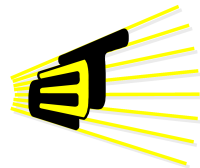


EVALUACIÓN DE ESTRATEGIAS DE IMPLEMENTACIÓN DE
UNA INVERSIÓN DE ONDA COMPLETA (FWI) 3D ACÚSTICA
CON DENSIDAD CONSTANTE EN EL DOMINIO TEMPORAL
SOBRE UNA ARQUITECTURA GPU

DAVID LEONARDO ABREO CARRILLO



Escuela de Ingenierías
Eléctrica, Electrónica y
de Telecomunicaciones



Universidad Industrial de Santander
Facultad de Ingenierías Fisicomecánicas
Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones
Bucaramanga
2017

EVALUACIÓN DE ESTRATEGIAS DE IMPLEMENTACIÓN DE
UNA INVERSIÓN DE ONDA COMPLETA (FWI) 3D ACÚSTICA
CON DENSIDAD CONSTANTE EN EL DOMINIO TEMPORAL
SOBRE UNA ARQUITECTURA GPU

Autor:
DAVID LEONARDO ABREO CARRILLO

Trabajo de investigación presentado como requisito para optar al título de
Magíster en Ingeniería Electrónica

Director:
MIE. Sergio Alberto Abreo Carrillo

Codirector:
PhD. Ana Beatriz Ramírez Silva

Universidad Industrial de Santander
Facultad de Ingenierías Fisicomecánicas
Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones
Bucaramanga
2017

Primeramente a Dios por bendecirme y protegerme cada día y permitirme terminar una fase más de mi vida.

A mis padres Roman Abreo y Clara Carrillo, por ser los pilares fundamentales en todo lo que soy, en toda mi educación, tanto académica, como de la vida, por su incondicional apoyo perfectamente mantenido a través del tiempo.

A Daniel Abreo, por ser mi hermanito y amigo todos estos años.

A Smith Tarazona, por estar a mi lado y ser mi soporte en los momentos más difíciles. Por su montón de paciencia conmigo y por su cariño y amor inmensurables.

Todo este trabajo es para ustedes.

AGRADECIMIENTOS

Este trabajo fue desarrollado con el apoyo de ECOPETROL y COLCIENCIAS como parte del proyecto de investigación No. 0266-2013.

Agradezco al cuerpo de docentes de la escuela de Ingeniería Eléctrica, Electrónica y de Telecomunicaciones de la Universidad Industrial de Santander y a todos aquellos participantes en mi formación profesional.

Un agradecimiento especial al grupo de investigación CPS y todos sus miembros. A mi Director y Codirector de Proyecto, los docentes Sergio Alberto Abreo y Ana Beatriz Ramírez, por su apoyo incondicional y académico en el desarrollo de este trabajo.

Quiero agradecer a mis padre y mi hermano por su perseverancia, por su fe y confianza en realizar esta meta; que con su dedicación, esfuerzo, amor y enseñanza me han dado las bases de formación humana y ética para continuar y fortalecer mi camino. A Smith por convertirse en parte de mi familia y ser uno de mis motivos para finalizar este trabajo.

Un agradecimiento a aquellas personas que me han apoyado incondicionalmente a lo largo de mi vida, brindándome su mano siempre en situaciones difíciles y sin los cuales no hubiese sido tan fácil llegar al sitio en el que estoy. A mis super amigos Fabián Sánchez y Fabián Noriega, que en medio de risas y largas charlas me han enseñado el verdadero valor de la amistad y de los cuales he aprendido mucho.

A todos, muchas Gracias!!!

David Leonardo Abreo Carrillo

CONTENIDO

	Pág.
1 INTRODUCCIÓN	14
2 ECUACIÓN DE ONDA	15
2.1. Ecuación de onda acústica con densidad constante en diferencias finitas	15
2.2. Condiciones de frontera	17
2.3. Dispersión y estabilidad numérica en la ecuación de onda acústica . . .	18
3 INVERSIÓN DE ONDA COMPLETA	24
3.1. Inversión de Onda Completa en el dominio del tiempo	25
3.1.1. Método del gradiente descendente	28
3.1.2. Método de L-BFGS	33
4 ESTRATEGIAS DE IMPLEMENTACIÓN	42
4.1. Implementación de referencia	42
4.2. Primera estrategia de implementación	43
4.3. Segunda estrategia de implementación	44
4.3.1. Reconstrucción de campo	46
4.4. División por disparos	47
5 RESULTADOS	51
5.1. Efectos de la dispersión numérica en la FWI 3D	51
5.2. FWI 3D en modelos complejos	53
5.2.1. Modelo sintético <i>overthrust</i>	53
5.2.2. Modelo sintético <i>Fuerte</i>	57
6 DISCUSIÓN Y CONCLUSIONES	61
A CAPAS CONVOLUCIONADAS PERFECTAMENTE ACOPLADAS	65
REFERENCIAS BIBLIOGRÁFICAS	70
BIBLIOGRAFÍA	73

LISTA DE FIGURAS

	Pág.
Figura 1. Representación gráfica FDTD.	17
Figura 2. Variables en coordenadas esféricas para un punto en el espacio. . .	21
Figura 3. Dispersión numérica de la ecuación en diferencias finitas para diferentes conjuntos de valores θ , φ y velocidad del medio.	21
Figura 4. Representación tridimensional para la dispersión numérica de la ecuación de onda en diferencias finitas para diferentes valores θ , φ y $\Delta h \times k$. .	22
Figura 5. Trazas sísmicas para dos diferentes conjuntos de parámetros de simulación	23
Figura 6. Mínimos locales dentro de la función de costo para un solo parámetro de inversión	25
Figura 7. Esquema general de una implementación FWI	27
Figura 8. Actualización del punto de partida usando el método del gradiente descendente.	29
Figura 9. Posición espacial de las fuentes en la superficie del modelo.	30
Figura 10. Resultados de la FWI 3D multiescala con una ondícula de frecuencia de 5 Hz	30
Figura 11. Resultados de la FWI 3D multiescala con una ondícula de frecuencia de 10 Hz	31
Figura 12. Resultados de la FWI 3D multiescala con una ondícula de frecuencia de 15 Hz	31
Figura 13. Perfiles de velocidad del modelo final obtenido en cada paso de frecuencia.	32
Figura 14. Modelo final de velocidad obtenido en la FWI 3D usando el método del gradiente descendente.	32
Figura 15. Resultados de la FWI 3D L-BFGS multiescala con una ondícula de Frecuencia de 10 Hz	35
Figura 16. Resultados de la FWI 3D L-BFGS multiescala con una ondícula de Frecuencia de 10 Hz	35
Figura 17. Resultados de la FWI 3D L-BFGS multiescala con una ondícula de Frecuencia de 15 Hz	36
Figura 18. Perfiles de velocidad del modelo final obtenido en cada paso de frecuencia.	36
Figura 19. Modelo final de velocidad obtenido en la FWI 3D usando el método de L-BFGS.	37

