herramienta principal en sus actividades exploratorias. Esto a pesar de que dicha rama de la geofísica no determina directamente la ubicación de los reservorios de petróleo, sino las estructuras geológicas del subsuelo, a partir de las cuales se asocian ciertas estructuras con la presencia de hidrocarburos. Las razones principales por las que se prefiere utilizar este método son que posee una alta precisión, alta resolución, y sobre todo, una gran profundidad de investigación. Los domos salinos son estructuras geológicas que pueden servir como trampas para los hidrocarburos, ya que la sal es impermeable e insoluble en aceite y gas. Por tal motivo, este tipo de estructuras geológicas son de gran interés para la industria del petróleo. Las cuencas salinas, especialmente las cuencas terciarias con sal movilizada, son notoriamente difíciles de explorar debido a las tradicionales imágenes del subsuelo obtenidas con tecnologías aplicadas

hace muchos años atrás y que presentan una imagen muy pobre alrededor y bajo la sal. El problema que plantea determinar la forma y ubicación de los cuerpos salinos es que poseen una velocidad de onda P superior a la de los sedimentos que lo rodean, lo cual junto a su forma irregular provoca un patrón de dispersión de ondas sísmicas en múltiples direcciones. Por ello se necesita métodos que puedan reconstruir este patrón de dispersión y evitar la pérdida de información que crea zonas de sombra en las imágenes sísmicas. Si bien, la parte teórica ya había sido desarrollada, es decir, ya existían los algoritmos de migración, su ejecución no era algo sencillo de llevar a cabo dados los avances en términos de ciencias computacionales en aquellos entonces. En 90 ´s la migración Kirchhoff era la que se usaba más ampliamente aunque algunas implementaciones de los otros métodos también se usaban pero limitadamente. Hacia el final de esta década, la migración WEM finalmente se volvió comercialmente disponible y durante la mitad de los 2000s fue reconocida como la tecnología líder para imágenes subsalinas. Junto a la rápida implementación de la WEM, se dio paralelamente una rápida adopción en la industria de la migración RTM, que comenzó su apogeo en 2005. Para la implementación de estos métodos se volvió esencial determinar de forma más precisa el modelo de velocidades. En este proyecto se mejorará un programa previamente desarrollado para su aplicación a datos sintéticos en un modelo 2D usando distintos lenguajes de programación y comparando sus tiempos de operación.

El propósito principal de este programa, es utilizarlo como una poderosa herramienta numérica para la validación de modelos de velocidad construidos mediante la integración de datos geofísicos y para la construcción de imágenes sísmicas del subsuelo en profundidad.

1.1. Hipótesis

Una de las áreas de interés actual en la industria petrolera es la exploración en aguas profundas. En nuestro país es frecuente encontrar estructuras salinas que están asociadas a la presencia de yacimientos de hidrocarburos. El problema que plantea determinar la forma y ubicación de estos cuerpos salinos es que poseen una velocidad de onda P superior a la de los sedimentos que lo rodean, lo cual junto a su forma irregular provoca un patrón de dispersión de ondas sísmicas en múltiples direcciones. Por ello se necesita métodos que puedan reconstruir este patrón de dispersión y evitar la pérdida de información que crea zonas de sombra en las imágenes sísmicas.

Al obtener un modelo con mínimo error y sin mínimos locales se podrá usar en un proyecto de iluminación sísmica en conjunto a un algoritmo de migración avanzado.

1.2. Objetivos

Implementar el algoritmo de migración reversa en tiempo con capas perfectamente ajustadas al lenguaje de programación CUDA C y Fortran MPI.

Comparar los tiempos de ejecución del algoritmo de migración reversa en tiempo mediante el uso de la CPU y la implementación de la GPU.

1.3. Antecedentes

Se estima que en los últimos 150 años se han consumido alrededor de 1 billón de barriles (de equivalente) de petróleo, debido al incremento de la demanda, durante los próximos veinte años será necesario encontrar y extraer la misma cantidad de petróleo que en los 150 años precedentes. Los hidrocarburos en forma de aceite y gas de yacimientos someros, con poca dificultad de extracción, están por agotarse y para ubicar nuevas reservas se tendrá que explorar en áreas que abarcan una geología compleja y que en el pasado fueron descartadas por el alto costo de extracción. Para tal fin, la industria petrolera deberá realizar imágenes del subsuelo con una mayor resolución a las realizadas anteriormente. Por lo que se requiere implementar a los algoritmos ya realizados para la industria petrolera el método de cómputo paralelo para generar imágenes de mayor calidad a menor tiempo de ejecución. El algoritmo de migración RTM se optimizará con iteraciones bajo el cómputo paralelo, mediante el lenguaje de programación CUDA C y Fortran MPI.

Capítulo 2

Marco teórico

En este capítulo se describirá con la mayor claridad posible la teoría detrás del método a utilizarse.

2.1. Ecuación de onda

Empezamos a partir de la segunda ley de Newton

$$F = ma = \rho \frac{\partial^2 \mu}{\partial t^2} \quad (2.1)$$

Las ondas se propagan como perturbaciones mecánicas en el subsuelo por lo que el esfuerzo resultante es diferente de cero para que puedan existir deformaciones. Debido a esto se recurre a la ley de Hooke, que relaciona el esfuerzo en términos de la deformación. Siendo el medio bidimensional y acústico, se hace un análisis de fuerzas resultantes en dirección del eje x, la superficie del suelo, y en dirección z, la profundidad, recordando que no existe esfuerzos cortantes en medios acústicos ($\mu = 0$), obteniendo las siguientes ecuaciones:

$$\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{zx}}{\partial z} = \rho \frac{\partial^2 \mu_x}{\partial t^2} \quad (2.2)$$

$$\frac{\partial \sigma_{xz}}{\partial x} + \frac{\partial \sigma_{zz}}{\partial z} = \rho \frac{\partial^2 \mu_z}{\partial t^2} \quad (2.3)$$

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{zz} \\ \sigma_{zx} \end{bmatrix} = \begin{bmatrix} \lambda & \lambda & 0 \\ \lambda & \lambda & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial \mu_x}{\partial x} \\ \frac{\partial \mu_z}{\partial z} \\ \frac{1}{2} \left(\frac{\partial \mu_x}{\partial z} + \frac{\partial \mu_z}{\partial x} \right) \end{bmatrix} \quad (2.4)$$

Se reescribe las dos ecuaciones de los desplazamientos (μ_x, μ_z) considerando las componentes de la velocidad de desplazamientos (v_x, v_z) y las tres ecuaciones de esfuerzos se derivan respecto al tiempo, obteniendo así las cinco ecuaciones utilizadas para la propagación de ondas en un medio bidimensional acústico.

$$\frac{\partial v_x}{\partial t} = \frac{1}{\rho} \frac{\partial \sigma_{xx}}{\partial x} \quad (2.5)$$

$$\frac{\partial v_z}{\partial t} = \frac{1}{\rho} \frac{\partial \sigma_{zz}}{\partial z} \quad (2.6)$$

$$\frac{\partial \sigma_{xx}}{\partial t} = \lambda \left(\frac{\partial v_x}{\partial x} + \frac{\partial v_z}{\partial z} \right) \quad (2.7)$$

$$\frac{\partial \sigma_{zz}}{\partial t} = \lambda \left(\frac{\partial v_x}{\partial x} + \frac{\partial v_z}{\partial z} \right) \quad (2.8)$$

$$\sigma_{xz} = \sigma_{zx} = 0 \quad (2.9)$$

2.2. Migración

Para entender mejor el algoritmo de inversión propuesto y por qué se eligió, primero se debe conocer qué lugar ocupa junto con otros métodos para obtener imágenes sísmicas. Se le llama generación de imágenes, al proceso mediante el cual las reflexiones sísmicas se despliegan en su posición correcta. Este proceso consiste de dos elementos principales: el apilamiento y la migración. La migración utiliza un modelo de velocidad para redistribuir la energía sísmica reflejada, desde la posición supuesta en el punto medio, a su verdadera posición.

Algunos problemas que se presentan en la generación de imágenes se pueden resolver con migración en el dominio del tiempo, pero aquellos que son más complejos requieren de una migración en el dominio de la profundidad. La migración se puede realizar en dos dominios: profundidad y tiempo y puede realizarse antes o después del apilamiento. En la migración en profundidad, el modelo de velocidad puede tener fuertes contrastes en las direcciones horizontal o vertical. Por ello, se elige este tipo de migración cuando hay pliegues, intrusiones o fallas de gran inclinación que se yuxtaponen con capas cuyas propiedades elásticas son muy diferentes. La migración en profundidad es una operación que requiere un considerable tiempo de ejecución y necesita un modelo de velocidad preciso en escala de profundidad. Podemos decir que al referirnos a las imágenes en profundidad estamos hablando de migración en profundidad PSDM (Pre-Stack Depth Migration) y sus elementos son los datos de entrada, preprocesado, algoritmo de migración y el modelo de velocidades. De acuerdo a la complejidad estructural del objetivo se elige el tipo de migración a utilizar. Con algoritmos sencillos de migración se pueden obtener imágenes de estructuras simples y velocidades que varían levemente, pero, en aquellos casos donde las estructuras son más complejas y las velocidades varían abruptamente, es posible que las rutinas más sencillas de procesamiento no funcionen. Una característica de la migración en profundidad, es que esta asume que el modelo de velocidades es conocido y calculará la forma

Algoritmo	Fidelidad en iluminación	Echados fuertes	Multitrayectoria	Rapidez de ejecución	Flexibilidad de la salida	Limitaciones
Kirchhoff	no	Si	No	Si	Si	Limitado por teoría de rayos
Beam	no	Si	Restringido	Si	si	Limitado por teoría de rayos
WEM	Restringido	No	si	no		
RTM	si	si	si	no	si	

Figura 2.1: Tabla de algoritmos de migración. (García Reyes, 2013)

correcta de la difracción para ese modelo de velocidad. El dato se migra de acuerdo a la forma de la difracción y la salida es un dato con eje vertical en profundidad. Si el modelo de velocidad es incorrecto, la migración será incorrecta y el error puede ser difícil de detectar en una migración post-stack. Realizar la migración pre-apilado, permitirá una estimación del error del resultado migrado. La migración en profundidad toma alrededor de diez veces más en ejecutarse que la migración en tiempo y es muy sensible a errores en la velocidad además de que puede requerir muchas iteraciones, con el consiguiente incremento en su tiempo de ejecución. Este trabajo está enfocado en la migración en profundidad en áreas donde la tectónica salina está presente y por lo tanto se debe realizar una migración PSDM preapilamiento. En la industria se realizan básicamente cuatro algoritmos principales de migración en profundidad pre-apilamiento:

- a) Migración Kirchhoff en profundidad
- b) Extrapolación de campo de ondas unidireccional (WEM)
- c) Migración Beam
- d) Extrapolación de campo de ondas bidireccional (RTM)

Para este trabajo se usara la inversión de forma de onda para brindar un modelo que pueda ser usado para un algoritmo de migración RTM, esto pensando en que posee las menores limitaciones y la mejor iluminación comparado con otros algoritmos de migración.

2.3. Migración RTM

La migración en tiempo reverso (RTM) es una técnica de migración antes de apilar que a diferencia del resto de las migraciones, deja de lado las simplificaciones y utiliza la ecuación de onda completa.

La migración reversa en tiempo fue introducida inicialmente a principios de los ochenta por Baysal and Sherwood [1983], pero no fue hasta la actualidad, en que se han revolucionado los dispositivos computacionales que se desarrolló y catapultó al escenario como una herramienta poderosa para generar imágenes sísmicas en profundidad.

La migración RTM es hoy en día un método estándar para afrontar los retos debidos a altos contrastes de velocidad y/o fuertes echados en los flancos de estructuras, como es el caso de la geología compleja en el Golfo de México. Como datos de entrada, esta técnica requiere un modelo de velocidades propuesto del medio, y como resultado, genera una imagen representativa de las estructuras presentes en el subsuelo. Se puede resumir en tres pasos principales.

1. El primer paso es la propagación directa de un frente de onda desde la fuente hasta los receptores.
2. El segundo paso es la propagación del campo de onda grabado en dirección reversa hasta la posición de la fuente.
3. El tercer paso consiste en que cada intervalo de tiempo donde el campo de onda directo y el retro-propagado se traslapan son las posiciones del reflector, por lo cual el valor de retraso cero de la correlación cruzada de los dos campos de onda nos proporciona la posición de los reflectores.

Por lo tanto, la condición de la imagen para la RTM se puede expresar como:

$$I(x) = \int_0^T S(x)R(x, T - t)dt, \quad (2.10)$$

En donde $S(x)$ el campo de onda de propagación directa y $R(x, T - t)$ es el campo de onda retro-propagado desde el receptor [Baysal and Sherwood, 1983]. De esta expresión se puede ver que la relación entre los campos de ondas de las propagaciones originales y de receptores es tal, que se inicia multiplicando el campo de ondas final de la propagación original con el campo inicial de la propagación por receptores y se termina con la multiplicación del campo inicial de la propagación original con el campo final de la propagación por receptores. Además de llevar a cabo las multiplicaciones, los resultados se van sumando y así, se genera la imagen correspondiente a cada una de las fuentes. El paso final consiste en sumar los resultados obtenidos para todas las fuentes. Esta imagen es el resultado final de la RTM para el modelo de velocidades introducido.

2.3.1. Principio de la Migración Reversa en Tiempo

Las investigaciones en la RTM se basaron en dos modelos principales con lo que el primero es el modelo de reflectores detonantes en donde los métodos convencionales de datos apilados migrados caen en las suposiciones de que la sección de entrada en tiempo se ajusta al modelo reflector detonante. Este modelo asume que toda la energía presente en el tiempo t de datos apilados, surge de la reflexión en el tiempo $0.5t$ y así la intensidad del coeficiente de reflexión puede ser determinada por la propagación de la energía de regreso a la Tierra a la mitad de su tiempo de llegada.

El segundo modelo fue el más aceptado, debido a la rápida comprensión por la comunidad científica, el realizado por Baysal and Sherwood [1983] los cuales usaron el modelo de reflector detonante y denotan sus soluciones para la ecuación de onda escalar:

$$P_{xx} + P_{zz} = \frac{1}{c^2(x, z)} P_{tt}, \quad (2.11)$$

Generalmente usada para propagar esos datos que pueden manipularse en modelos directos o inversos en tiempo por simplemente invirtiendo el eje del tiempo.

Así es, si $P(x, z, t)$ es una solución a la (2.11), entonces también es $P(x, z, t_0 - t)$ para cualquier constante fija t_0 [Levin, 1984].

2.3.2. Aplicaciones y desafíos en la RTM

Con las investigaciones sobre el funcionamiento de la migración RTM, salió a flote que aunque las reflexiones primarias proporcionan información sísmica, los múltiples superficiales y los múltiples internos también proporcionan información valiosa. Los múltiples son ondas reflejadas más de una vez entre y en las interfaces antes de que lleguen a los receptores. Normalmente viajan distancias más grandes que las reflexiones primarias cubriendo así mayor área de estudio. Pueden proporcionar iluminación adicional en zonas de sombra de reflexiones primarias. Además, los ángulos de reflexión relativamente son pequeños y más adecuados para la obtención de imágenes de reflectores no horizontales y con pendientes [Berkhout and Verschuur, 2003]. Incluso con estas ventajas extraer eventos de los múltiples no es tan sencillo Weglein [2014]. La idea de extraer eventos valiosos de múltiples se remonta a la obra de Claerbout [1968] sobre interferometría sísmica, lo que demuestra que la función de Green en la superficie de la Tierra puede obtenerse mediante la autocorrelación de trazas generadas por fuentes enterradas. De esta manera, los múltiple se transforman cinemáticamente en reflexiones primarias con fuentes virtuales en la superficie, seguido por la migración utilizando métodos convencionales de migración.

Por lo que a pesar de los esfuerzos por obtener la mayor información posible, incluso con la extracción de eventos de los múltiples, la necesidad de adaptar un modelo de velocidades a la

migración reversa en tiempo se ve detenida por tiempos de ejecución de cómputo y la obtención de los kernels de sensibilidad por medio de la inversión de onda completa. Por ello, se define en la siguiente sección la importancia del modelo de velocidades y más adelante la aplicación de la inversión de onda completa.

2.4. Kernels de sensibilidad

Para poder realizar la migración RTM se debe considerar los núcleos(kernels) de sensibilidad que se ingresan al modelo de velocidades. De acuerdo con Tarantola [1984], para el caso acústico (sísmico) se puede realizar con un proceso iterativo en el que se pretende disminuir el valor de la función de error entre los datos observados y los datos sintéticos mediante la perturbación de cada uno de los parámetros del modelo, para realizarlo eficazmente se debe emplear la diferencial de Fréchet de la función de error en la dirección de la perturbación que son sometidos los parámetros, donde dicha diferencial representa la sensibilidad del error respecto a cada uno de los parámetros del medio.

Según Tromp et al. [2005a], la diferencial de Fréchet de una función de error entre sismogramas observados y sintéticos generados con un conjunto de parámetros \bar{m} , debido a la perturbación $\delta\bar{m}$, tiene la forma general

$$D_{\delta\bar{m}}\chi(\bar{m}) = \int_V \sum_{i=1}^P K_{m_i}(\bar{x}) \frac{\delta m_i(\bar{x})}{m_i(\bar{x})} dV(\bar{x}), \quad (2.12)$$

en donde P es el número de tipos de parámetros del modelo, K_{m_i} es el kernel asociado al parámetro m_i y $\delta m_i(\bar{x})$ es la perturbación que sufre el parámetro m_i por lo que $\frac{\delta m_i(\bar{x})}{m_i(\bar{x})}$ representa la perturbación relativa de cada parámetro. Existe un kernel asociado a cada parámetro del medio y su valor depende de la posición.

Ahora bien, para poder definir matemáticamente los kernels que caracterizan a las diferenciales, es necesario calcular la diferencial de Fréchet de la función de error en la dirección del vector de perturbaciones de los parámetros. En otras palabras, calcular la variación de la función de error $X(\bar{m})$ debida a la perturbación $\delta\bar{m}$ de los parámetros \bar{m} . Siguiendo el desarrollo de Tromp et al. [2005a], se utiliza la función de error entre datos observados y datos sintéticos dada por

$$\chi(\bar{m}) = \frac{1}{2} \sum_{i=1}^N \int_0^T \sum_{i=1}^M [s_i(\bar{x}_r, t, \bar{m}) - d_i(\bar{x}_r, t)]^2 dt, \quad (2.13)$$

en donde N es el número de receptores, T el tiempo de propagación, M el número de componentes que registra cada receptor, \bar{x}_r el vector con la ubicación del receptor r , $s_i(\bar{x}, t, \bar{m})$ la componente i de los desplazamientos sintéticos generados en la posición \bar{x} en el tiempo t para el modelo que tiene como parámetros al vector \bar{m} y, por último $d_i(\bar{x}_r, t)$ es la componente i de los datos observados en el receptor r en el tiempo t . La diferencial de Fréchet queda expresada por

$$D_{\delta\bar{m}}\chi(\bar{m}) = \sum_{i=1}^N \int_0^T \sum_{i=1}^M [s_i(\bar{x}_r, t, \bar{m}) - d_i(\bar{x}_r, t)] D_{\delta\bar{m}}s_i(\bar{x}_r, t, \bar{m}) dt, \quad (2.14)$$

en donde $D_{\delta\bar{m}}s_i(\bar{x}_r, t, \bar{m})$ es otra diferencial de Fréchet, que representa la variación en los desplazamientos debido a la perturbación $\delta\bar{m}$ que sufrieron los parámetros \bar{m} . Hacer el cálculo directo de esta expresión es impráctico, pues implica hacer una propagación inicial con los parámetros originales \bar{m} y una para cada uno de los parámetros del modelo con su perturbación correspondiente, lo cual no es conveniente si tomamos en cuenta el gran número de parámetros a invertir y el tiempo requerido para simular el problema directo.

Para expresar la diferencial de Fréchet que quedó dentro de la integral de una forma tal que su cálculo sea menos costoso en términos computacionales, es necesario recurrir a la teoría de la dispersión de ondas elásticas, proceso que se refiere a la desviación de ondas respecto a la dirección inicial de su movimiento provocada por la interacción con cierta heterogeneidad del medio. Según esta teoría, es posible determinar el campo de ondas dispersado por una heterogeneidad que tiene densidades y constantes elásticas ligeramente diferentes del medio que los rodea usando el método de la fuente equivalente y la aproximación de Born Wu and Aki [1985]. La aproximación de Born es una herramienta muy utilizada para estudiar la dispersión de partículas subatómicas en la mecánica cuántica. De acuerdo con Tromp et al. [2005a], si se

tiene una heterogeneidad con variaciones $\delta\rho(\bar{x})$ y $\delta c_{jklm}(\bar{x})$ de densidades y constantes elásticas, respectivamente, de las del resto del medio, el campo dispersado queda determinado por

$$D_{\delta\bar{m}}s_i(\bar{x}_r, t, \bar{m}) = - \int_0^t \int_V \left[\delta\rho(\bar{x}') \sum_{j=1}^M G_{ij}(\bar{x}, \bar{x}'; t - t') \frac{\partial^2}{\partial t'^2} s_j(\bar{x}', t') \right. \\ \left. + \sum_{j=1}^M \sum_{k=1}^M \sum_{l=1}^M \sum_{m=1}^M \delta c_{jklm}(\bar{x}') \frac{\partial}{\partial x'_k} G_{ij}(\bar{x}, \bar{x}'; t - t') \frac{\partial}{\partial x'_l} s_m(\bar{x}_r, t) \right] dV(\bar{x}') dt', \quad (2.15)$$

en donde V es todo el volumen del medio, pero se reduce al volumen del cuerpo heterogéneo porque en el resto del espacio las perturbaciones son nulas, M es el número de componentes del campo de desplazamientos y G_{ij} es la función de Green de la elastodinámica. $G_{ij}(\bar{x}, \bar{x}'; t - t')$ representa el desplazamiento en la dirección i en la posición \bar{x} y en el tiempo t , originado por una fuente impulsiva $f_i(\bar{x}, t)$ que se activó en el tiempo t' [Aki and Richards, 2002], definida como

$$f_i(\bar{x}, t) = C\delta(\bar{x} - \bar{x}_f)\delta(t - t')\delta_{ij}, \quad (2.16)$$

Donde C es la amplitud de la fuente y δ_{ij} es la delta de Kronecker, lo cual indica que se trata de una fuerza aplicada en la dirección j . Esto significa que la función de Green de la elastodinámica es un tensor que representa el campo de desplazamientos en cualquier dirección, por lo que satisface la ecuación de onda.

Introduciendo la ecuación (2.4) en la ecuación (2.16), la diferencial de Fréchet debida a la perturbación de parámetros $\delta\bar{m}$ se puede expresar como

$$D_{\delta\bar{m}}\chi(\bar{m}) = - \sum_{i=1}^N \int_0^T \sum_{i=1}^M [s_i(\bar{x}_r, t, \bar{m}) - d_i(\bar{x}_r, t)] \int_0^t \int_V \left[\delta\rho(\bar{x}') \sum_{j=1}^M G_{ij}(\bar{x}_r, \bar{x}'; t - t') \frac{\partial^2}{\partial t'^2} s_j(\bar{x}', t') \right. \\ \left. + \sum_{j=1}^M \sum_{k=1}^M \sum_{l=1}^M \sum_{m=1}^M \delta c_{jklm}(\bar{x}') \frac{\partial}{\partial x'_k} G_{ij}(\bar{x}, \bar{x}'; t - t') \frac{\partial}{\partial x'_l} s_m(\bar{x}_r, t) \right] dV(\bar{x}') dt' dt, \quad (2.17)$$

Ecuación que se puede reescribir en términos de los parámetros perturbados,

$$D_{\delta\bar{m}}\chi(\bar{m}) = D_{\delta\rho}\chi(\bar{m}) + D_{\delta c_{jklm}}\chi(\bar{m}), \quad (2.18)$$

en donde

$$D_{\delta\rho}\chi(\bar{m}) = - \sum_{i=1}^N \int_0^T \sum_{i=1}^M [s_i(\bar{x}_r, t, \bar{m}) - d_i(\bar{x}_r, t)] \int_0^t \int_V \delta\rho(x') \sum_{j=1}^M G_{ij}(\bar{x}_r, \bar{x}'; t - t') \frac{\partial^2}{\partial t'^2} s_j(\bar{x}', t') dV(\bar{x}') dt' dt \quad (2.19)$$

$$D_{\delta c_{jklm}}\chi(\bar{m}) = - \sum_{i=1}^N \int_0^T \sum_{i=1}^M [s_i(\bar{x}_r, t, \bar{m}) - d_i(\bar{x}_r, t)] \int_0^t \int_V \sum_{j=1}^M \sum_{k=1}^M \sum_{l=1}^M \sum_{m=1}^M \delta c_{jklm}(\bar{x}') \frac{\partial}{\partial x'_k} G_{ij}(\bar{x}, \bar{x}'; t - t') \frac{\partial}{\partial x'_l} s_m(\bar{x}_r, t) dV(\bar{x}') dt' dt \quad (2.20)$$

Además si se hace lo mismo con la ecuación (2.12),

$$D_{\delta\bar{m}}\chi(\bar{m}) = \int_V \left[K_\rho(\bar{x}) \frac{\delta\rho(\bar{x})}{\rho(\bar{x})} + \sum_{j=1}^M \sum_{k=1}^M \sum_{l=1}^M \sum_{m=1}^M K_{c_{jklm}}(\bar{x}) \frac{\delta c_{jklm}(\bar{x})}{c_{jklm}(\bar{x})} \right] dV(\bar{x}) \quad (2.21)$$

se tiene que

$$\int_V K_\rho(\bar{x}) \frac{\delta\rho(\bar{x})}{\rho(\bar{x})} dV(\bar{x}) = - \sum_{i=1}^N \int_0^T \sum_{i=1}^M [s_i(\bar{x}_r, t, \bar{m}) - d_i(\bar{x}_r, t)] \int_0^t \int_V \delta\rho(x') \sum_{j=1}^M G_{ij}(\bar{x}_r, \bar{x}'; t - t') \frac{\partial^2}{\partial t'^2} s_j(\bar{x}', t') dV(\bar{x}') dt' dt \quad (2.22)$$

y

$$\int_V K_{c_{jklm}}(\bar{x}) \frac{\delta c_{jklm}(\bar{x})}{c_{jklm}(\bar{x})} dV(\bar{x}) = - \sum_{i=1}^N \int_0^T \sum_{i=1}^M [s_i(\bar{x}_r, t, \bar{m}) - d_i(\bar{x}_r, t)] \int_0^t \int_V \sum_{j=1}^M \sum_{k=1}^M \sum_{l=1}^M \sum_{m=1}^M \delta c_{jklm}(\bar{x}') \frac{\partial}{\partial x'_k} G_{ij}(\bar{x}, \bar{x}'; t - t') \frac{\partial}{\partial x'_l} s_m(\bar{x}_r, t) dV(\bar{x}') dt' dt$$

$$\frac{\partial}{\partial x'_k} G_{ij}(\bar{x}, \bar{x}'; t - t') \frac{\partial}{\partial x'_l} s_m(\bar{x}_r, t) dV(\bar{x}') dt' dt \quad (2.23)$$

Cambiando el orden de integración y de las sumas en la ecuación (2.22), se obtiene

$$\int_V K_\rho(\bar{x}) \frac{\delta\rho(\bar{x})}{\rho(\bar{x})} dV(\bar{x}) = - \int_V \int_0^T \int_0^t \sum_{j=1}^M \sum_{r=1}^N \sum_{i=1}^M \delta\rho(x') G_{ij}(\bar{x}_r, \bar{x}'; t - t') [s_i(\bar{x}_r, t, \bar{m}) - d_i(\bar{x}_r, t)] \frac{\partial^2}{\partial t'^2} s_j(\bar{x}', t') dt' dt dV(\bar{x}') \quad (2.24)$$

$$K_\rho(\bar{x}) \frac{\delta\rho(\bar{x})}{\rho(\bar{x})} = - \int_0^T \int_{t'}^T \sum_{j=1}^M \sum_{r=1}^N \sum_{i=1}^M \delta\rho(x') G_{ij}(\bar{x}_r, \bar{x}'; t - t') [s_i(\bar{x}_r, t, \bar{m}) - d_i(\bar{x}_r, t)] \frac{\partial^2}{\partial t'^2} s_j(\bar{x}', t') dt' dt \quad (2.25)$$

$$K_\rho(\bar{x}) \frac{\delta\rho(\bar{x})}{\rho(\bar{x})} = - \int_0^T \sum_{j=1}^M \delta\rho(x') \left\{ \sum_{r=1}^N \int_{t'}^T \sum_{i=1}^M G_{ij}(\bar{x}_r, \bar{x}'; t - t') [s_i(\bar{x}_r, t, \bar{m}) - d_i(\bar{x}_r, t)] dt \right\} \frac{\partial^2}{\partial t'^2} s_j(\bar{x}', t') dt' \quad (2.26)$$

Si además, se utiliza la propiedad de reciprocidad que tiene la función de Green, según la cual

$$G_{ij}(\bar{x}_r, \bar{x}'; t - t') = G_{ji}(\bar{x}', \bar{x}_r; t - t'), \quad (2.27)$$

la ecuación (2.26) se puede escribir como

$$K_\rho(\bar{x}) \frac{\delta\rho(\bar{x})}{\rho(\bar{x})} = - \int_0^T \sum_{j=1}^M \delta\rho(x') \left\{ \sum_{r=1}^N \int_{t'}^T \sum_{i=1}^M G_{ji}(\bar{x}', \bar{x}_r; t - t') [s_i(\bar{x}_r, t, \bar{m}) - d_i(\bar{x}_r, t)] dt \right\}$$

$$\frac{\partial^2}{\partial t'^2} s_j(\bar{x}', t') dt' \quad (2.28)$$

Y haciendo el cambio de variables

$$t'' = T - t, \quad (2.29)$$

$$t = T - t'', \quad (2.30)$$

la ecuación (2.28) se transforma en

$$K_\rho(\bar{x}) \frac{\delta\rho(\bar{x})}{\rho(\bar{x})} = - \int_0^T \sum_{j=1}^M \delta\rho(x') \left\{ \sum_{r=1}^N \int_0^{T-t'} \sum_{i=1}^M G_{ji}(\bar{x}', \bar{x}_r; T - t'' - t') \left[s_i(\bar{x}_r, T - t'', \bar{m}) - d_i(\bar{x}_r, T - t'') \right] dt'' \right\} \frac{\partial^2}{\partial t'^2} s_j(\bar{x}', t') dt' \quad (2.31)$$