Figura 20. Corte interno modelo final obtenido por el método de gradiente descendente.	37
Figura 21. Corte interno modelo final obtenido por el método de L-BFGS. . .	38
Figura 22. Problema de salto de fase	39
Figura 23. Salto de fase presentado en un corte en dirección del eje y con una fuente situada en una esquina de la superficie de los modelos	40
Figura 24. Salto de fase presentado en un corte en dirección del eje y con una fuente situada en el centro de la superficie de los modelos	40
Figura 25. Comparación del salto de fase del modelo obtenido por gradiente descendente y L-BFGS.	41
Figura 26. Comparación del salto de fase del modelo obtenido por gradiente descendente y L-BFGS.	41
Figura 27. Cálculo del gradiente usando la implementación de referencia	43
Figura 28. Cálculo del gradiente guardando un solo campo	44
Figura 29. Cálculo del gradiente guardando la información de fronteras	45
Figura 30. Secciones del modelo donde se almacena la información de frontera.	47
Figura 31. Representación de los recursos físicos al usar MPI.	48
Figura 32. Esquema FWI usando división de disparos a través de MPI	49
Figura 33. Efectos de la dispersión numérica en el modelo final de velocidad . .	52
Figura 34. Sección 2D del modelo final de velocidad obtenido usando una dispersión numérica de 0,2% y 1%.	52
Figura 35. Perfiles de velocidad del modelo original y de los modelos obtenidos usando dos diferentes valores de dispersión numérica.	53
Figura 36. Posición espacial de las fuentes en la superficie del modelo.	54
Figura 37. Secciones centradas en dirección del eje x y del eje y del modelo <i>overthrust</i>	55
Figura 38. Desfase del conjunto del modelo inicial y modelo final en comparación al dato observado en <i>overthrust</i>	56
Figura 39. Comparación del salto de fase antes y después de la FWI 3D en <i>overthrust</i>	56
Figura 40. Perfiles de velocidad del modelo real, inicial y obtenido en <i>overthrust</i> .	57
Figura 41. Secciones centradas en dirección del eje x y del eje y del modelo <i>fuerte</i> .	58
Figura 42. Desfase del conjunto del modelo inicial y modelo final en comparación al dato observado en <i>fuerte</i>	59
Figura 43. Comparación del salto de fase antes y después de la FWI 3D en <i>fuerte</i> .	59
Figura 44. Perfiles de velocidad del modelo real, inicial y obtenido en <i>fuerte</i> . .	60
Figura 45. Vector F_q usado en CPML para la atenuación de las reflexiones no naturales para cada una de las direcciones de propagación.	66
Figura 46. Vector d_q usado en CPML para la atenuación de las reflexiones no naturales para cada una de las direcciones de propagación.	67
Figura 47. Valor de b_q usado en las variables auxiliares de las ecuaciones de CPML para la atenuación de las reflexiones no naturales para cada una de las direcciones de propagación.	67

Figura 48. Valor de a_q usado en las variables auxiliares de las ecuaciones de CPML para la atenuación de las reflexiones no naturales para cada una de las direcciones de propagación 68

LISTA DE TABLAS

Tabla 1.	Conjunto de parámetros para dos valores de dispersión diferente . .	23
Tabla 2.	Resultados del gradiente descendente contra L-BFGS	39
Tabla 3.	Tiempo de ejecución y memoria requerida para cada una de las es- trategias implementadas en múltiples GPUs	50
Tabla 4.	Dispersión numérica vs Memoria requerida	52
Tabla 5.	Memoria requerida en modelos de velocidad complejos usando la es- trategia 2 de implementación.	60

RESUMEN

TÍTULO: Evaluación de estrategias de Implementación de una Inversión de Onda Completa (FWI) 3D acústica con densidad constante en el dominio temporal sobre una arquitectura GPU*

AUTOR: David Leonardo Abreo Carrillo**

PALABRAS CLAVE: Inversión sísmica, FWI, GPU, FTDT, Propagación dominio temporal, 3D, MPI

DESCRIPCIÓN:

Uno de los principales objetivos de la exploración de hidrocarburos, es encontrar imágenes sísmicas confiables. A partir de la sísmica de reflexión es posible extraer información de profundidad del subsuelo mediante la inyección de energía en superficie que viaja por el subsuelo y vuelve a la superficie cuando se produce un cambio en las propiedades del medio.

La Inversión de Onda Completa (FWI) es un proceso iterativo que busca estimar las propiedades del subsuelo mediante la minimización de la diferencia entre los datos registrados y los datos modelados. Los tres principales aspectos que restringen la implementación de la FWI es la determinación de un punto de partida adecuado, el tiempo de ejecución del algoritmo y la memoria requerida por el método, donde los dos últimos aspectos son mucho más críticos en el caso 3D.

Asumiendo un buen punto de partida, este proyecto se enfoca en definir estrategias de implementación que reduzcan la memoria requerida y tiempo de cómputo de la FWI 3D. Se usa como implementación de referencia aquella donde se usa los dos campos de propagación para calcular el gradiente usando el FOASM planteado por Plessix. Se definen dos estrategias para la reducción de la memoria requerida por el gradiente de la FWI y la metodología de división de disparos en múltiples GPUs para disminuir el tiempo de ejecución usando MPI.

Los resultados muestran que la primera estrategia usa el mismo tiempo de ejecución que la implementación de referencia, pero con un 50 % de la memoria requerida por esta. En la segunda estrategia sólo se usa 30 % de la memoria de la primera estrategia pero esta última es un 35 % más rápido que la segunda estrategia. La división por disparos ofrece un comportamiento casi lineal en el factor de aceleración en proporción a la relación del número de disparos y las GPUs usadas.

* Trabajo de investigación.

** Facultad de Ingenierías Fisicomecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Maestría en Ingeniería Electrónica. Director: MIE. Sergio Alberto Abreo Carrillo.

ABSTRACT

TITLE: Evaluation of implementation strategies of acoustic 3D Full Waveform Inversion with constant density in time domain over a GPU architecture.*

AUTHOR: David Leonardo Abreo Carrillo**

KEYWORDS: Seismic Inversion, FWI, GPU, FTDT, Propagation in time domain, 3D, MPI

DESCRIPTION:

One of the main objectives of hydrocarbon exploration is to find reliable seismic images. From the seismic of reflection it is possible to extract depth information from the subsoil by injecting energy on the surface that travels through the subsoil and returns to the surface when a change in the properties of the medium occurs and is recorded by sensors called geophones.

The Full Wave Investment (FWI) is an iterative process that seeks to estimate the properties of the subsoil by minimizing the difference between the data recorded and the data modeled within the investment process. The three main aspects that restrict the implementation of the FWI is the determination of a suitable starting point, the execution time of the algorithm and the memory required by the method, where the last two aspects are much more critical in the 3D case.

Assuming a good starting point, this project focuses on defining implementation strategies that reduce the required memory and compute time of the FWI 3D. It is used as a reference implementation where the two propagation fields are used to calculate the gradient using the FOASM proposed by Plessix. Two strategies are defined for the reduction of the memory required by the gradient of the FWI and the methodology of division of shots in multiple GPUs to reduce the execution time using MPI.

The results show that the first strategy uses the same execution time as the reference implementation, but with 50 % of the memory required by it. In the second strategy only 30 % of the memory of the first strategy is used, but the latter is approximately 35 % faster than the second strategy. The cost in time is compensated with the reduction of the used memory. The division by shots offers an almost linear behavior in the acceleration factor in proportion to the ratio of the number of shots and the GPUs used for the implementation.