Si introducimos una integral de volumen y una función impulso espacial $\delta(\bar{x}' - \bar{x}_r)$, y cambiamos la variable muda t'' por t , la ecuación anterior se transforma en

$$K_\rho(\bar{x}) \frac{\delta\rho(\bar{x})}{\rho(\bar{x})} = - \int_0^T \sum_{j=1}^M \delta\rho(x') \left\{ \sum_{r=1}^N \int_0^{T-t'} \int_V \sum_{i=1}^M G_{ji}(\bar{x}', \bar{x}_r; T - t - t') \left[s_i(\bar{x}_r, T - t, \bar{m}) - d_i(\bar{x}_r, T - t) \right] \delta(\bar{x}' - \bar{x}_r) dV(\bar{x}) dt \right\} \frac{\partial^2}{\partial t'^2} s_j(\bar{x}', t') dt' \quad (2.32)$$

Finalmente, agrupando términos y multiplicando por $\frac{\rho(\bar{x})}{\delta\rho(\bar{x})}$, se tiene la expresión para el kernel $K_\rho(\bar{x})$,

$$K_\rho(\bar{x}) = - \int_0^T \sum_{j=1}^M \rho(x') \hat{s}_j(\bar{x}', T - t') \frac{\partial^2}{\partial t'^2} s_j(\bar{x}', t') dt' \quad (2.33)$$

en donde

$$\hat{s}_j(\bar{x}', T - t') = \int_0^{T-t'} \int_V \sum_{i=1}^M G_{ji}(\bar{x}', \bar{x}_r; T - t - t') \hat{f}_i(\bar{x}, t) dV(\bar{x}) dt \quad (2.34)$$

y

$$\hat{f}_i(\bar{x}, t) = \sum_{r=1}^N [s_i(\bar{x}_r, T - t, \bar{m}) - d_i(\bar{x}_r, T - t)] \delta(\bar{x}' - \bar{x}_r) \quad (2.35)$$

Comparando las ecuaciones (2.35) y (2.16) se puede ver que $\hat{f}_i(\bar{x}, t)$ tiene la forma de una fuente impulsiva y $\hat{s}_j(\bar{x}', T - t')$, de acuerdo a la definición de función de Green de arriba, corresponde al campo de desplazamientos originado por dicha fuente, es decir, $\hat{s}_j(\bar{x}', t')$ es el campo de desplazamientos en la dirección de j generado por la diferencia entre los datos observados \bar{s} y los sintéticos \bar{d} . A esta propagación se le conoce en la literatura como propagación adjunta y a los campos involucrados en ella campos adjuntos. La función de Green de la elastodinámica dentro de la integral indica que se trata del problema directo de propagación de ondas, y el argumento $(\bar{x}', T - t')$ de \hat{s}_j indica que la relación entre el campo original y el adjunto es tal que el último elemento del campo adjunto, se relaciona con el primero del campo original, el penúltimo con el segundo, y así sucesivamente. Esto quiere decir que si la discretización en tiempo se hace en N pasos, al construir los kernels se relacionan los valores de la propagación adjunta \hat{s}_j del paso 1 con los valores de $\frac{\partial^2}{\partial t'^2} \bar{s}$ del paso N de la propagación original, los valores del paso 2 de la propagación adjunta con los del paso $N - 1$ de la propagación original, y así sucesivamente. De lo que se concluye que a pesar de que se invirtió el orden de integración en (2.31), no se trata de una retropropagación, sino que, como lo indica la ecuación (2.34) se resuelve el mismo problema directo, y este cambio de orden únicamente hace referencia a la relación que tiene esta propagación con la original. Haciendo un desarrollo similar se puede probar que

$$K_{c_{jklm}}(\bar{x}') = - \int_0^T \hat{\epsilon}_{jk}(\bar{x}', T - t') c_{jklm}(\bar{x}') \hat{\epsilon}_{lm}(\bar{x}', T - t') dt, \quad (2.36)$$

en donde $\hat{\epsilon}_{jk}(\bar{x}', T - t')$ es el elemento jk del tensor de deformaciones de la propagación adjunta correspondiente a $\hat{s}_j(\bar{x}', T - t')$, y $\hat{\epsilon}_{lm}(\bar{x}', T - t')$ el elemento lm del tensor de deformaciones de la propagación original correspondiente a $\bar{s}(\bar{x}, t)$. No obstante de que con estos

dos kernels se puede caracterizar la sensibilidad de la función de error respecto a cada uno de los parámetros de cualquier medio elástico, en muchos casos se hace la consideración de que el medio objeto de estudio es isótropo, el cual se caracteriza con solamente dos de las constantes elásticas del tensor c_{jklm} , por lo que es necesario reducir este último kernel. Haciendo las reducciones correspondientes, estos medios se pueden caracterizar por un total de tres kernels, cada uno correspondiente a parámetros independientes. Tomando en cuenta que para medios isótropos se cumple que

$$c_{jklm} = \left(\kappa \frac{2}{3} \mu \right) \delta_{jk} \delta_{lm} + \mu (\delta_{jl} + \delta_{km} + \delta_{jm} \delta_{kl}) \quad (2.37)$$

se puede expresar la diferencial de Fréchet en términos de ρ , μ y κ ,

$$D_{\delta \bar{m}} \chi(\bar{m}) = \int_V \left[K_\rho(\bar{x}) \frac{\delta \rho(\bar{x})}{\rho(\bar{x})} + K_\mu(\bar{x}) \frac{\delta \mu(\bar{x})}{\mu(\bar{x})} + K_\kappa(\bar{x}) \frac{\delta \kappa(\bar{x})}{\kappa(\bar{x})} \right] dV(\bar{x}) \quad (2.38)$$

en donde

$$K_\mu(\bar{x}) = - \int_0^T 2\mu(\bar{x}) \hat{D}(\bar{x}, T-t) : D(\bar{x}, t) dt, \quad (2.39)$$

$$K_\kappa(\bar{x}) = - \int_0^T \kappa(\bar{x}) [\nabla \cdot \hat{s}(\bar{x}, T-t)] [\nabla \cdot \bar{s}(\bar{x}, T-t)] dt, \quad (2.40)$$

en las que $D(\bar{x}, t)$ es el tensor de esfuerzos desviador de la propagación original y $\hat{D}(\bar{x}, T-t)$ el correspondiente a la propagación adjunta. Además de estas formas de obtener la diferencial de Fréchet, también es posible obtenerla en términos de kernels que dependen de las velocidad de ondas P y S,

$$D_{\delta \bar{m}} \chi(\bar{m}) = \int_V \left[K_\rho(\bar{x}) \frac{\delta \rho(\bar{x})}{\rho(\bar{x})} + K_\alpha(\bar{x}) \frac{\delta \alpha(\bar{x})}{\alpha(\bar{x})} + K_\beta(\bar{x}) \frac{\delta \beta(\bar{x})}{\beta(\bar{x})} \right] dV(\bar{x}) \quad (2.41)$$

en donde

$$K_X(\bar{x}) = K_\rho(\bar{x}) + K_\mu(\bar{x}) + K_\kappa(\bar{x}) \quad (2.42)$$

$$K_\alpha(\bar{x}) = 2 \left(1 + \frac{4\mu}{3\kappa} \right) K_\kappa(\bar{x}) \quad (2.43)$$

y

$$K_\beta(\bar{x}) = 2 \left[K_\mu(\bar{x}) - \frac{4\mu}{3\kappa} K_\kappa(\bar{x}) \right] \quad (2.44)$$

De este modo se cuenta con varios kernels que representan la sensibilidad entre las observaciones y los parámetros estructurales del subsuelo y constituyen la base de la inversión de forma de onda completa. Si se trata de un medio isótropo basta con conocer los kernels correspondientes a tres parámetros del subsuelo independientes a partir de los cuales se puede calcular cualquier combinación, sin embargo este trabajo tiene como objetivo estudiar únicamente los medios acústicos(sísmicos), los cuales se caracterizan con una sola de las constantes elásticas del tensor c_{jklm} . Por lo tanto, basta con obtener dos kernels para calcular sus diferenciales de Fréchet. Esto se aprecia claramente si consideramos que en estas condiciones $\mu = 0$ y por lo tanto $K_\mu = 0$ por lo que basta con calcular K_ρ y $K_\kappa = 0$ para conocer cualquier combinación de kernels deseada.

Considerando lo anterior, la construcción de estos kernels involucra únicamente dos simulaciones de la propagación de ondas. La primera es la propagación original, en la que las fuentes actúan como fuentes y los receptores como receptores, mientras que la segunda, conocida como propagación adjunta, utiliza a los receptores originales como fuentes, y como funciones de fuente a las diferencias entre los sismogramas sintéticos y los sismogramas observados. Con estas dos simulaciones se generan los kernels correspondientes a los parámetros del medio seleccionados para caracterizar la sensibilidad de la función de error. Este planteamiento del problema permite cambiar la integral en tiempo de la ecuación

Por otra parte, en la tomografía de tiempos de arribo, la función de error está dada por

$$\chi(\bar{m}) = \frac{1}{2} \sum_{r=1}^N [T_r(\bar{m}) - T_r^{obs}]^2, \quad (2.45)$$

En donde $T_r(\bar{m})$ es el tiempo de arribo en la estación r calculado con un conjunto de parámetros (\bar{m}) y T_r^{obs} es el tiempo observado para la misma estación [Tromp et al., 2005b]. La

diferencial de Fréchet de esta función de error en la dirección de la perturbación de parámetros $\delta\bar{m}$ está dada por

$$D_{\delta\bar{m}}\chi(\bar{m}) = \sum_{r=1}^N [T_r(\bar{m}) - T_r^{obs}] D_{\delta\bar{m}} T_r(\bar{m}) \quad (2.46)$$

en donde $D_{\delta\bar{m}} T_r(\bar{m})$ es la diferencial de Fréchet de los tiempos de arribo en dirección de la perturbación de los parámetros $\delta\bar{m}$. De acuerdo con el trabajo de [Tromp et al., 2005b] esta última diferencial se puede expresar, al igual que la diferencial de Fréchet de la función de error de la inversión de forma de onda completa, en términos de kernels de sensibilidad,

$$D_{\delta\bar{m}} T_r(\bar{m}) = \int_V \left[K_{\rho}^*(\bar{x}, \bar{x}_r) \frac{\delta\rho(\bar{x})}{\rho(\bar{x})} + K_{\mu}^*(\bar{x}, \bar{x}_r) \frac{\delta\mu(\bar{x})}{\mu(\bar{x})} + K_{\kappa}^*(\bar{x}, \bar{x}_r) \frac{\delta\kappa(\bar{x})}{\kappa(\bar{x})} \right] dV(\bar{x}) \quad (2.47)$$

en donde

$$K_{\rho}^*(\bar{x}, \bar{x}_r) = - \int_0^T \sum_{j=1}^M \rho(\bar{x}) \hat{s}_j(\bar{x}, \bar{x}_r, T-t) \frac{\partial^2}{\partial t^2} s_j(\bar{x}, t) dt \quad (2.48)$$

$$K_{\mu}^*(\bar{x}, \bar{x}_r) = - \int_0^T 2\mu(\bar{x}) \hat{D}(\bar{x}, \bar{x}_r, T-t) : D(\bar{x}, t) dt \quad (2.49)$$

y

$$K_{\kappa}^*(\bar{x}, \bar{x}_r) = - \int_0^T \kappa(\bar{x}) [\nabla \cdot \hat{\bar{s}}(\bar{x}, \bar{x}_r, T-t)] [\nabla \cdot \bar{s}(\bar{x}, t)] dt \quad (2.50)$$

De los que, la única diferencia con sus similares de las ecuaciones (2.33), (2.39) y (2.40) es la fuente del campo adjunto. Mientras que para los kernels de la inversión de forma de onda completa la fuente está dada por la diferencia entre los desplazamientos sintéticos y los observados, en este último caso está dada por la velocidad de los desplazamientos sintéticos. Esto significa que la propagación adjunta relacionada con los kernels de tomografía de tiempos de arribo es independiente de los datos observados. La expresión matemática de la fuente está dada por

$$\hat{f}_i^*(\bar{x}, t) = \frac{1}{N_r} w_r(T-t) \frac{\partial}{\partial t} s_i(\bar{x}_r, T-t, \bar{m}) \delta(\bar{x} - \bar{x}_r) \quad (2.51)$$

en donde w_r es una ventana de correlación cruzada y N_r es un factor de normalización.

Con el aporte de la generación de estos kernels de sensibilidad, procederíamos al siguiente paso

en la teoría que nos conlleva a crear un modelo de velocidades. Modelo el cual con ayuda de los Kernels generamos a través de distintos métodos.

2.5. Modelo de velocidades

En la industria existen diferentes métodos usados para construir el modelo de velocidades para PSDM. “Con estos métodos se obtiene un modelo de velocidad suficiente para producir una imagen preliminar, y posteriormente, el análisis de velocidad de migración incluye muchas iteraciones de migración en profundidad para producir la imagen final” Van Trier [1990]; Tieman [1995]. “Generalmente estos procedimientos proporcionan imágenes satisfactorias en áreas con buena calidad de datos, muchos pozos e información geológica a priori” Ratcliff et al. [1992].

Sin embargo, es frecuente que no exista mucha información en regiones de interés geológico fuera de las cuencas sedimentarias. Es por ello que en este caso, debe recurrirse a otros métodos de construcción del modelo de velocidades.

Hay distintas formas de realizar los modelos, pero la mayoría corresponde a una categoría de métodos conocidos como inversión tomográfica. La tomografía utiliza la información de tiempo de tránsito derivada de los datos sísmicos para refinar los modelos de velocidad. Una tomografía de reflexión clásica utiliza la diferencia entre tiempos de tránsito estimados y observados. El trazado de rayos calcula los tiempos de arribo de las reflexiones en conjuntos de datos de punto común de reflexión en los puntos de control. En cada conjunto de trazas el tiempo de arribo real del reflector más somero se compara con los tiempos de arribo estimados y utiliza la velocidad que mejor nivela los tiempos de arribo reales para actualizar el modelo. Este paso consume muchas horas hombre y requiere la participación de especialistas tanto en procesamiento como en interpretación, para confeccionar un modelo que se ajuste a todos los datos en todos los puntos de control.

La inversión de forma de onda completa es una herramienta utilizada para encontrar los parámetros del modelo deseado con una alta resolución, siendo el más común más no ex-

clusivo, la velocidad de onda p .

Se busca los parámetros del modelo más óptimos al minimizar el error entre los datos medidos y el sismograma sintético. Esto se logra mediante un proceso iterativo basado en la optimización de un gradiente como modelo base. La inversión toma en cuenta la minimización del error en tiempos de viaje, amplitudes, ondas convertidas, etc..., lo cual lo diferencia de métodos como la tomografía de tiempos de viaje.

Al trabajar sobre un modelo inicial de gradiente reduce el número de iteraciones comparado con el método de Monte Carlo Nocedal and Wright [2006]; Tarantola [2005] que genera modelos aleatorios. En el interés por solucionar la problemática generada por la sísmica sub-salina se han desarrollado distintas metodologías para la inversión, siendo la más importante la Inversión de diferencias finitas en contraste de fuente (FDCSI), en esta se trabaja la ecuación de onda en frecuencias y se supone una densidad constante en viaje de ida y regreso de las ondas. Este método se ha usado para la RTM en datos 2D con una condición simple de correlación cruzada en la imagen, McMechan [1983] y Symes [2007]. Una de las metas del trabajo propuesto es crear un algoritmo que trabaje de forma conjunta a una RTM para datos 3D, que además no asuma una densidad constante, sino que la tome como un parámetro a la hora de calcular el modelo. Para solucionar el costo de cómputo debido a las numerosas iteraciones del programa se han utilizado diversas técnicas como la codificación de fase, (Krebs et al., 2009), que combina los disparos como una fuente simultánea, esto reduce el costo de cómputo en proporción al número de disparos.

También se ha intentado transformar los datos usando Fourier u otras transformadas, sin embargo tienen el problema de no incluir estructuras geológicas que se ven en el sismograma y tienden a dar como resultado modelos geológicamente ilógicos. Otro problema notorio en los métodos de inversión es el de la no unicidad dando múltiples soluciones, este se ha intentado solucionar con diversos métodos que ignoran los mínimos locales. Una solución propuesta es

usar los dominios logarítmicos y de Laplace, Shin and Min [2006]; Shin and Ha [2008], además de la inclusión de información a priori obtenida de pozos y conocimiento geológico regional.

2.6. Problema Inverso

Un concepto de gran importancia para entender la teoría del problema inverso discreto es el de estado de la información sobre un parámetro dado. "Se postula que la forma más general de describir tal estado de información es definir una densidad de probabilidad sobre el espacio del parámetro", Tarantola [2005]. De ello se puede sacar que los resultados de las mediciones de los parámetros observables, conocidos como datos, la información a priori sobre los parámetros del modelo, y la información sobre la correlación física entre parámetros observables y los parámetros del modelo pueden ser descritos como densidades de probabilidad. El problema inverso generalmente se puede ver como la combinación de toda esta información. La solución de los problemas inversos, y el análisis de la incertidumbre puede realizarse de forma totalmente no lineal con la limitante de tener una gran cantidad de tiempo de computación.

El estudio de un sistema físico puede reducirse a tres pasos a seguir según la teoría de inversión:

- 1) Parametrización del sistema: descubrimiento de un pequeño conjunto de parámetros del modelo cuyos valores caracterizan completamente al sistema, desde un punto de vista dado.
- 2) Modelado directo: descubrimiento de las leyes físicas que nos permiten, para valores dados de los parámetros del modelo, hacer predicciones sobre algunas de las mediciones de los parámetros observables.
- 3) Modelado inverso: uso de los resultados reales de algunas mediciones de los parámetros observable para inferir los valores reales de los parámetros del modelo.

2.7. Discretización 3D

Para lograr una buena base de la distribución espacial de esfuerzos y velocidades se parte de un modelo 3D, por ende la figura 2.2 representa la discretización del dominio 3D, el sistema de referencia define el dominio sobre el cual la propagación es realizada. Los ejes x, y forman un plano horizontal y el eje vertical z con dirección positiva hacia abajo representa la profundidad. En las siguientes ecuaciones n es el índice derivado de la discretización temporal, i representa los nodos de la malla distribuidos sobre el eje x , j el índice para la discretización en y y k es utilizada para identificar los puntos de cálculo del eje z . El paso temporal lo define Δt y $\Delta x, \Delta y, \Delta z$ reflejan el tamaño en la discretización de los ejes x, y, z respectivamente. Los cálculos para la velocidad v se realizan en el tiempo $(t + \frac{1}{2}) \Delta t$ y los esfuerzos para el tiempo $(t + 1) \Delta t$. La discretización está basada en trabajos de Komatitsch and Martin [2007].

2.8. Condición de estabilidad

Con el análisis de estabilidad es evaluada la aplicabilidad de los métodos numéricos para estudiar la evolución de la propagación de ondas, el criterio determina los incrementos en espacio y tiempo necesarios para realizar las simulaciones. Las condiciones de estabilidad se expresan por lo general como una condición de Courant-Friedrichs-Lewy (CFL), la cual es una desigualdad que acota el incremento en tiempo, con una constante que multiplica el tamaño de celda y divide por la velocidad, donde la constante depende del método numérico en cuestión (e.g., Komatitsch and Martin [2007]).

En un medio homogéneo la condición de estabilidad para un esquema explícito está dado por:

$$v_p \Delta t = \sqrt{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2}} < 1 \quad (2.52)$$

donde v_p es la velocidad de onda P. La condición de estabilidad es independiente de la velocidad de onda v_s ; para un espacio n -dimensional la condición es descrita por la siguiente desigualdad:

$$v_p \Delta t = \sqrt{\sum_{i=1}^n \frac{1}{\Delta x_i^2}} < 1 \quad (2.53)$$

si la discretización es uniforme para todas las dimensiones, es decir $\Delta x_i = \Delta x$, entonces se cumple la siguiente expresión Virieux [1986]:

$$v_p \frac{\Delta t}{\Delta x} < \frac{1}{\sqrt{n}} \quad (2.54)$$

La Figura 2.2 presenta la celda unitaria que describe la distribución de los nodos, sobre la cual se lleva a cabo el cálculo de esfuerzos y velocidades. Las líneas continuas muestran la

2.9. Diferencias finitas

Los métodos matemáticos analíticos no proporcionan soluciones con modelos de estructuras complejas, sin embargo, estos modelos son necesarios si se quiere aproximar a estructuras reales. Los métodos numéricos transforman una diferencial o formulación integral de un problema en un sistema de ecuaciones algebraicas, esto es porque las operaciones algebraicas se pueden realizar eficientemente usando las computadoras. Una función continua tiene que ser representada por un conjunto finito de números. Los métodos numéricos difieren en cómo resuelven esta tarea. Hay dos aspectos básicos de cada método numérico: la precisión y la eficiencia computacional. Estos dos aspectos son en la mayoría de los casos contradictorios y debe considerarse que tan importante es la precisión o la eficiencia computacional.

En general, los métodos de malla-punto son más sencillos y fáciles de implementar en los códigos informáticos, en comparación con los métodos de expansión en serie. Probablemente debido a esto, han estado en uso por más tiempo. En los métodos de malla-punto, cada función está representada por sus valores en puntos de la malla. La distribución espacial-temporal de puntos de malla puede ser, en principio, arbitraria, pero afecta significativamente a las propiedades de la aproximación numérica resultante.

En el método de diferencias finitas usualmente no hay supuestos sobre valores funcionales entre los puntos de la malla. Un derivado de una función puede encontrarse utilizando una fórmula de diferencias finitas que hace uso de valores de función en un conjunto puntos de la malla.

En el método del volumen finito se hace una suposición acerca de los valores funcionales intermedios de los puntos de la malla.

La aplicación del método a un problema diferencial particular conlleva la construcción de un modelo discreto de diferencias finitas del problema, Michéa and Komatitsch [2010],

la cobertura del dominio computacional por una malla espacio-tiempo, funciones, condición inicial y / o condición de frontera en los puntos de la malla, construcción de un sistema de diferencia finita, así como el análisis del modelo FD (consistencia y orden de la aproximación, estabilidad, convergencia), el análisis del modelo FD o cálculos numéricos puede conducir a una redefinición de la red y aproximaciones FD, si el comportamiento numérico no es satisfactorio.

Consideramos una geología elástica lineal isotrópica para el medio sólido, Y por lo tanto la ecuación de onda sísmica se puede escribir en la forma diferencial

$$\begin{aligned}\rho\ddot{u} &= \nabla \cdot \sigma + f, \\ \sigma &= C : \varepsilon, \\ \varepsilon &= \frac{1}{2}[\nabla u + (\nabla u)^T],\end{aligned}\tag{2.55}$$

Donde u denota el vector de desplazamiento, σ el simétrico, tensor de esfuerzo de segundo orden, ε el tensor de deformación simétrico de segundo orden, C el tensor de rigidez de cuarto orden, ρ la densidad y f una fuente fuerza externa. La operación de contracción de doble tensor se denomina por un dos puntos, un exponente T denota la transposición, y un punto sobre un símbolo indica la diferenciación del tiempo. El dominio físico de la modelo se denotan por Ω y su límite exterior por Γ . En la formulación de velocidad-esfuerzo clásica que se utiliza en la mayoría de los usos de FD se reescribe (2.55)

$$\begin{aligned}\rho\partial_t v &= \nabla \cdot \sigma, \\ \partial_t \sigma &= C : \nabla v,\end{aligned}\tag{2.56}$$

La condición de frontera en la superficie libre del medio es que el vector de tracción τ debe ser cero en todas partes en la superficie Γ , que es,

$$\tau = \sigma \cdot \hat{n} = 0, \tag{2.57}$$

Donde \hat{n} es la normal a la superficie Γ .

2.10. Cómputo paralelo

El objetivo de paralelizar un programa es reducir sus tiempos de cómputo, en este caso se busca mejorar un algoritmo existente en Fortran 95 diseñado en Instituto Mexicano del Petróleo.

El algoritmo en CUDA C es realizado con una computadora portátil con procesador NVIDIA GPU GeForce GTX 970M, que a su vez cuenta con 1280 núcleos CUDA, también cuenta con un procesador CPU Intel Core i7 que procesa a una frecuencia de 2.60 GHz y RAM de 8 GB. Esto es necesario para la programación paralela con el cual se optimizará el algoritmo.

El algoritmo en Fortran MPI se trabajó en un servidor con una tarjeta Matrox Electronics Systems MGA G200eW, con un procesador Intel Xeon CPU de 10 núcleos E5-2690 de 3.00GHz.

Para la construcción de imágenes del subsuelo las empresas tradicionalmente han usado plataformas de cómputo hasta con miles de CPU, sin embargo los sistemas de cómputo basados en procesadores tradicionales se han visto frenados principalmente por tres causas: la primera consiste en la relación cúbica entre la frecuencia y la potencia (power wall); otra causa es la gran diferencia que existe entre la velocidad a la cual los datos son transferidos al procesador y la velocidad a la que éste ejecuta las instrucciones, esta limitación se conoce con el nombre de muro de memoria (memory wall); finalmente las posibilidades para que los procesadores continúen aumentando el paralelismo a nivel de instrucción IPL (Instruction Level Parallelism) no son prometedoras, este efecto se conoce como IPL Wall Brodtkorb [2010]. Debido a las tres barreras mencionadas anteriormente, la computación de alto rendimiento se ha encaminado hacia la computación híbrida en donde las CPU trabajan conjuntamente con una serie de nuevas arquitecturas que aprovechan mejor el paralelismo de las aplicaciones. Donde en esta tesis se basará en el uso de CUDA el cual es una arquitectura de cálculo paralelo de NVIDIA que aprovecha la gran potencia de la GPU (unidad de procesamiento gráfico) para proporcionar

un incremento extraordinario del rendimiento del sistema.

Es un pensamiento obvio que aumentando aun más la potencia de procesamiento se convierte inútil en algún punto si no estas listo, como programador, para alimentar el procesador con datos a una velocidad suficiente. Los CPUs tradicionalmente han contrarrestado el vacío en expansión entre el procesador y el rendimiento de memoria con caches cada vez más grandes que mantienen frecuentemente datos usados en memoria de uso rápido en el chip. Como sea, incrementando el tamaño del cache incrementa el estado latente cayendo en una disminución de ganancia en el rendimiento R. Holdahl and Lie [1999].

2.10.1. Programación en CUDA C

La plataforma de cálculo paralelo CUDA proporciona unas cuantas extensiones de C y C++ que permiten implementar el paralelismo en el procesamiento de tareas y datos con diferentes niveles de granularidad. El programador puede expresar ese paralelismo mediante diferentes lenguajes de alto nivel como C, C++ y Fortran. En la actualidad, la plataforma CUDA se utiliza en miles de aplicaciones aceleradas en la GPU y en miles de artículos de investigación publicados. La computación sobre tarjetas gráficas se puede definir como el uso de una tarjeta gráfica (GPU - Graphics Processing Unit) para realizar cálculos científicos de propósito general. Este modelo de computación sobre tarjetas gráficas consiste en usar conjuntamente una CPU (Central Processing Unit) y una GPU de manera que formen un modelo de computación heterogéneo. Siguiendo este modelo, la parte secuencial de una aplicación se ejecutaría sobre la CPU (comúnmente denominada host) y la parte más costosa del cálculo se ejecutaría sobre la GPU (que se denomina device). Desde el punto de vista del usuario, la aplicación simplemente se va a ejecutar más rápido porque está utilizando las altas prestaciones de la GPU para incrementar el rendimiento (Represa Pérez et al. [2016]).

NVIDIA fue consciente del potencial que suponía acercar este enorme rendimiento a la

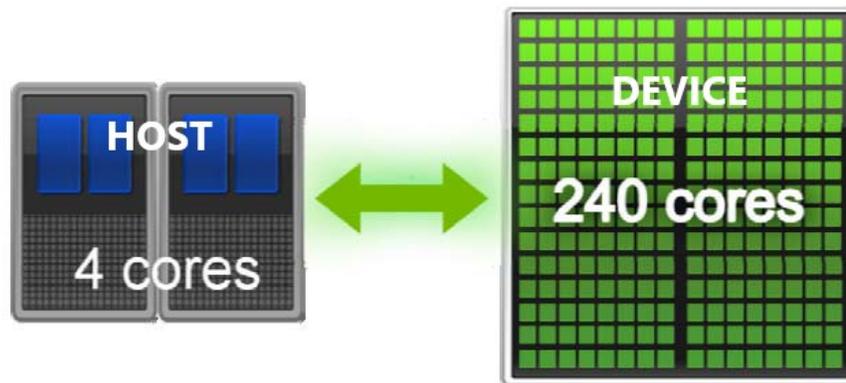


Figura 2.3: Modelo de computación en sistemas heterogéneos: CPU + GPU

comunidad científica en general y decidió investigar la forma de modificar la arquitectura de sus GPUs para que fueran completamente programables para aplicaciones científicas además de añadir soporte para lenguajes de alto nivel como C y C++. De este modo, en Noviembre de 2009, NVIDIA introdujo para sus tarjetas gráficas la arquitectura CUDATM (Compute Unified Device Architecture), una nueva arquitectura para cálculo paralelo de propósito general, con un nuevo repertorio de instrucciones y un nuevo modelo de programación paralela, con soporte para lenguajes de alto nivel (Represa Pérez et al. [2016]).