* Bachelor thesis.

** Physical-Mechanical Engineering Faculty. Electrical Engineering, Electronics and Telecommunications School. Master Degree in Electronic Engineering. Director: MIE. Sergio Alberto Abreo Carrillo.

1. INTRODUCCIÓN

Durante los últimos 20 años, la inversión de onda completa (FWI por sus siglas en inglés) ha estado en transición desde un enfoque defendido y usado casi exclusivamente por las instituciones académicas a ser un método habitual empleado en exploración sísmica en todo el mundo. Esta nueva técnica permite resolver zonas complejas con una gran resolución superando a técnicas como por ejemplo la tomografía basada en el trazado de rayos. [1].

Debido a que FWI resuelve un problema inverso mal puesto, diferentes modelos pueden entrar en una región de modelos igualmente válidos con un margen de tolerancia razonable. Este problema de no unicidad lo causan principalmente los mínimos locales de la función de ajuste entre los datos, donde estos corresponden a modelos de velocidad que hacen coincidir los desplazamientos de fase de los datos observados con los datos modelados (problema de salto de fase) [4].

Para mitigar este problema, se han propuesto los enfoques multiescala [5]. Estos métodos añaden de forma recursiva los detalles de mayor frecuencia a los primeros modelos calculados a partir de datos de baja frecuencia, dado que teóricamente, a bajas frecuencias, existe una alta probabilidad de que los datos modelados y observados coincidan dentro de medio ciclo de onda [3]. Se han propuesto otros métodos para abordar estos problemas, por ejemplo, invirtiendo los componentes de alto número de onda y de bajo número de onda por separado [6].

Para llevar la inversión de onda completa 3D en el dominio del tiempo a una implementación en una plataforma de cómputo paralelo como las GPUs, se requiere evaluar las limitantes que conlleva este proceso. En [7] se implementa la FWI 2D en el dominio temporal usando GPUs para reducir el costo computacional. En este trabajo se destaca que aunque las GPUs son eficientes en el procesamiento masivo de datos (ideal para procesamiento tridimensional), presentan dificultades para almacenar grandes cantidades de información.

De esta manera, es necesario plantear estrategias para poder implementar la inversión 3D en GPUs. Así, en este proyecto de investigación se muestran las estrategias implementadas, enfocadas en reducir la memoria requerida para la implementación de la FWI 3D, y el uso de un cluster multi-GPU para reducir el tiempo de computo de la inversión.

2. ECUACIÓN DE ONDA

En técnicas de análisis sísmico como la inversión de onda completa (FWI) o la migración reversa en el tiempo (MRT) es importante definir e implementar adecuadamente un método que permita la realización del modelado de la propagación de la onda de presión a través de un dominio temporal, frecuencial, espacial, entre otros [8]. Cada uno de estos métodos presenta ventajas y desventajas, adicionalmente requiere del análisis de la estabilidad y la dispersión numérica si se implementan en plataformas de cómputo.

2.1. Ecuación de onda acústica con densidad constante en diferencias finitas

Para realizar el modelado de la propagación de las ondas en el subsuelo, se usa la ecuación (2.1)

$$\frac{1}{\mathbf{m}^2(x, y, z)} \frac{\partial^2 \mathbf{p}(x, y, z, t)}{\partial t^2} = \frac{\partial^2 \mathbf{p}(x, y, z, t)}{\partial x^2} + \frac{\partial^2 \mathbf{p}(x, y, z, t)}{\partial y^2} + \frac{\partial^2 \mathbf{p}(x, y, z, t)}{\partial z^2} + \mathbf{s}(x, y, z, t), \quad (2.1)$$

la cuál, representa la ecuación de onda acústica en tres dimensiones con densidad constante para un medio isótropo, donde $\mathbf{m}(x, y, z)$ es la velocidad acústica en el medio, $\mathbf{p}(x, y, z, t)$ denota el campo de presión escalar; x , y y z son las variables espaciales; t es la variable temporal y \mathbf{s} es la fuente.

La ecuación (2.1) se implementa usando diferencias finitas centradas en el dominio temporal (FDTD), usando una aproximación de segundo orden para la derivada temporal, y una aproximación de octavo orden para las derivadas espaciales [9].

La aproximación de segundo orden para la derivada temporal se define como

$$\frac{\partial^2 \mathbf{p}(x, y, z, t)}{\partial t^2} \approx \frac{\mathbb{U}_{i,j,k}^{n+1} - 2 \cdot \mathbb{U}_{i,j,k}^n + \mathbb{U}_{i,j,k}^{n-1}}{\Delta t^2}, \quad (2.2)$$

donde $\mathbb{U}_{i,j,k}^n$ es el campo de presión discreto; i, j, k y n representa las variables discretas para x , y , z y t , respectivamente, y Δt es el paso temporal. Puede verse como una expresión en términos de los campos de presión pasado ($\mathbb{U}_{i,j,k}^{n-1}$), presente ($\mathbb{U}_{i,j,k}^n$) y futuro ($\mathbb{U}_{i,j,k}^{n+1}$).

La aproximación de octavo orden para cada una de las componentes x , y , y z se define como sigue:

$$\frac{\partial^2 \mathbf{p}(x, y, z, t)}{\partial x^2} \approx \frac{\sum_{c=-4}^4 C_s[c] \cdot \mathbb{U}_{i+c,j,k}^n}{\Delta x^2}, \quad (2.3)$$

$$\frac{\partial^2 \mathbf{p}(x, y, z, t)}{\partial y^2} \approx \frac{\sum_{c=-4}^4 C_s[c] \cdot \mathbb{U}_{i,j+c,k}^n}{\Delta y^2}, \quad (2.4)$$

$$\frac{\partial^2 \mathbf{p}(x, y, z, t)}{\partial z^2} \approx \frac{\sum_{c=-4}^4 C_s[c] \cdot \mathbb{U}_{i,j,k+c}^n}{\Delta z^2}, \quad (2.5)$$

donde C_s es un vector cuyas posiciones desde -4 hasta 4 están definidas como $C_s = [\frac{-1}{560}, \frac{8}{315}, \frac{-1}{5}, \frac{8}{5}, \frac{-205}{72}, \frac{8}{5}, \frac{-1}{5}, \frac{8}{315}, \frac{-1}{560}]$ y representa los coeficientes de la aproximación de octavo orden en diferencias finitas centradas donde Δx , Δy y Δz representan la resolución espacial del modelo.