Cuando se trabaja en este ambiente el programador escribe un programa secuencial dentro del cual llama a lo que se conoce como kernel, estos pueden ser una simple función o un programa completo. Este kernel se ejecuta de forma paralela dentro de la GPU como un conjunto de hilos (threads) y que el programador organiza dentro de una jerarquía en la que pueden agruparse en bloques (blocks), y que a su vez se pueden distribuir formando una malla (grid).

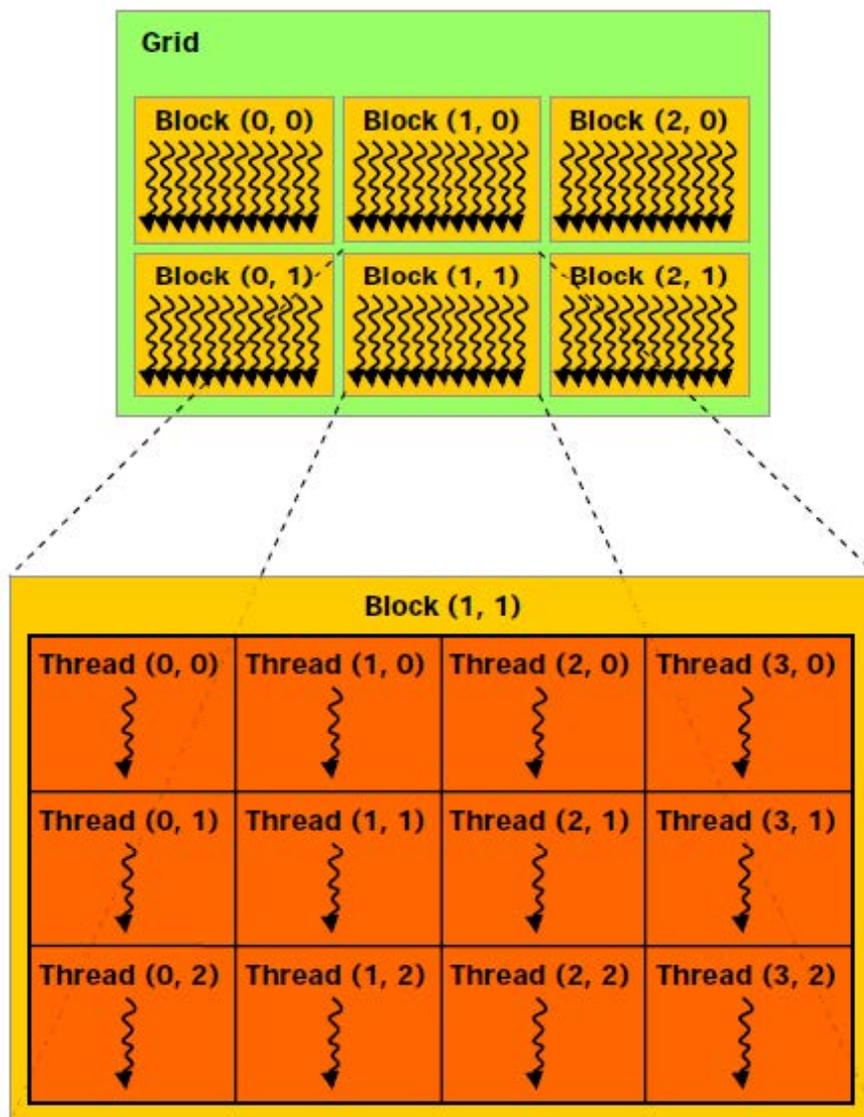


Figura 2.4: Jerarquía de hilos en una aplicación CUDA

2.10.2. Programación en Fortran MPI

El pase de mensajes es un modelo de comunicación ampliamente usado en computación paralela. En años recientes se han logrado desarrollar aplicaciones importantes basadas en este paradigma. Dichas aplicaciones (incluyendo algunas del dominio público) han demostrado que es posible implementar sistemas basados en el pase de mensajes de una manera eficiente y portable. El crecimiento en el volumen y diversidad de tales aplicaciones originaron la necesidad de crear un estándar, es así como surge MPI o Message Passing Interface MPI es un estándar para la implementación de sistemas de pase de mensajes diseñado por un grupo de investigadores de la industria y la academia para funcionar en una amplia variedad de computadores paralelos y de forma tal que los códigos sean portables. Su diseño esta inspirado en máquinas con una arquitectura de memoria distribuida en donde cada procesador es propietario de cierta memoria y la única forma de intercambiar información es a través de mensajes, sin embargo, hoy en día también encontramos implementaciones de MPI en máquinas de memoria compartida, (Hidrobo and Hoeger [2005]).

Capítulo 3

Esquema numérico

En este capítulo se describirá con la mayor claridad posible el esquema numérico que bajo una metodología planea cumplirse los objetivos.

3.1. Definición de parámetros

Los parámetros a utilizar para este algoritmo de migración reversa en tiempo se basan en una sección de dos dimensiones:

- Intervalo de muestreo (x,z): 20 metros.
- Intervalo de muestreo (t): 0.0025 segundos.
- Tiempo total: 8 segundos.
- Distancia horizontal total: 17000 metros.
- Profundidad total: 11280 metros.

- Fuente:

$$f(t) = -2A\gamma(t - t_0)e^{-\gamma(t-t_0)^2}, \gamma = 2\pi^2 f_0^2 \quad (3.1)$$

Se define para esta sección de dos dimensiones que se contemplen 85 fuentes y 850 receptores dando así 10 receptores para cada fuente. El número total de muestras para x, z, y el tiempo son la razón de la distancia horizontal total y el intervalo de muestreo, la distancia vertical total y el intervalo de muestreo y el tiempo total con el intervalo de muestreo respectivamente. Dando lugar a:

- $nm_x=850$ muestras para x
- $nm_z= 564$ muestras para z
- $nmt= 3200$ muestras para t

Las fronteras horizontales y verticales para el dominio central están definidas como el número de muestras totales más el intervalo de muestreo horizontales y verticales, es decir:

- frontera x (izquierdo) = $nm_x+(\text{intervalo de muestreo})=870$
- frontera x (derecho) = $nm_x+2*(\text{intervalo de muestreo})=890$
- frontera z (superior) = $nm_z+(\text{intervalo de muestreo})=584$
- frontera z (inferior) = $nm_z+2*(\text{intervalo de muestreo})=604$

Ahora para evaluar la condición de estabilidad, se propone dos variables las cuales no deben sobrepasar la velocidad del medio con

$$a = \frac{dt}{dx} \quad (3.2)$$

y

$$b = \frac{1}{(\sqrt{2})5480} \quad (3.3)$$

con lo que la velocidad queda definida en base a los intervalos de muestreo horizontal y el de muestreo en tiempo.

La frecuencia fundamental es utilizada para definir la variable alfa denotada como:

$$\alpha = 2 (f_0 \pi)^2 \quad (3.4)$$

dónde $f_0 = 5,0[s]$ y el periodo fundamental quedaría como $\frac{1}{f_0}$.

3.2. Kernels de sensibilidad

Debido a que se necesita de kernels de sensibilidad para la generación de imágenes sísmicas, que para este trabajo se enfocará en 2D (x, z), se parte de la idea de que se emplearan para la migración reversa en tiempo (RTM) el cual se podrá visualizar en un trabajo posterior, que tiene como base la ecuación que define la forma de onda completa, utilizando como condición de imagen para un parámetro asociado (densidad o parámetros elásticos) el kernel de sensibilidad dentro del cual de manera implícita conlleva la inversión de forma de onda completa; la inversión se define en función del error entre los datos observados y los sintéticos, para este objetivo Tromp et al. [2005a] apoya su trabajo en la diferencial de Fréchet de una función error.

El kernel caracteriza la sensibilidad de la función error respecto a cada uno de los parámetros, una perturbación en el medio se puede identificar como δp o δc_{jklm} que caracteriza las propiedades de un medio elástico definiendo una variación en densidad ρ o sus parámetros elásticos c_{jklm} ; considerando un medio isótropo basta caracterizar las constantes elásticas κ y ρ

De acuerdo a los desarrollos de Tromp et al. [2005a], los kernels quedan expresados de la siguiente manera:

$$K_{\rho}(x) = - \int_0^T \rho(x) s^+ (\bar{x}, t - t') \cdot \partial_t^2 s(x, t) dt \quad (3.5)$$

$$K_{\kappa}(x) = - \int_0^T \kappa(x) [\nabla \cdot s^+ (\bar{x}, t - t')] [\nabla \cdot s(x, t)] dt \quad (3.6)$$

donde s^+ y s es el campo de desplazamiento adjunto y original. K_{ρ} involucra el producto punto del campo de desplazamientos adjunto y el campo de aceleraciones de la propagación original, en el caso de K_{κ} lo definen las derivadas espaciales del campo de desplazamiento

adjunto y original, en los dos casos se tiene un periodo de registro de $[0, T]$.

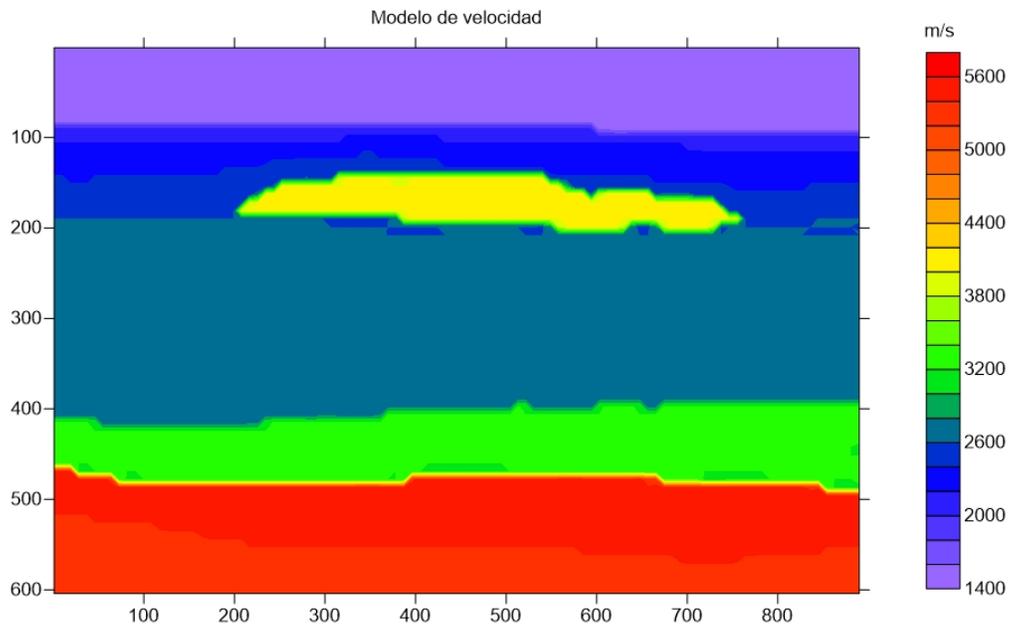


Figura 3.1: Modelo de velocidades en gradiente con un cuerpo de mayor velocidad en las capas superiores

Con la construcción de los kernels realizada damos paso a la paralelización del algoritmo e implementación en datos reales los cuales involucran migración RTM 2D.

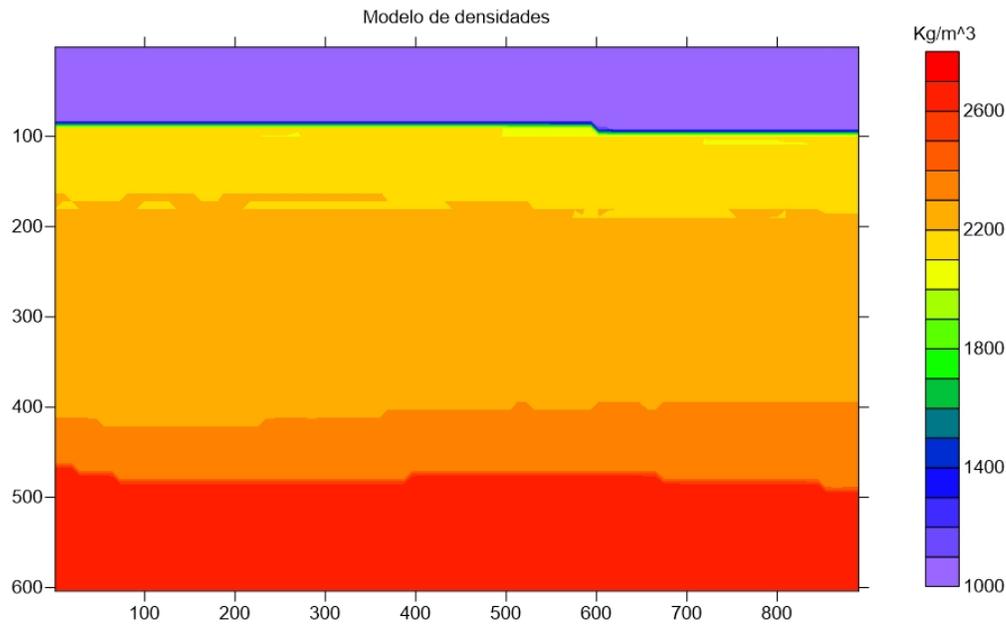


Figura 3.2: Modelo de densidades en gradiente

3.3. Fuentes (s) y Receptores (r)

Para las posiciones de las fuentes(s) y receptores(r) se contemplan 85 fuentes y 850 receptores dando así 10 receptores para cada fuente, los receptores para la dirección horizontal x del punto inicial 0 se recorre el primer receptor a 21 metros de distancia quedando en la posición $x = 22[m]$, es decir el primer receptor se va a ubicar

$$x_r(i) = 20 + i$$

y en la dirección de profundidad z , todos los receptores estarán a la misma profundidad de 22 metros

$$z_r(i) = 22$$

Ψ	x_1	z	x_2	x_{1i}	z_{1i}	x_{2i}	z_s	x_{1s}	z_{1s}	x_{2s}	z_{2s}
s_{xx}	$\Psi_{s_{xxx_1}}$	$\Psi_{s_{xxz}}$	$\Psi_{s_{xxx_2}}$	$\Psi_{s_{xxx_{1i}}}$	$\Psi_{s_{xxz_{1i}}}$	$\Psi_{s_{xxx_{2i}}}$	$\Psi_{s_{xxz_s}}$	$\Psi_{s_{xxx_{1s}}}$	$\Psi_{s_{xxz_{1s}}}$	$\Psi_{s_{xxx_{2s}}}$	$\Psi_{s_{xxz_{2s}}}$
s_{xz}	$\Psi_{s_{xxz_1}}$	$\Psi_{s_{xxz}}$	$\Psi_{s_{xxz_2}}$	$\Psi_{s_{xxz_{1i}}}$	$\Psi_{s_{xxz_{1i}}}$	$\Psi_{s_{xxz_{2i}}}$	$\Psi_{s_{xxz_s}}$	$\Psi_{s_{xxz_{1s}}}$	$\Psi_{s_{xxz_{1s}}}$	$\Psi_{s_{xxz_{2s}}}$	$\Psi_{s_{xxz_{2s}}}$
v_x	$\Psi_{v_{xx_1}}$	$\Psi_{v_{xz}}$	$\Psi_{v_{xx_2}}$	$\Psi_{v_{xx_{1i}}}$	$\Psi_{v_{xz_{1i}}}$	$\Psi_{v_{xx_{2i}}}$	$\Psi_{v_{xz_s}}$	$\Psi_{v_{xx_{1s}}}$	$\Psi_{v_{xz_{1s}}}$	$\Psi_{v_{xx_{2s}}}$	$\Psi_{v_{xz_{2s}}}$
v_z	$\Psi_{v_{zx_1}}$	$\Psi_{v_{zz}}$	$\Psi_{v_{zx_2}}$	$\Psi_{v_{zx_{1i}}}$	$\Psi_{v_{zz_{1i}}}$	$\Psi_{v_{zx_{2i}}}$	$\Psi_{v_{zz_s}}$	$\Psi_{v_{zx_{1s}}}$	$\Psi_{v_{zz_{1s}}}$	$\Psi_{v_{zx_{2s}}}$	$\Psi_{v_{zz_{2s}}}$

Cuadro 3.1: Variables a utilizar para las fronteras absorbentes

Para las fuentes (sources), se ubican centradas para lograr cubrir una cantidad de receptores equilibrada en la dirección horizontal x del punto inicial 1 hacia una parte central

$$x_s(i) = 21 + (i - 1) * 10$$

es decir que, la primer fuente se ubicará a 22 metros de distancia del origen, de igual manera en la dirección de profundidad z , todas las fuentes estarán a la misma profundidad de 22 metros

$$z_s(i) = 22$$

Una vez establecido lo anterior se inicializan las variables de las fronteras absorbentes denominadas con la letra griega Ψ , dando lugar a cuatro variables por frontera

$$s_{xx}, s_{xz}, v_x, v_z$$

las cuales describen los esfuerzos de compresión (s_{xx}), esfuerzos de cizalla (s_{xz}), la velocidad en dirección x (v_x) y en dirección z (v_z), como se observa en el cuadro 3.1.

Se necesitaron de dos perfiles de amortiguamiento para las fronteras absorbentes una en

dirección x d_x y otra en dirección z d_z expresados de la siguiente manera:

$$\begin{aligned} dx_0 &= -(N + 1)(5450) \frac{\log(R_c)}{2 * pml_x * dx} \\ dz_0 &= -(N + 1)(5450) \frac{\log(R_c)}{2 * pml_z * dz} \end{aligned} \quad (3.7)$$

y de acuerdo a la ecuación 3.4, alfa máxima quedaría como:

$$\alpha_{max} = 2 * \pi * \frac{f_0}{2} \quad (3.8)$$

Dónde en la ecuación 3.7, N es la dimensión total del dominio (x,z), R_c es el coeficiente teórico de reflexión =0.1, dx , dz son el intervalo de muestreo en la dirección x y en la dirección z y respectivamente pml_x y pml_z son las muestras que las fronteras PML tienen en la dirección x y z.

3.4. Fronteras CPML

En las décadas pasadas se han desarrollado una gran variedad de fronteras absorbentes, especialmente para el modelado de la onda sísmica. Para el caso de estudio en cuestión, como el dominio en el que se propagan las ondas es una malla rectangular que representa parte del subsuelo, se tienen cuatro fronteras. La frontera superior corresponde a la superficie libre, un límite físico en el que los esfuerzos son nulos, por lo que al momento de programarla es suficiente con fijar $P = \sigma_{xx} = \sigma_{zz} = 0$ a esta condición también se le conoce como condición de Neuman.

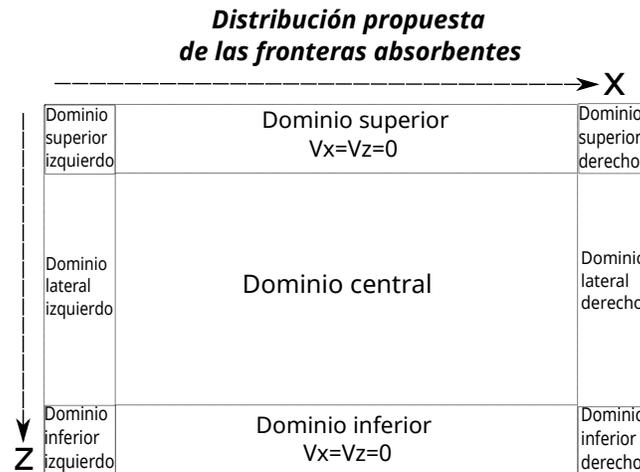


Figura 3.3: Distribución de las capas CPML laterales e inferiores en nuestro problema bidimensional

La dificultad de modelar estas fronteras yace en que no representan un límite físico, ya que en la realidad el subsuelo continúa en las tres direcciones. La consecuencia que conlleva el tratar las fronteras de manera convencional es que la energía se refleja al llegar a ellas, provocando que la energía quede atrapada en el espacio discreto, a diferencia de lo que ocurre en una propagación real, en donde las ondas continúan propagándose por el subsuelo sin sufrir

reflexión por el borde del área de estudio. Una solución propuesta podría ser el incrementar el tamaño de la malla hasta que no se registren reflexiones debidas a las fronteras del área de interés durante el tiempo en el que se quiere simular el fenómeno. Sin embargo, esto supondría un incremento en el número de operaciones, resultando ineficaz. Para evitar esto, se debe absorber la energía en las fronteras utilizando condiciones de frontera absorbentes.

En Berenger [1994] introdujo una técnica de condiciones de frontera absorbentes para las ecuaciones de Maxwell, la condición de capa perfectamente ajustadas, conocida en la literatura como PML (por sus siglas en inglés, Perfectly Matched Layer), cuya principal característica son los coeficientes de reflexión nulos para cualquier ángulo de incidencia y para cualquier frecuencia, antes de discretizar el modelo. Esta PML, conocida como PML clásica, tiene el problema de que los coeficientes de reflexión dejan de ser nulos después de la discretización, y que se vuelven muy grandes cuando los ángulos de incidencia son bajos. Posteriormente, de entre los varios intentos por mejorar la PML clásica, Kuzuoglu and Mittra [1996] y, Roden and Gedney [2000] desarrollaron la PML convolucional o CPML (por sus siglas en inglés, Convolutional Perfectly Matched Layer) para las ecuaciones de Maxwell y fueron adaptadas a las ecuaciones de la elastodinámica por Komatitsch and Martin [2007]. Éstas últimas son las que son utilizadas para simular el problema directo en el presente trabajo.

Matemáticamente hablando en las PML se siguen cumpliendo las mismas ecuaciones diferenciales que en el dominio real, siendo la única diferencia de que los ejes coordenados son transformados del dominio real al dominio complejo, por lo que resulta necesario reescribir las ecuaciones diferenciales en términos de las nuevas variables independientes. Según el desarrollo de Komatitsch and Martin [2007] se parte de que tiene una frontera absorbente PML lateral paralela al eje z , ubicada en la parte positiva del eje x , la transformación del eje x está determinada por

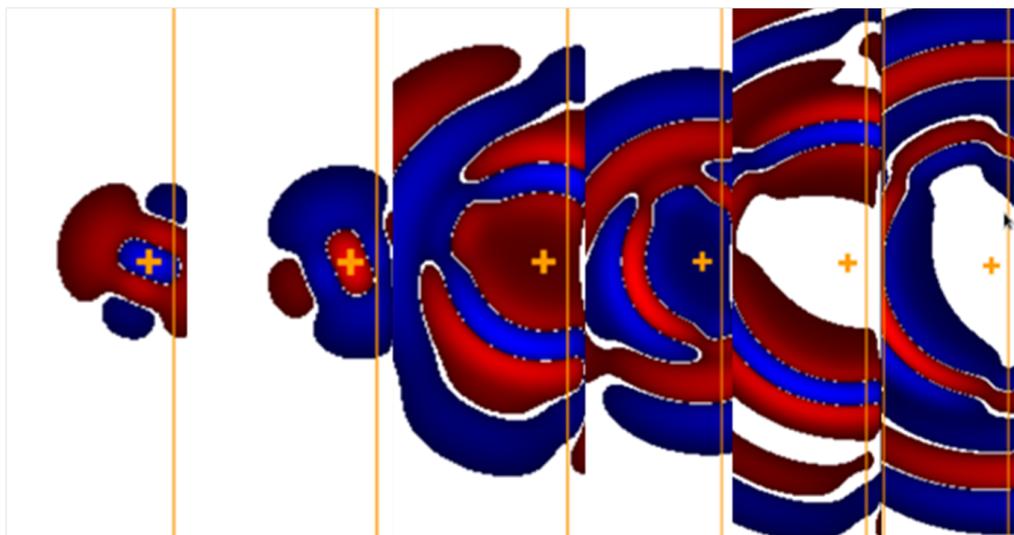


Figura 3.4: Imágenes de la componente v_y en un plano 2-D situado en el centro de la malla a lo largo del eje z para un medio visco elástico correspondiente a una rebanada delgada con condiciones CPML

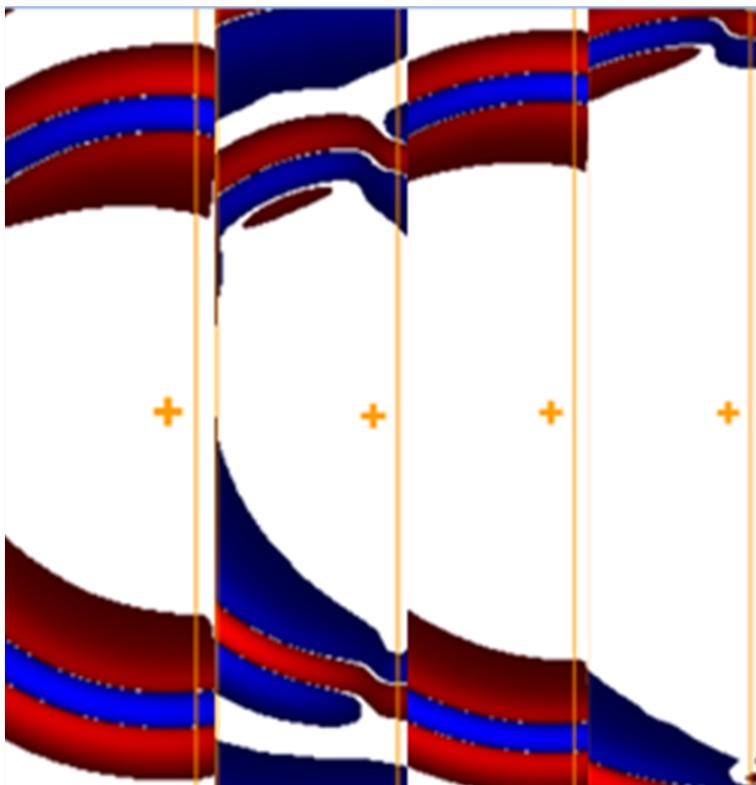


Figura 3.5: Imágenes de la componente v_y en un plano 2-D situado en el centro de la malla a lo largo del eje z para un medio visco elástico correspondiente a una rebanada delgada con condiciones CPML

$$\vec{x}(x) = x - \frac{i}{\omega} \int_0^x d_x(s) ds, \quad (3.9)$$

donde $d_x(x)$ es una función de amortiguamiento en la región de la PML. El eje x se ve afectado por el cambio de ejes coordenados, pero la función de amortiguamiento es nula en el dominio original, es decir, $d_x(x) = 0$ en el dominio sin PML. Esta transformación de ejes implica reescribir las ecuaciones diferenciales sustituyendo los ejes originales por los transformados y para implementarlas numéricamente, se deben expresar en términos de los ejes originales. La expresión para la derivada respecto al nuevo eje coordenado se puede obtener por la regla de la cadena,

$$\frac{\partial}{\partial \vec{x}} = \frac{\frac{\partial}{\partial x}}{\frac{\partial \vec{x}}{\partial x}} = \frac{\frac{\partial}{\partial x}}{\frac{\partial}{\partial x} [x - \frac{i}{\omega} \int_0^x d_x(s) ds]} = \frac{\frac{\partial}{\partial x}}{1 - \frac{i}{\omega} d_x(x)} \quad (3.10)$$

$$\frac{\partial}{\partial \vec{x}} = \frac{i\omega}{i\omega + d_x(x)} \frac{\partial}{\partial x} \quad (3.11)$$

Introduciendo el auxiliar S_x

$$S_x(x, \omega) = \frac{i\omega + d_x(x)}{i\omega} = 1 + \frac{d_x(x)}{i\omega} \quad (3.12)$$

$$\frac{\partial}{\partial \vec{x}} = \frac{1}{S_x(x, \omega)} \frac{\partial}{\partial x} \quad (3.13)$$

La diferencia principal entre la PML clásica y la CPML es la introducción de dos parámetros nuevos en la función auxiliar S_x , tales que

$$S_x(x, \omega) = k_x(x) + \frac{d_x(x)}{\alpha_x(x) + i\omega} \quad (3.14)$$

Para dado que S_x se encuentra en el dominio de la frecuencia, hay que cambiarlo para que cumpla el esquema de diferencias finitas en el dominio del tiempo. Para cambiar de dominio se utiliza el teorema de la convolución

$$s_x(x, t) = \mathcal{F}^{-1} \left\{ \frac{1}{S_x(x, \omega)} \right\} \quad (3.15)$$

Para tener una notación más corta se introducen las funciones auxiliares

$$b_x(x) = e^{-[\frac{d_x(x)}{K_x(x)} + \alpha_x(x)]\Delta t} \quad (3.16)$$

$$a_x(x) = \frac{d_x(x)}{K_x(x)[d_x(x) + \alpha_x(x)K_x(x)]} [b_x(x) - 1] \quad (3.17)$$

$$\psi_x^n(x) = b_x(x)\psi_x^{n-1}(x) + a_x(x) \left(\frac{\partial}{\partial x} \right)^{n+\frac{1}{2}} \quad (3.18)$$

y la derivada queda

$$\frac{\partial}{\partial \vec{x}} = \frac{1}{K_x(x)} \frac{\partial}{\partial x} + \psi_x^n(x) \quad (3.19)$$

Para un caso bidimensional se tiene que hacer un cambio de ejes horizontales y verticales si se desea introducir fronteras absorbentes en ambas direcciones. De acuerdo con Gedney [1996] y Collino and Tsogka [2001], las funciones de amortiguamiento $d_x(x)$ y $d_z(z)$ se pueden definir como

$$d_x(x) = d_{x0} \left(\frac{x}{L_x} \right)^N, \quad (3.20)$$

$$d_z(z) = d_{z0} \left(\frac{z}{L_z} \right)^N, \quad (3.21)$$

donde L_x es la longitud de la PML lateral y L_z la de la PML inferior, con un coeficiente de reflexión teórico R_c y $N = 2$

$$d_{x0} = -(N + 1)V_p \frac{\text{Log}(R_c)}{2L_x}, \quad (3.22)$$

$$d_{z0} = -(N + 1)V_p \frac{\text{Log}(R_c)}{2L_z}, \quad (3.23)$$

Se definen $\alpha_x(x)$ y $\alpha_z(z)$ como funciones que varían linealmente desde un valor máximo $\alpha_{max} = \pi f_0$ en el inicio de la PML hasta cero en el fin de la PML. Y el valor de $K_x(x)$ y $K_z(z)$ se saca del artículo Komatitsch and Martin [2007] que $K_x(x) = K_z(z) = 1$ dando las ecuaciones de CPML.