Reemplazando las ecuaciones (2.2), (2.3), (2.4) y (2.5) en la ecuación (2.1) se tiene:

$$\begin{aligned} \frac{\mathbb{U}_{i,j,k}^{n+1} - 2 \cdot \mathbb{U}_{i,j,k}^n + \mathbb{U}_{i,j,k}^{n-1}}{(\mathbf{m}_{i,j,k})^2 \cdot \Delta t^2} &= \frac{\sum_{c=-4}^4 C_s[c] \cdot \mathbb{U}_{i+c,j,k}^n}{\Delta x^2} + \\ &\frac{\sum_{c=-4}^4 C_s[c] \cdot \mathbb{U}_{i,j+c,k}^n}{\Delta y^2} + \frac{\sum_{c=-4}^4 C_s[c] \cdot \mathbb{U}_{i,j,k+c}^n}{\Delta z^2}. \end{aligned} \quad (2.6)$$

Si se despeja el campo de presión futuro y se toma $\Delta x = \Delta y = \Delta z = \Delta h$, entonces se tiene que:

$$\begin{aligned} \mathbb{U}_{i,j,k}^{n+1} &= 2 \cdot \mathbb{U}_{i,j,k}^n + \frac{(\mathbf{m}_{i,j,k})^2 \cdot \Delta t^2}{\Delta h^2} \left(\sum_{c=-4}^4 C_s[c] \cdot \mathbb{U}_{i+c,j,k}^n + \right. \\ &\left. \sum_{c=-4}^4 C_s[c] \cdot \mathbb{U}_{i,j+c,k}^n + \sum_{c=-4}^4 C_s[c] \cdot \mathbb{U}_{i,j,k+c}^n \right) - \mathbb{U}_{i,j,k}^{n-1}; \end{aligned} \quad (2.7)$$

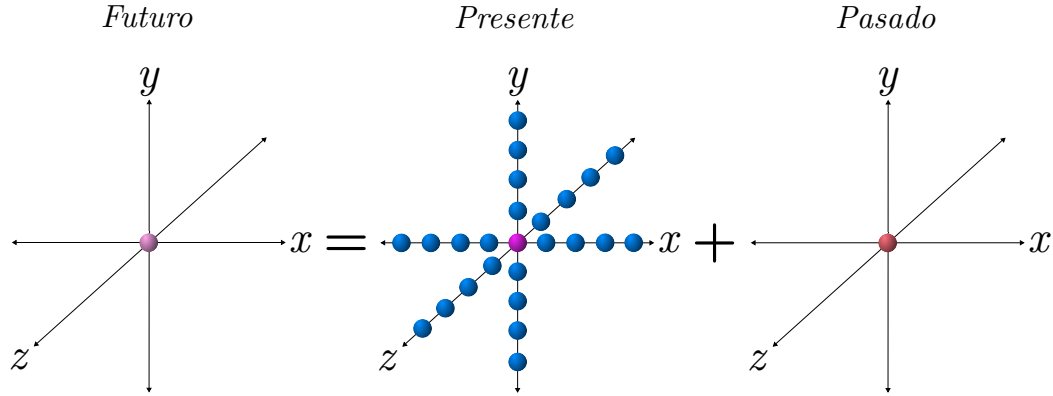
y factorizando:

$$\mathbb{U}_{i,j,k}^{n+1} = 2 \cdot \mathbb{U}_{i,j,k}^n + \frac{(\mathbf{m}_{i,j,k})^2 \cdot \Delta t^2}{\Delta h^2} \left(\sum_{c=-4}^4 C_s[c] \cdot (\mathbb{U}_{i+c,j,k}^n + \mathbb{U}_{i,j+c,k}^n + \mathbb{U}_{i,j,k+c}^n) \right) - \mathbb{U}_{i,j,k}^{n-1}. \quad (2.8)$$

Esta ecuación muestra que el campo de presión se va construyendo iterativamente, calculando el siguiente campo hacia adelante en el tiempo ($n+1$), usando la información de los campos de presión actual (n) y anterior ($n-1$), donde la iteración temporal está dada por el número n .

Una representación gráfica de esta ecuación para una iteración n , se muestra en la Figura 1. Se puede observar que un solo punto del futuro requiere la información de 25 puntos del campo de onda presente y un punto en el campo de onda pasado, los cuales se suponen conocidos.

Figura 1: Representación gráfica de la implementación del *stencil* de diferencias finitas en el dominio temporal. a) \mathbb{U}^{n+1} b) \mathbb{U}^n c) \mathbb{U}^{n-1}



2.2. Condiciones de frontera

Idealmente, en un medio que no tiene límites naturales en sus fronteras, la propagación de las ondas se extienden hasta el infinito. Este es un aspecto que se debe tener presente en la implementación la ecuación (2.8), dado que el modelo usado para realizar la propagación, es un modelo finito, lo que generará reflexiones no naturales dentro de la adquisición. Por lo tanto, se requiere de las condiciones de frontera para minimizar los reflejos artificiales en el interior del área de interés. Las condiciones de frontera se deben tener en cuenta en la solución de la ecuación de onda acústica utilizando alguna técnica de análisis numérico. Se han propuesto varios métodos para incluir estas condiciones en la solución de la ecuación de onda acústica, entre ellos los métodos de capas perfectamente acopladas (PML por sus siglas en inglés) y de capas convolucionadas perfectamente acopladas (CPML por sus siglas en inglés) [10]. La ecuación de onda acústica con densidad constante para un medio isótropo incluyendo CPML requiere de dos variables auxiliares ψ y ζ para atenuar las reflexiones que provienen de las fronteras no naturales. Modificando la ecuación (2.1) con base en lo propuesto por [10] se tiene que:

$$\frac{1}{\mathbf{m}^2(x, y, z)} \frac{\partial^2 \mathbf{p}(x, y, z, t)}{\partial t^2} = \frac{\partial^2 \mathbf{p}(x, y, z, t)}{\partial x^2} + \frac{\partial^2 \mathbf{p}(x, y, z, t)}{\partial y^2} + \frac{\partial^2 \mathbf{p}(x, y, z, t)}{\partial z^2} + \partial\psi(x, y, z) + \zeta(x, y, z) + \mathbf{s}(x, y, z, t), \quad (2.9)$$

donde :

$$\partial\psi(x, y, z) = \frac{\partial\psi_x(x, y, z)}{\partial x} + \frac{\partial\psi_y(x, y, z)}{\partial y} + \frac{\partial\psi_z(x, y, z)}{\partial z}, \quad (2.10)$$

$$\zeta(x, y, z) = \zeta_x(x, y, z) + \zeta_y(x, y, z) + \zeta_z(x, y, z). \quad (2.11)$$

Las variables auxiliares que minimizan las reflexiones están definidas en el Apéndice A. De igual manera se calcula el nuevo operador de onda usando FDTD.

La ecuación (2.9) es el nuevo operador de propagación que se usará para implementar la ecuación de onda en la plataforma de cómputo, y su expresión en diferencias finitas se formula con las ecuaciones acopladas (2.12), (2.13) y (2.14).

$$\psi_q = \begin{cases} \Psi_{\mathbf{x}(i,j,k)} = b_{x(i)} \Psi_{\mathbf{x}(i,j,k)} + a_{x(i)} \sum_{c=-4}^4 \left\{ \frac{1}{\Delta h} C_f[c] \cdot \mathbb{U}_{i+c,j,k}^n \right\}, \\ \Psi_{\mathbf{y}(i,j,k)} = b_{y(j)} \Psi_{\mathbf{y}(i,j,k)} + a_{y(j)} \sum_{c=-4}^4 \left\{ \frac{1}{\Delta h} C_f[c] \cdot \mathbb{U}_{i,j+c,k}^n \right\}, \\ \Psi_{\mathbf{z}(i,j,k)} = b_{z(k)} \Psi_{\mathbf{z}(i,j,k)} + a_{z(k)} \sum_{c=-4}^4 \left\{ \frac{1}{\Delta h} C_f[c] \cdot \mathbb{U}_{i,j,k+c}^n \right\}, \end{cases} \quad (2.12)$$

$$\zeta_q = \begin{cases} \mathbf{Z}_{\mathbf{x}(i,j,k)} = b_{x(i)} \mathbf{Z}_{\mathbf{x}(i,j,k)} + a_{x(i)} \sum_{c=-4}^4 \left\{ \frac{1}{\Delta h^2} C_s[c] \cdot \mathbb{U}_{i+c,j,k}^n + \frac{1}{\Delta h} C_f[c] \cdot \Psi_{\mathbf{x}(i+c,j,k)} \right\}, \\ \mathbf{Z}_{\mathbf{y}(i,j,k)} = b_{y(j)} \mathbf{Z}_{\mathbf{y}(i,j,k)} + a_{y(j)} \sum_{c=-4}^4 \left\{ \frac{1}{\Delta h^2} C_s[c] \cdot \mathbb{U}_{i,j+c,k}^n + \frac{1}{\Delta h} C_f[c] \cdot \Psi_{\mathbf{y}(i,j+c,k)} \right\}, \\ \mathbf{Z}_{\mathbf{z}(i,j,k)} = b_{z(k)} \mathbf{Z}_{\mathbf{z}(i,j,k)} + a_{z(k)} \sum_{c=-4}^4 \left\{ \frac{1}{\Delta h^2} C_s[c] \cdot \mathbb{U}_{i,j,k+c}^n + \frac{1}{\Delta h} C_f[c] \cdot \Psi_{\mathbf{z}(i,j,k+c)} \right\}, \end{cases} \quad (2.13)$$

$$\begin{aligned} \mathbb{U}_{i,j,k}^{n+1} = & 2 \cdot \mathbb{U}_{i,j,k}^n + \frac{(\mathbf{m}_{i,j,k} \cdot \Delta t)^2}{\Delta h^2} \left\{ \sum_{c=-4}^4 C_s[c] \cdot \langle \mathbb{U}_{i+c,j,k}^n + \mathbb{U}_{i,j+c,k}^n + \mathbb{U}_{i,j,k+c}^n \rangle \right\} \\ & + \frac{(\mathbf{m}_{i,j,k} \cdot \Delta t)^2}{\Delta h} \left\{ \sum_{c=-4}^4 C_f[c] \cdot \langle \Psi_{\mathbf{x}(i+c,j,k)} + \Psi_{\mathbf{x}(i,j+c,k)} + \Psi_{\mathbf{x}(i,j,k+c)} \rangle \right\} \\ & + (\mathbf{m}_{i,j,k} \cdot \Delta t)^2 \left\{ \mathbf{Z}_{\mathbf{x}(i,j,k)} + \mathbf{Z}_{\mathbf{y}(i,j,k)} + \mathbf{Z}_{\mathbf{z}(i,j,k)} \right\} - \mathbb{U}_{i,j,k}^{n-1} + \mathbb{S}_{i,j,k}^n, \end{aligned} \quad (2.14)$$

donde $\Psi_{\mathbf{q}(i,j,k)}$, $\mathbf{Z}_{\mathbf{q}(i,j,k)}$ son los volúmenes discretos de ψ_q y ζ_q . La variable C_f representa los coeficientes de la aproximación de octavo orden en diferencias finitas centradas para la derivada de primer orden, $C_f = [\frac{1}{280}, \frac{-4}{105}, \frac{1}{5}, \frac{-4}{5}, 0, \frac{4}{5}, \frac{-1}{5}, \frac{4}{105}, \frac{-1}{280}]$; Las posiciones del vector C_f van desde -4 hasta 4 .

2.3. Dispersión y estabilidad numérica en la ecuación de onda acústica

Al realizar la implementación de la ecuación de onda acústica en el dominio temporal usando una plataforma de cómputo, siempre estará presente la dispersión numérica en la solución de la ecuación. Para determinar el valor de dispersión numérica en una implementación, y a su vez, el conjunto de parámetros donde es estable la solución, se resuelve la ecuación de onda usando como fuente un frente de onda plano [11]. Se va a considerar una propagación de ondas sinusoidales a través de un espacio tridimensional de la siguiente manera:

$$\mathbb{U}_{l,m,p}^n = U_0 e^{i(nw\Delta t - lk_x\Delta x - mk_y\Delta y - pk_z\Delta z)}, \quad (2.15)$$

donde $i = \sqrt{-1}$, w es la frecuencia angular, k_x, k_y, k_z son los componentes del vector número de onda $\vec{\mathbf{k}}$ en las direcciones x, y y z ; n representa la iteración temporal y las posiciones discretas espaciales para x, y, z son respectivamente l, m, p .

La aproximación en diferencias finitas de segundo orden en tiempo y octavo orden en espacio de la ecuación de onda 3D es:

$$\mathbb{U}_{l,m,p}^{n+1} - 2 \cdot \mathbb{U}_{l,m,p}^n + \mathbb{U}_{l,m,p}^{n-1} = \left(\frac{\mathbf{m}^2 \Delta t^2}{\Delta h^2} \right) \cdot \sum_{c=-4}^4 C_s[c] \cdot (\mathbb{U}_{l+c,m,p}^n + \mathbb{U}_{l,m+c,p}^n + \mathbb{U}_{l,m,p+c}^n), \quad (2.16)$$

recordando que C_s representa los coeficientes de la aproximación de octavo orden en diferencias finitas centradas.

Reemplazando la ecuación (2.15) en la ecuación (2.16), en la parte izquierda de la igualdad tenemos lo expresado en la ecuación 2.17.

$$\mathbb{U}_{l,m,p}^{n+1} - 2 \cdot \mathbb{U}_{l,m,p}^n + \mathbb{U}_{l,m,p}^{n-1} = U_0 e^{i(nw\Delta t - lk_x \Delta x - mk_y \Delta y - pk_z \Delta z)} (e^{iw\Delta t} - 2 + e^{-iw\Delta t}). \quad (2.17)$$

De igual manera, en la parte derecha de la igualdad se obtiene:

$$\begin{aligned} \sum_{c=-4}^4 C_s[c] \cdot (\mathbb{U}_{l+c,m,p}^n + \mathbb{U}_{l,m+c,p}^n + \mathbb{U}_{l,m,p+c}^n) = \\ U_0 e^{i(nw\Delta t - lk_x \Delta x - mk_y \Delta y - pk_z \Delta z)} \sum_{c=-4}^4 (C_s[c] \cdot (e^{cik_x \Delta x} + e^{cik_y \Delta y} + e^{cik_z \Delta z})). \end{aligned} \quad (2.18)$$

Simplificando $U_0 e^{i(nw\Delta t - lk_x \Delta x - mk_y \Delta y - pk_z \Delta z)}$ en las ecuaciones (2.17) y (2.18), tomando $\Delta x = \Delta y = \Delta z = \Delta h$; la ecuación (2.16) se reescribe como:

$$(e^{iw\Delta t} - 2 + e^{-iw\Delta t}) = \left(\frac{\mathbf{m}^2 \Delta t^2}{\Delta h^2} \right) \cdot \sum_{c=-4}^4 [C_s[c] \cdot (e^{cik_x \Delta h} + e^{cik_y \Delta h} + e^{cik_z \Delta h})]. \quad (2.19)$$