CPML laterales

$$v_{x(i,j)}^{k+\frac{1}{2}} = v_{x(i,j)}^{k-\frac{1}{2}} + \frac{\Delta t}{\rho(i,j)} \left[\frac{\sigma_{xx(i+\frac{1}{2},j)}^k - \sigma_{xx(i-\frac{1}{2},j)}^k}{\Delta x} + \psi_{d\sigma_{xx}(i,j)}^k + \frac{\sigma_{xz(i,j+\frac{1}{2})}^k - \sigma_{xz(i,j-\frac{1}{2})}^k}{\Delta z} \right] \quad (3.24)$$

La figura 3.3 muestra la distribución de las capas CPML laterales e inferiores para el problema bidimensional en cuestión.

En la figura 3.4 y 3.5 se muestran los estados de los campos de presión P y velocidad vertical V_z durante diferentes instantes de la propagación obtenidos al emplear capas CPML.

Capítulo 4

Resultados

En este capítulo se muestran los resultados tanto de la propagación y retro propagación de la onda, los kernels obtenidos y una comparación entre los tiempos de cómputo de los distintos programas.

4.1. Resultado de la propagación

Se obtienen las imágenes de la propagación de la RTM con fronteras absorbentes, como se puede observar en la figura 4.1

Se aprecia que al inicio de la propagación de las ondas se tiene el cúmulo de las primeras ondas en reflejarse debido al modelo de velocidades identificando el primer cambio de velocidad, seguidas de su propagación hacia los límites horizontales, estas logran disipar la señal. Finalmente observamos como se disipa la energía a la frontera inferior absorbente tanto en sus flancos izquierdo, central y derecho, con lo que evitamos señales espuria.

Ahora en la propagación, las ondas reflejadas limpias se observan en la figura 4.2 y se puede observar que la señal se disipa a través de las fronteras absorbentes laterales continuando la

onda principal hacia la parte superior como se observa en los eventos finales. Se aprecian los kernel de sensibilidad kappa y rho en las figuras 4.3 y 4.4.

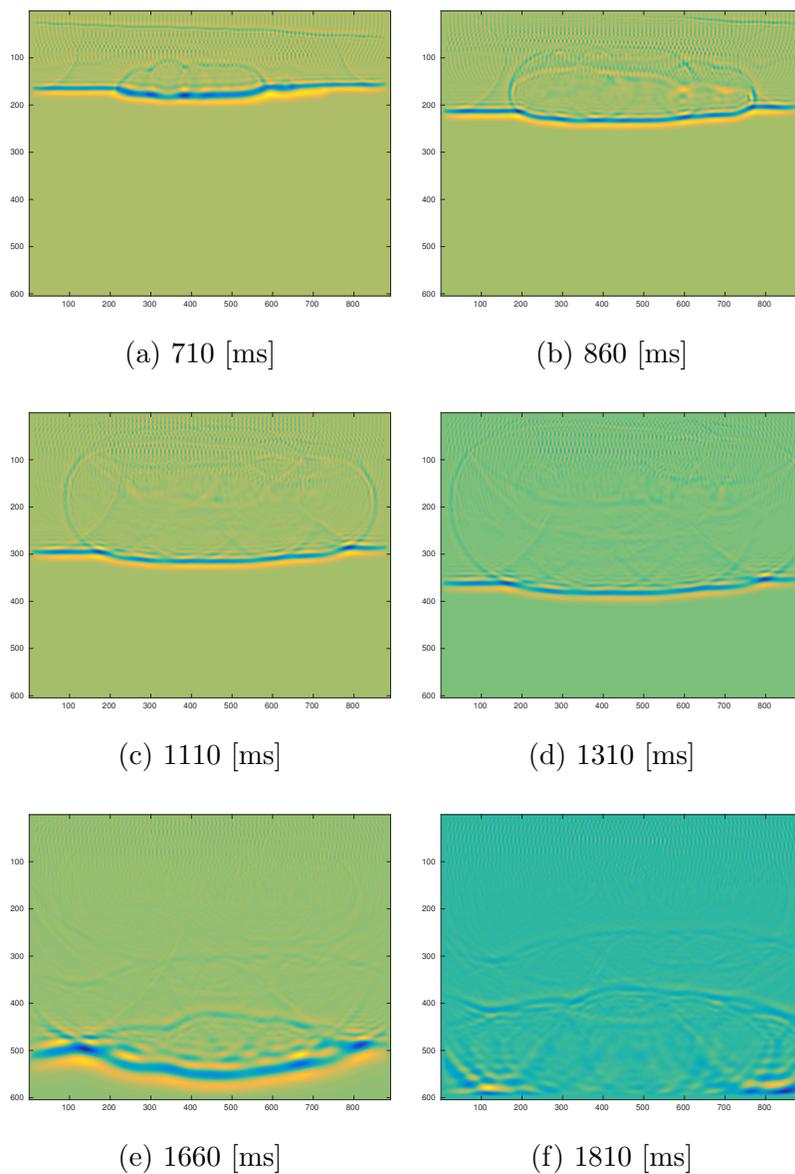


Figura 4.1: Inicio de la propagación de las ondas con las fronteras absorbentes

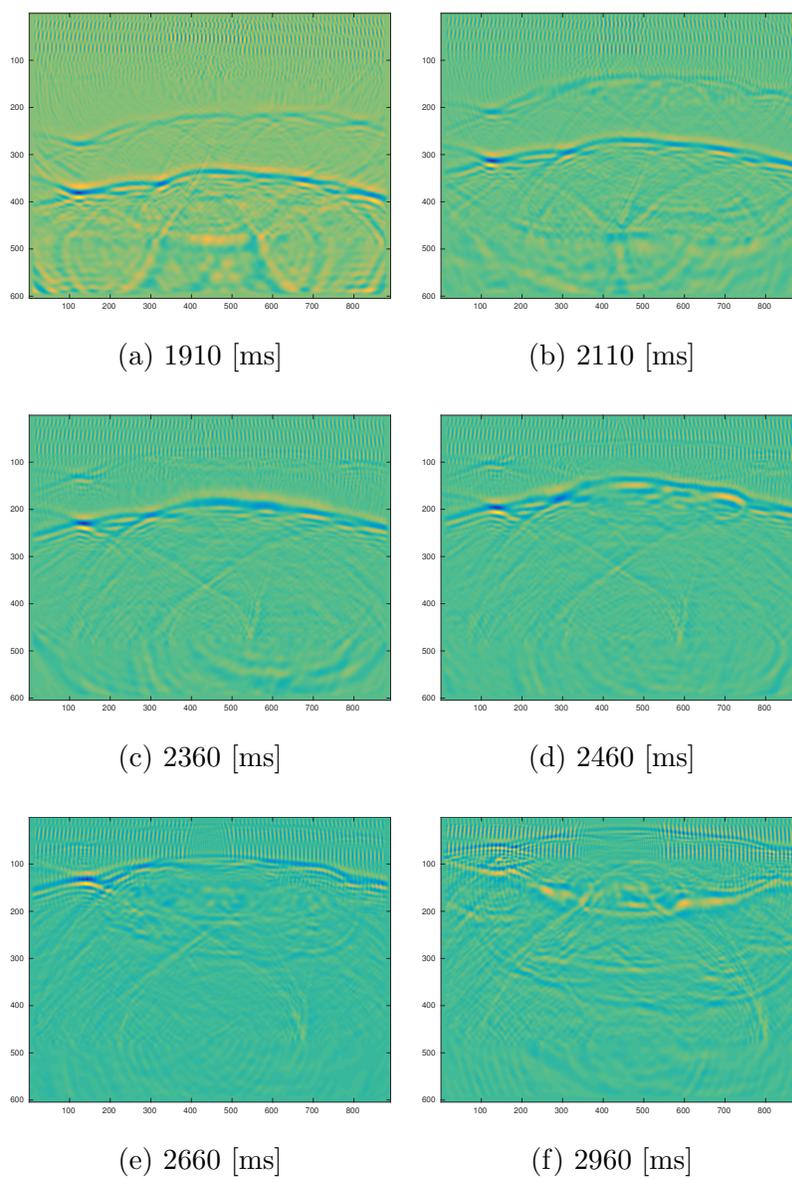


Figura 4.2: Final de la propagación de las ondas con las fronteras absorbentes

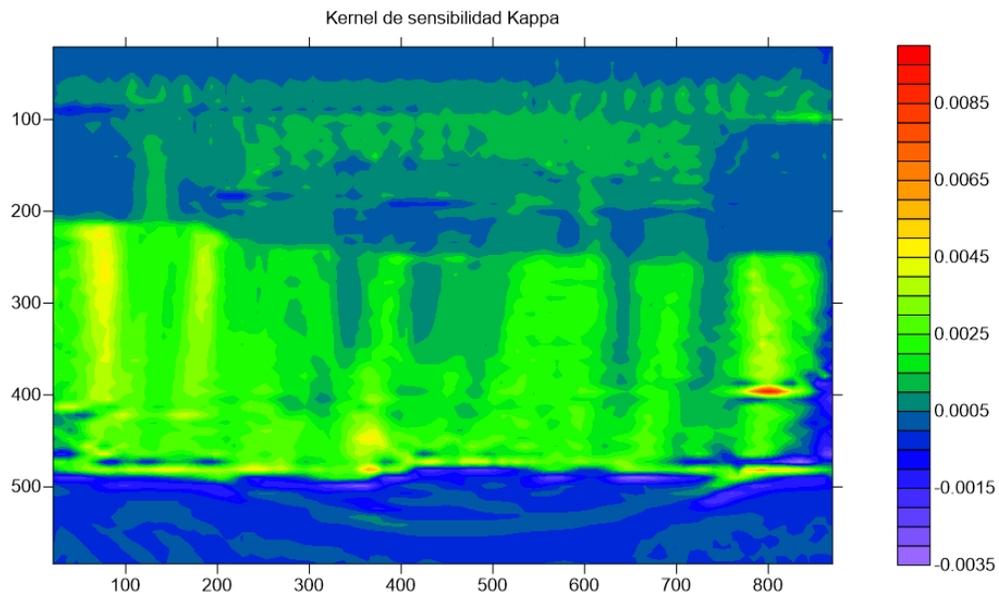


Figura 4.3: Kernel de sensibilidad Kappa obtenido de los datos del modelo de velocidad, figura 3.1 y densidad, figura 3.2

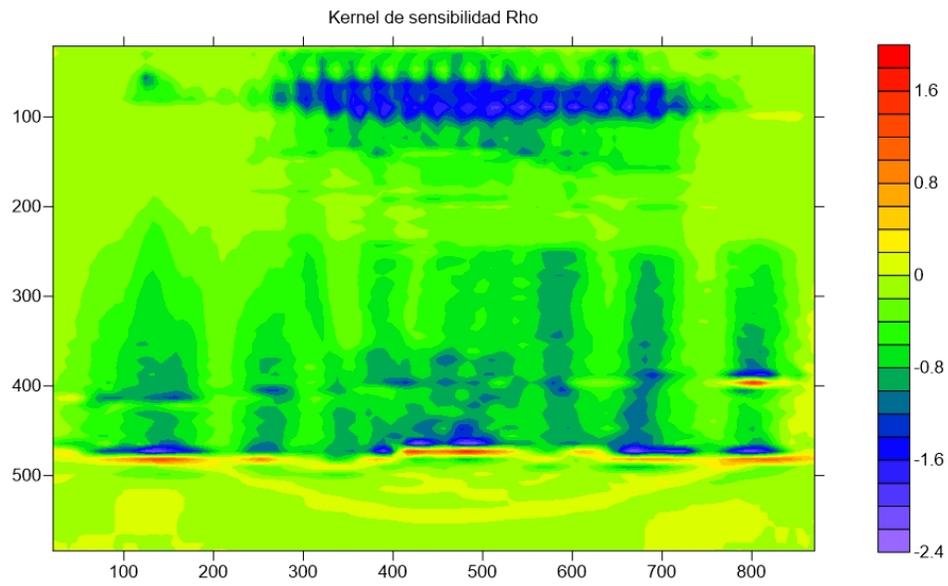


Figura 4.4: Kernel de sensibilidad Rho obtenido de los datos del modelo de velocidad, figura 3.1 y densidad, figura 3.2

4.2. Comparación de tiempos

Ahora se hace la comparación entre los tiempos de cómputo entre el programa en Fortran 95, Fortran MPI y CUDA C++, se aprecia una gran diferencia con respecto a los tiempos del programa en simple Fortran 95 y los paralelizados, como se puede apreciar en la tabla 4.1 el proceso más tardado es la retropropagación siendo el Fortran 95 el más lento, seguido por CUDA C++ y como el de menor tiempo el Fortran MPI. Se calculó que el programa en CUDA C++ es 22.43 veces más rápido que el de Fortran 95, y el Programa Fortran MPI es unas 41.16 veces más rápido. El Programa Fortran MPI es 1.8 veces más rápido que el de CUDA C++.