Recordando que $e^{ix} + e^{-ix} = 2 \cdot \cos(x)$, y expandiendo la sumatoria, la ecuación (2.19) puede ser escrita como

$$\begin{aligned} (2\cos(w\Delta t) - 2) = \left(\frac{\mathbf{m}^2 \Delta t^2}{\Delta h^2} \right) \cdot (3 \cdot C_s[0] + \sum_{c=1}^4 (2 \cdot C_s[c] \cdot \\ (\cos(ck_x \Delta h) + \cos(ck_y \Delta h) + \cos(ck_z \Delta h))))). \end{aligned} \quad (2.20)$$

Ahora recordando que $\cos(2\theta) = 1 - 2\sin^2(\theta)$, se reescribe la ecuación (2.20) como:

$$\begin{aligned} -4\sin^2(w\Delta t/2) = \left(\frac{\mathbf{m}^2 \Delta t^2}{\Delta h^2} \right) \cdot \left(-3 \cdot \frac{205}{72} + \sum_{c=1}^4 (2 \cdot C_s[c] \cdot \right. \\ \left. ((1 - 2\sin^2(ck_x \Delta h/2)) + (1 - 2\sin^2(ck_y \Delta h/2)) + (1 - 2\sin^2(ck_z \Delta h/2))) \right). \end{aligned} \quad (2.21)$$

Reduciendo algunos términos se obtiene:

$$-4\sin^2(w\Delta t/2) = \left(\frac{\mathbf{m}^2\Delta t^2}{\Delta h^2} \right) \cdot (-4) \sum_{c=1}^4 (C_s[c] \cdot (\sin^2(ck_x\Delta h/2) + \sin^2(ck_y\Delta h/2) + \sin^2(ck_z\Delta h/2))) , \quad (2.22)$$

tomando $\alpha = \frac{\mathbf{m}\Delta t}{\Delta h}$ y despejando w de la ecuación (2.22) se tiene:

$$w = \frac{2}{\Delta t} \sin^{-1} \left(\alpha \sqrt{\sum_{c=1}^4 (C_s[c] (\sin^2(ck_x\Delta h/2) + \sin^2(ck_y\Delta h/2) + \sin^2(ck_z\Delta h/2)))} \right) \quad (2.23)$$

Si se usa la representación en coordenadas polares de $\vec{\mathbf{k}}$, entonces $k_x = k\cos(\theta)\cos(\varphi)$, $k_y = k\cos(\theta)\sin(\varphi)$, $k_z = k\sin(\theta)$ y $k = \sqrt{k_x^2 + k_y^2 + k_z^2}$. Recordando que $V_p = w/k$, se reescribe la ecuación (2.23) como:

$$\frac{V_p}{\mathbf{m}} = \frac{2}{\alpha\Delta h \cdot k} \sin^{-1} \left(\alpha \sqrt{\sum_{c=1}^4 \left(C_s[c] \left(\sin^2 \left(c \cdot k \frac{\Delta h}{2} \cos(\theta)\cos(\varphi) \right) + \sin^2 \left(c \cdot k \frac{\Delta h}{2} \cos(\theta)\sin(\varphi) \right) + \sin^2 \left(c \cdot k \frac{\Delta h}{2} \sin(\theta) \right) \right) \right) , \quad (2.24)$$

donde θ y φ son los ángulos esféricos en las coordenadas polares (latitud y azimut en la Figura 2).

A partir de la ecuación (2.24), la estabilidad numérica se mantiene en la propagación del campo de onda mientras $\alpha \leq 0,45$, es decir,

$$\frac{\mathbf{m}\Delta t}{\Delta h} \leq 0,45 . \quad (2.25)$$

Por otra parte, la dispersión numérica se mantiene al mínimo cuando $V_p/\mathbf{m} = 1$. Haciendo un barrido para los valores de Δh , Δt , θ y φ , se obtienen diferentes curvas que relacionan los parámetros $\frac{V_p}{\mathbf{m}}$ y $\Delta h \times k$ (Figura 3). Las líneas rojas representan diferentes valores de α en función de los ángulos θ y φ con el valor de máxima velocidad de \mathbf{m} y las líneas azules con el valor de velocidad mínima. Dependiendo de Δh , se puede mover el conjunto de parámetros en el eje horizontal de izquierda a derecha, donde la dispersión numérica solo incrementa en el lado derecho, porque el rango es mayor. En la Figura 3 se muestra las curvas de dispersión numérica para diferentes conjuntos de parámetros de modelado en un plano bidimensional.

Figura 2: Variables asociadas con el punto A cuando se representa en coordenadas esféricas en un plano cartesiano.

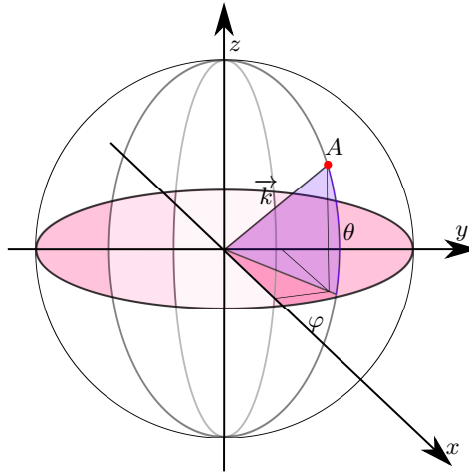
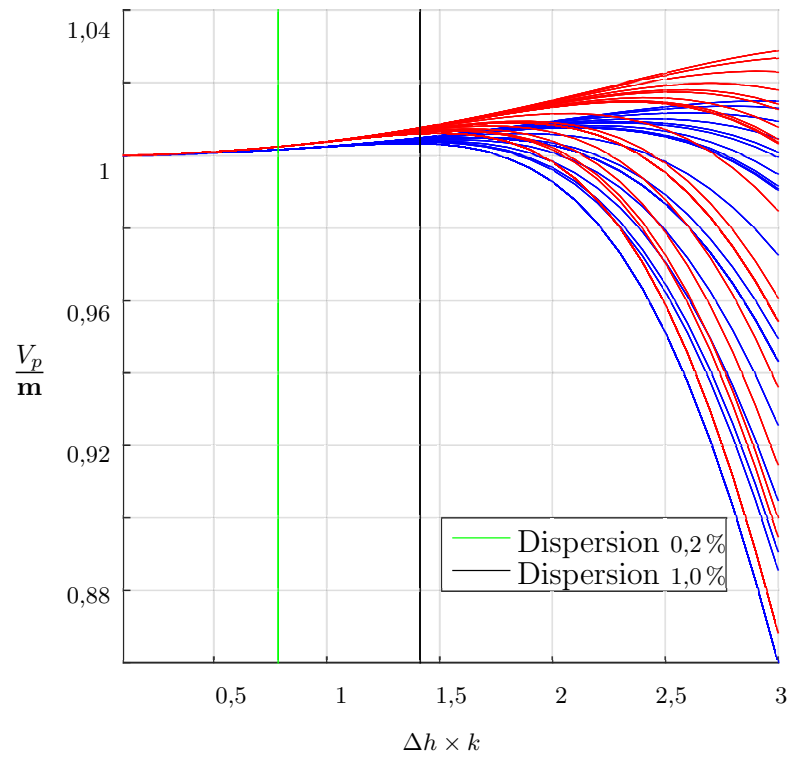
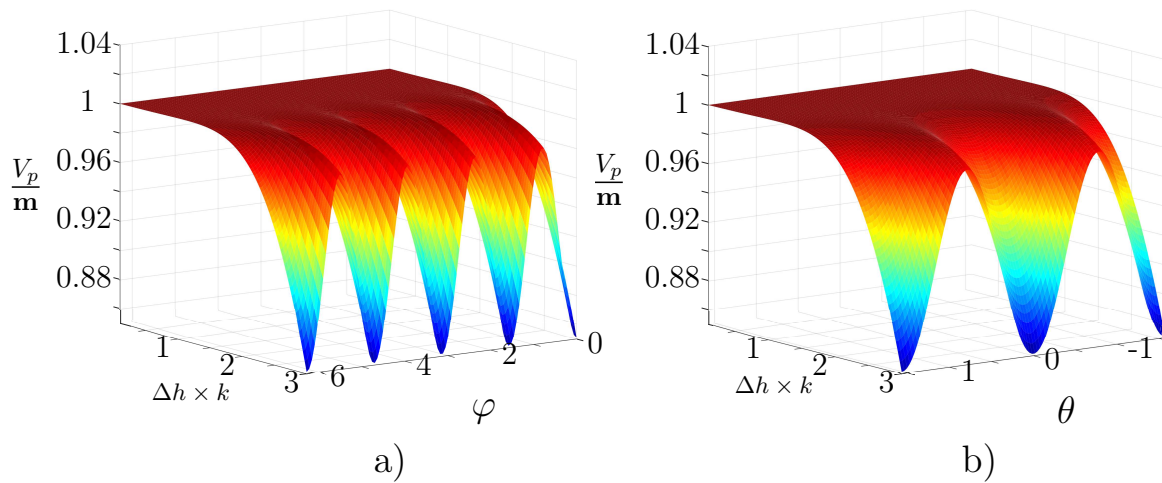


Figura 3: Dispersión numérica de la ecuación en diferencias finitas para diferentes conjuntos de valores θ , φ y velocidad del medio.



En la Figura 4 se muestra la representación tridimensional de la dispersión numérica realizando el análisis sobre un plano paralelo a la superficie (Figura 4-a) es decir, con $\theta = 0$ y $\varphi \in [0, 2\pi)$ y cuando se analiza en un plano perpendicular a la superficie y al eje y (Figura 4-b) es decir, con $\varphi = 0$ y $\theta \in [-\pi/2, \pi/2]$. Se toma la velocidad máxima del modelo para realizar estas gráficas.

Figura 4: Representación tridimensional para la dispersión numérica de la ecuación de onda para el análisis en (a) un plano paralelo la superficie de la propagación de la onda variando los valores de φ y para el análisis en (b) un plano paralelo al plano XZ , variando los valores de θ .



Se realiza la propagación de una fuente puntual centrada en superficie, usando un modelo sintético de velocidad de dos capas, con velocidades de 2000 m/s y 4000 m/s separadas por un reflector paralelo a la superficie, ubicado en la mitad del modelo. Se escogen dos conjuntos de parámetros diferentes, que presentan valores de dispersión numérica de 0,2% y 1,0%. La fuente usada para la propagación es definida en la ecuación (2.26).

$$s(t) = (1 - 2(t - t_0)^2 f^2 \pi^2) e^{-2(t-t_0)^2 f^2 \pi^2}, \quad (2.26)$$

donde t_0 es el corrimiento en tiempo de la ondícula y f es la frecuencia central del espectro de la fuente.

El primer conjunto de parámetros tiene dimensiones de $N_x=211$, $N_y=211$, $N_z=101$ y un paso espacial de 25 m, con 2,16 segundos de grabación con un paso temporal de 1,8 ms. El segundo conjunto de parámetros se muestra con un paso espacial de 45m, obteniendo dimensiones espaciales de $N_x=131$, $N_y=131$, $N_z=63$; el tiempo de grabación se mantiene, pero usando un paso temporal de 2,7 ms. Se usa una fuente en superficie con una frecuencia de 7 Hz.

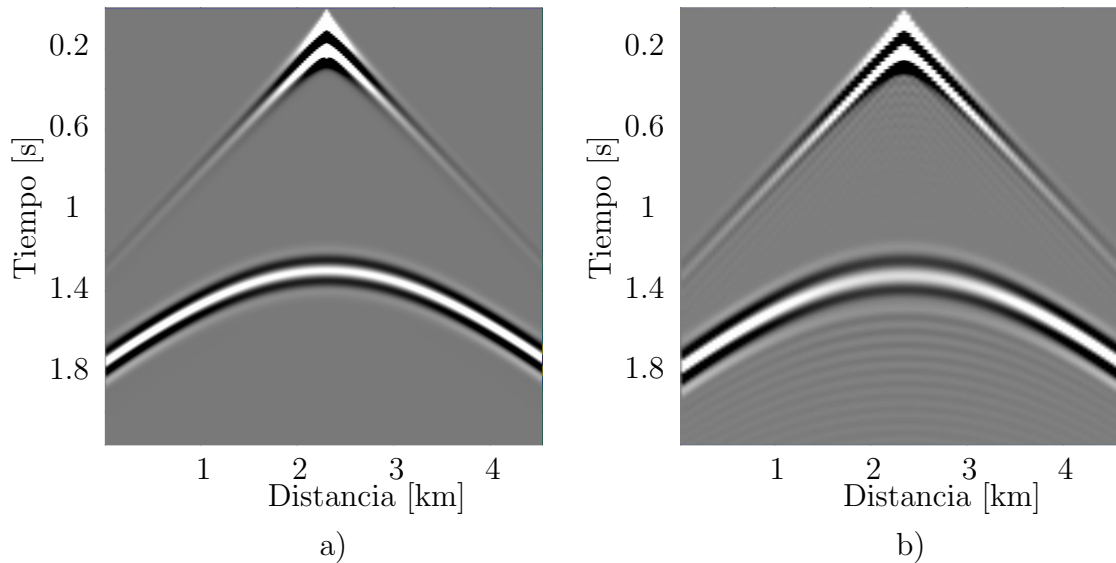
Los parámetros escogidos, la memoria requerida y el tiempo de cómputo se muestra en la Tabla 1.

Tabla 1: Conjunto de parámetros para dos valores de dispersión diferente

	Nx	Ny	Nz	Nt	dh	dt	Memoria	Tiempo	Dispersión
Parámetros 1	211	211	101	1200	25m	1,8ms	424MiB	75,5s	0,2%
Parámetros 2	131	131	63	800	45m	2,7ms	128MiB	9,2s	1,0%

Los resultados se observan en la Figura 5, donde se muestran dos cortes en el eje x del conjunto de trazas agrupadas por disparo común, donde la figura de la izquierda representa una dispersión numérica de 0,2% y la figura de la derecha una dispersión numérica de 1,0%.