Proceso	Fortran 95	CUDA C++	Fortran MPI
CPML	7.5[s]	8.69E-05[s]	5.79E-05[s]
Propagación directa	245.648[s]	12.4677065[s]	7.333945[s]
Retropropagación	671.78[s]	28.752871[s]	15.13309[s]
Total	924.928[s]	41.2206644[s]	22.46709294[s]

Cuadro 4.1: Tabla de resultados de los tiempos de cómputo

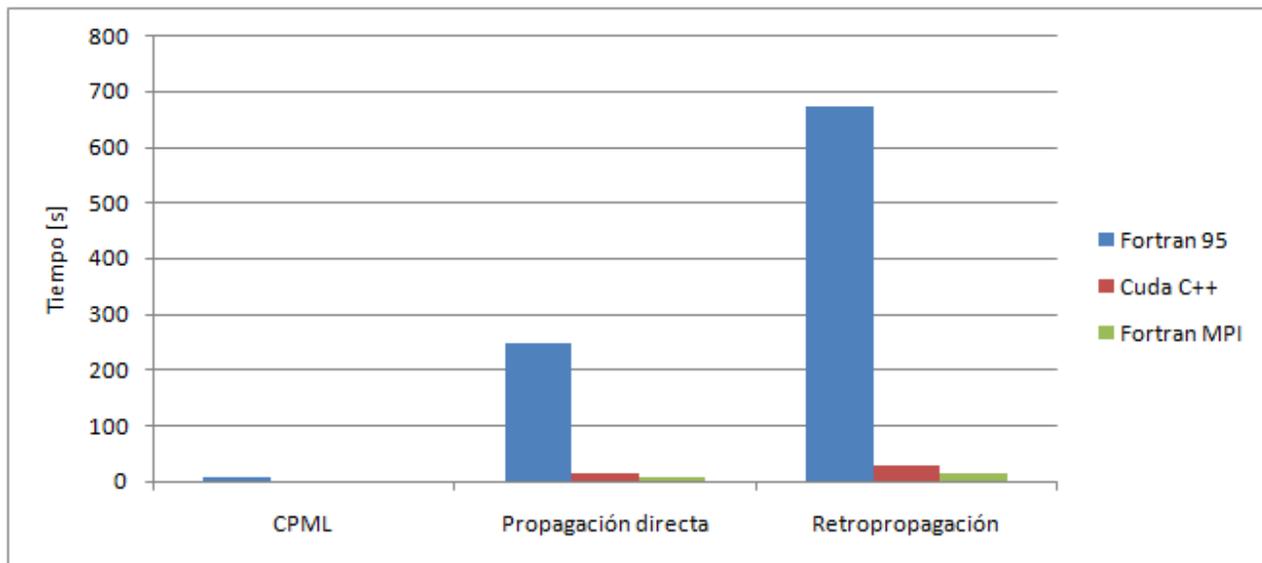


Figura 4.5: Gráfica de barras entre los tiempos de los programas Fortran 95, CUDA C++ y Fortran MPI

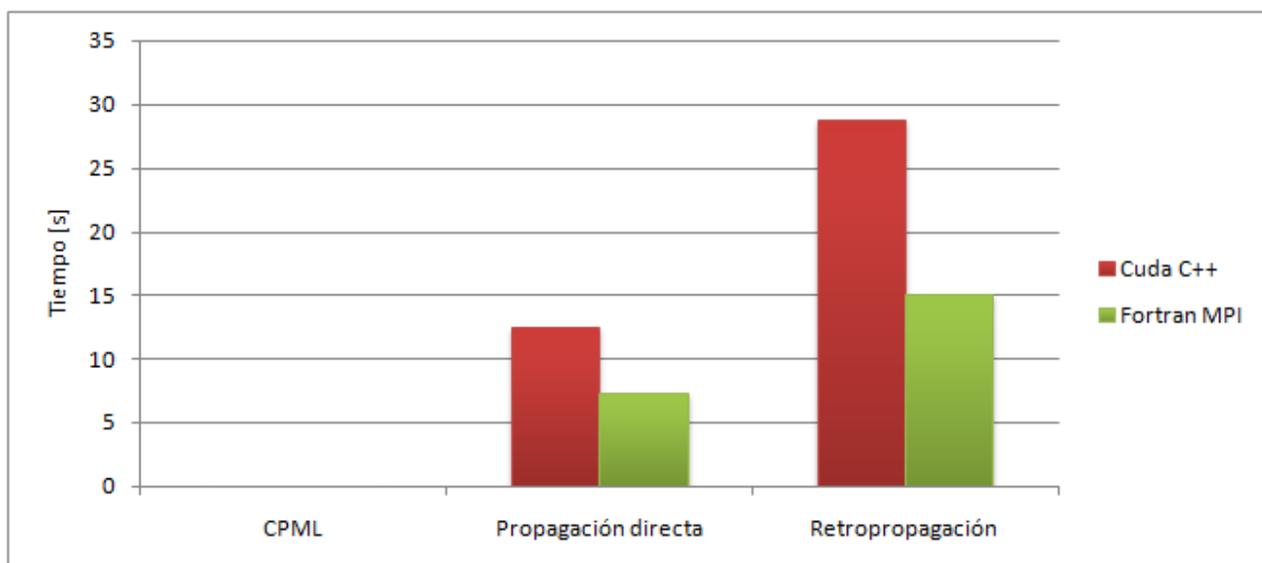


Figura 4.6: Gráfica de barras entre los tiempos de los programas CUDA C++ y Fortran MPI

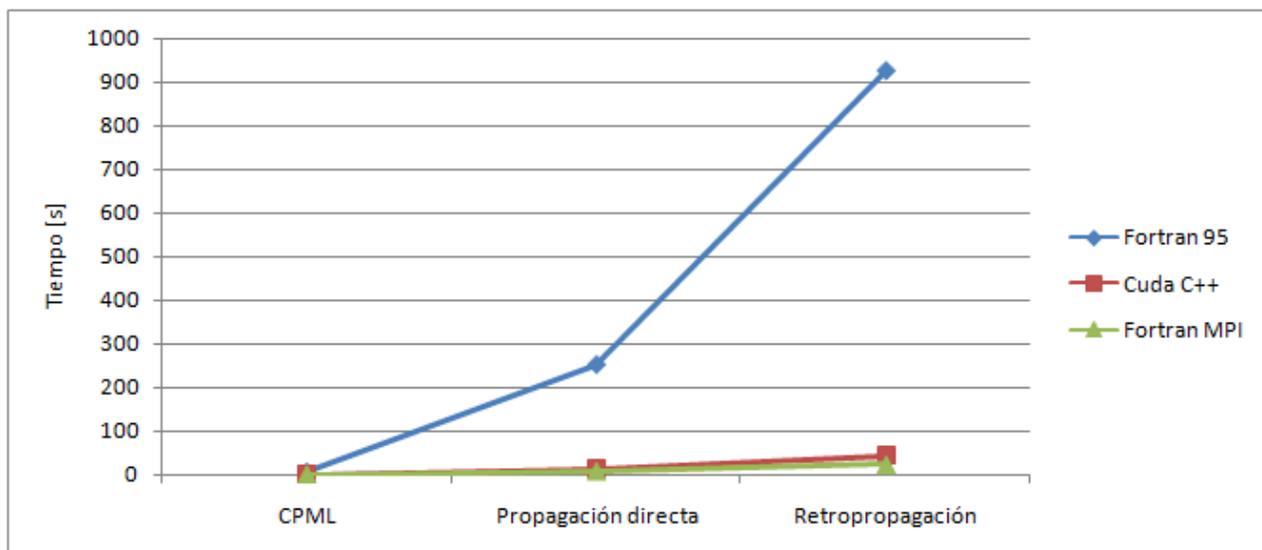


Figura 4.7: Gráfica lineal entre los tiempos de los programas Fortran 95, CUDA C++ y Fortran MPI

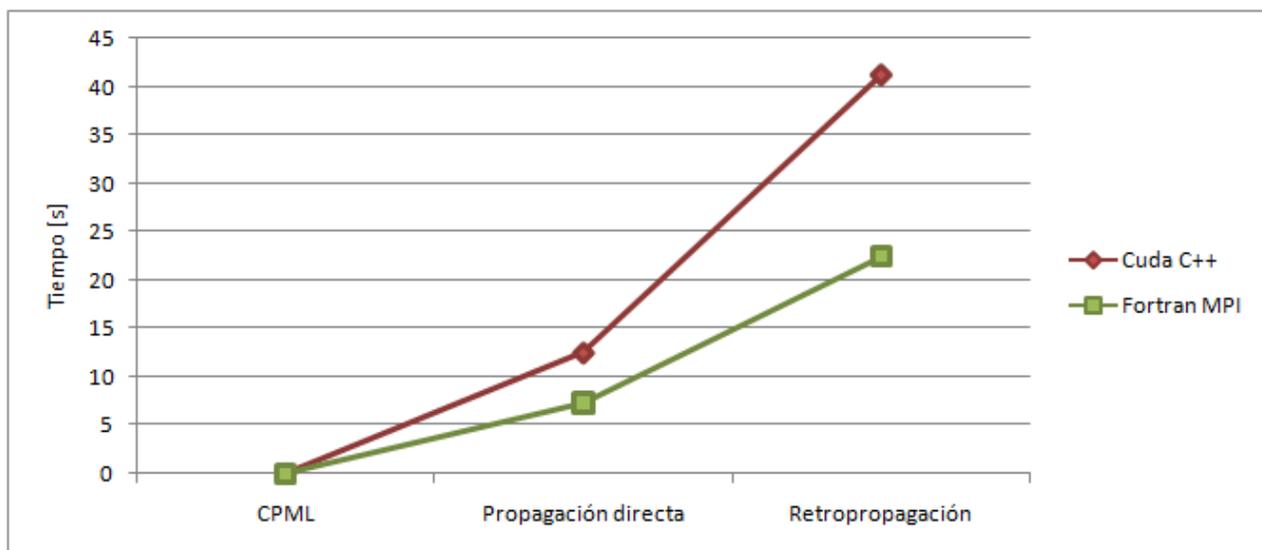


Figura 4.8: Gráfica lineal entre los tiempos de los programas CUDA C++ y Fortran MPI

Capítulo 5

Conclusión

Se pudo apreciar a través de este trabajo que existe más de una forma de crear un algoritmo de migración reversa en tiempo con barreras CPML que además opere con el menor gasto posible de tiempo de computo. El programa base sobre el que se trabajó fue escrito en Fortran 95 y aunque efectivo en sus resultados, la creación de kernels de sensibilidad y la propagación de la onda con el uso de fronteras CPML, es claramente deficiente en cuanto al tiempo que le toma obtener los resultados. Se crearon y compararon dos programas en lenguajes y ambientes diferentes para poder discernir cual sería el más óptimo para ser utilizado en un futuro, ambos tienen ventajas y desventajas asociadas a sus características. El programa en Fortran MPI es el más rápido, y muestra sus mejores cualidades al ser usado en un cluster ya que cada asigna la misma cantidad de memoria a sus procesos, en lugar de usar una memoria compartida, lo cual significa que sin acceso a un cluster no se puede aprovechar su potencial.

El programa en CUDA C++ sigue en velocidad, sin embargo pueden mejorarse los kernels para dar una operación más eficiente, sugiero que se trabaje más sobre este para lograr mayor eficiencia y usar un ambiente más poderoso que el usado para este trabajo.

Como recomendación final se propone el seguir adaptando el algoritmo para que este a la

par con el acelerado desarrollo computacional y probar otros lenguajes como el OpenCL ya que el algoritmo es solido y da una buena respuesta a la migración reversa en tiempo.

Referencias

- Aki, K. and Richards, P. (2002). *Quantitative Seismology*. Geology (University Science Books).: Seismology. University Science Books.
- Baysal, E., K. D. and Sherwood, J. (1983). Reverse time migration. *Geophysics*, 48(11):1514–1524.
- Berenger, J.-P. (1994). A perfectly matched layer for the absorption of electromagnetic waves. *Journal of computational physics*, 114(2):185–200.
- Berkhout, A. J. and Verschuur, D. J. (2003). Transformation of multiples into primary reflections. *73rd Annual International Meeting*, pages 1925–1928.
- Brodtkorb, A. R. (2010). *Scientific Computing on Heterogeneous Architectures*. PhD thesis, University of Oslo.
- Claerbout, J. F. (1968). Synthesis of a layered medium from its acoustic transmission response. *Geophysics*, 33(2):264–269.
- Collino, F. and Tsogka, C. (2001). Application of the perfectly matched absorbing layer model to the linear elastodynamic problem in anisotropic heterogeneous media. *Geophysics*, 66(1):294–307.

- Gedney, S. D. (1996). An anisotropic pml absorbing media for the fdtd simulation of fields in lossy and dispersive media. *Electromagnetics*, 16(4):399–415.
- Hidrobo, F. and Hoeger, H. (2005). Introducción a mpi (message passing interface).
- Komatitsch, D. and Martin, R. (2007). An unsplit convolutional perfectly matched layer improved at grazing incidence for the seismic wave equation. *Geophysics*, 72(5):SM155–SM167.
- Kuzuoglu, M. and Mittra, R. (1996). Mesh truncation by perfectly matched anisotropic absorbers in the finite-element method. *Microwave and Optical Technology Letters*, 12(3):136–140.
- Levin, S. A. (1984). Principle of reverse time migration. *Geophysics*, 49(5):581–583.
- McMechan, G. A. (1983). Seismic tomography in boreholes. *Geophysical Journal International*, 74(2):601–612.
- Michéa, D. and Komatitsch, D. (2010). Accelerating a three-dimensional finite-difference wave propagation code using gpu graphics cards. *Geophysical Journal International*, 182(1):389–402.
- Nocedal, J. and Wright, S. J. (2006). *Sequential quadratic programming*. Springer.
- R. Holdahl, H. H. and Lie, K.-A. (1999). Unconditionally stable splitting methods for the shallow water equations. *BIT Numerical Mathematics*, 39(3):451–472.
- Ratcliff, D. W., Gray, S. H., and Whitmore Jr, N. (1992). Seismic imaging of salt structures in the gulf of mexico. *The Leading Edge*, 11(4):15–31.
- Represa Pérez, C., Cámara Nebreda, J. M., Sánchez Ortega, P. L., et al. (2016). Introducción a la programación en cuda: v. 3.1.

- Roden, J. A. and Gedney, S. D. (2000). Convolution pml (cpml): An efficient fdtd implementation of the cfs-pml for arbitrary media. *Microwave and Optical Technology Letters*, 27(5):334–339.
- Shin, C. and Ha, W. (2008). A comparison between the behavior of objective functions for waveform inversion in the frequency and laplace domains. *Geophysics*, 73(5):VE119–VE133.
- Shin, C. and Min, D.-J. (2006). Waveform inversion using a logarithmic wavefield. *Geophysics*, 71(3):R31–R42.
- Symes, W. W. (2007). Reverse time migration with optimal checkpointing. *Geophysics*, 72(5):SM213–SM221.
- Tarantola, A. (1984). Inversion of seismic reflection data in the acoustic approximation. *Geophysics*, 49(8):1259–1266.
- Tarantola, A. (2005). *Inverse problem theory and methods for model parameter estimation*. SIAM.
- Tieman, H. J. (1995). Migration velocity analysis: Accounting for the effects of lateral velocity variations. *Geophysics*, 60(1):164–175.
- Tromp, J., Tape, C., and Liu, Q. (2005a). Seismic tomography, adjoint methods, time reversal and banana-doughnut kernels. *Geophysical Journal International*, 160(1):195–216.
- Tromp, J., Tape, C., and Liu, Q. (2005b). Seismic tomography, adjoint methods, time reversal and banana-doughnut kernels. *Geophysical Journal International*, 160(1):195–216.
- Van Trier, J. A. (1990). *Tomographic determination of structural velocities from depth-migrated seismic data*. PhD thesis, Stanford University.

-
- Virieux, J. (1986). Psv-wave propagation in heterogeneous media: velocity-stress finite difference method. *Geophysics*, 51:889–901.
- Weglein, A. B. (2014). Multiples: Signal and noise? *84th Annual International Meeting*, pages 4393–4399.
- Wu, R. S. and Aki, K. (1985). Elastic wave scattering by a random medium and the small-scale inhomogeneities in the lithosphere. *Journal of Geophysical Research: Solid Earth*, 90(B12):10261–10273.