Figura 5: Trazas sísmicas para dos diferentes conjuntos de parámetros de simulación que producen una dispersión numérica de a) 0,2% y b) 1% .



La traza obtenida de la propagación de onda en el medio de dos capas se ve afectada por la dispersión numérica, notándose una estela de reflexiones que no son propias de la propagación, generadas cuando los parámetros de propagación sólo se seleccionan para mantener la estabilidad numérica y ahorrar memoria. De esta manera, es necesario encontrar un equilibrio entre los requisitos de RAM y la dispersión numérica, dado que una traza que presente un valor de dispersión alto, se puede reflejar en el resultado de la inversión en eventos que no existen dentro del modelado de la propagación, generando resultados erróneos en el proceso de inversión.

3. INVERSIÓN DE ONDA COMPLETA

Durante los últimos 20 años, la inversión de onda completa (FWI por sus siglas en inglés) ha estado en transición desde un enfoque defendido y usado casi exclusivamente por las instituciones académicas a ser un método habitual empleado en exploración sísmica en todo el mundo. A medida que se logra entender y aplicar las inversiones complejas no lineales que se han desarrollado, y se tienen equipos de cómputo más potentes para realizar estas operaciones, la inversión de onda completa se ha llevado a la práctica usando menos aproximaciones. Esta nueva técnica permite resolver zonas complejas con una gran resolución superando a técnicas como por ejemplo la tomografía basada en el trazado de rayos. [1].

Desde que el algoritmo FWI iterativo basado en gradiente se introdujo por primera vez a la comunidad geofísica por Lailly [2], se han desarrollado muchas estrategias y esquemas computacionales, trabajando en el dominio del tiempo y de la frecuencia.

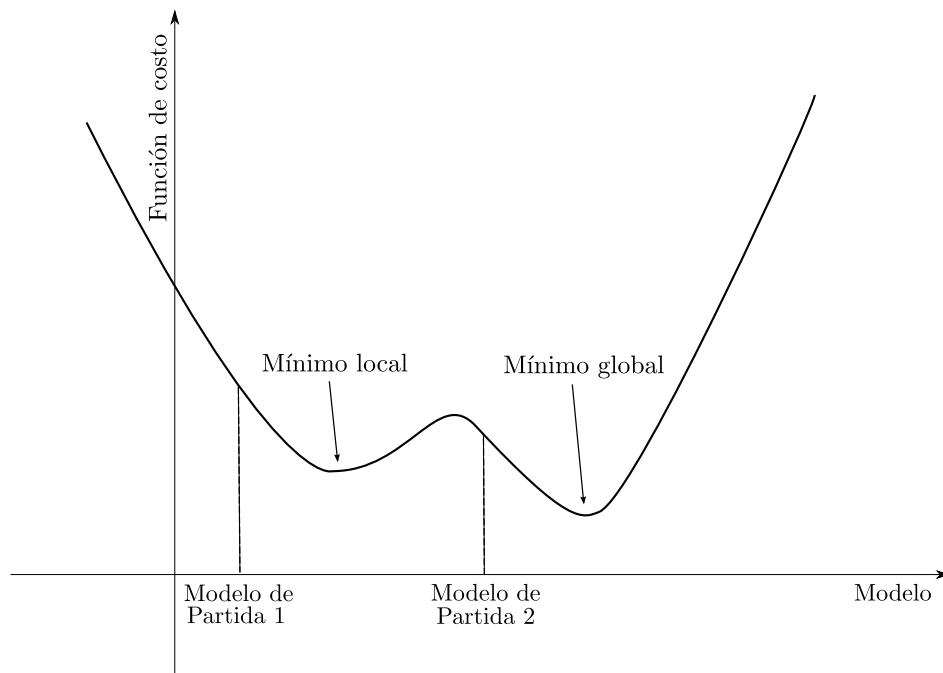
La inversión de onda completa es un proceso iterativo que estima las propiedades del subsuelo a partir de un proceso que busca minimizar el desajuste entre los datos registrados en una adquisición sísmica y los datos sintéticos modelados. Para que la inversión de onda completa funcione correctamente es necesario que el máximo desfase en tiempo entre los datos observados y modelados no supere más de la mitad del periodo de la ondícula central. [3].

Debido a que FWI resuelve un problema inverso mal puesto, diferentes modelos pueden entrar en una región de modelos igualmente válidos con un margen de tolerancia razonable. Este problema de no unicidad lo causan principalmente los mínimos locales de la función de ajuste entre los datos, donde estos corresponden a modelos de velocidad que hacen coincidir los desplazamientos de fase de los datos observados con los datos modelados (problema de salto de fase) [4].

La función de ajuste del modelo, es una función n -dimensional, con una dimensión por cada parámetro que se desee ajustar. Analizando un solo parámetro de ajuste del modelo del subsuelo, la Figura 6 muestra como se vería la función de costo y los resultados en función del punto de partida elegido. El punto de partida se obtiene por procesos de tomografía, empleando información *a priori* de la zona o por métodos de optimización global [12] [13].

Para mitigar este problema, se han propuesto los enfoques multiescala [5]. Estos métodos añaden de forma recursiva los detalles de mayor frecuencia a los primeros modelos calculados a partir de datos de baja frecuencia, dado que teóricamente, a bajas frecuencias, existe una alta probabilidad de que los datos modelados y observados

Figura 6: Mínimos locales dentro de la función de costo para un solo parámetro de inversión [14].



coincidan dentro de medio ciclo de onda [3]. La fidelidad de las técnicas multiescala depende fundamentalmente de la fidelidad del contenido de baja frecuencia en los datos registrados. Se han propuesto otros métodos para abordar estos problemas, por ejemplo, invirtiendo los componentes de alto número de onda y de bajo número de onda por separado [6].

3.1. Inversión de Onda Completa en el dominio del tiempo

El método FWI requiere una función de costo $\Phi(\hat{\mathbf{m}})$, una ecuación de onda para encontrar el campo de presión propagado hacia adelante y hacia atrás y un método de actualización para estimar un modelo sísmico de velocidad. La función de costo $\Phi(\hat{\mathbf{m}})$ es un escalar que mide el error entre los datos registrados por los geófonos en superficie d_{obs} y los datos modelados d_{mod} [14]. La FWI usa la norma ℓ_2 de la diferencia como función de costo, la cual está dado por

$$\Phi(\hat{\mathbf{m}}) = \arg \min_{\hat{\mathbf{m}}} \|d_{mod}(\hat{\mathbf{m}}) - d_{obs}(\mathbf{m})\|_2^2, \quad (3.1)$$

donde los datos modelados d_{mod} se calculan usando la ecuación (2.9).

La minimización de la función de costo (ecuación (3.1)) se logra de manera iterativa,

donde la actualización para $\hat{\mathbf{m}}$ se obtiene como:

$$\hat{\mathbf{m}}^{k+1} = \hat{\mathbf{m}}^k + \alpha_k \delta \hat{\mathbf{m}}, \quad (3.2)$$

donde k denota la iteración del proceso de inversión y $\delta \hat{\mathbf{m}}$ se calcula empleando la formulación del método de Newton [15]. El método de Newton aproxima la función de costo mediante la serie de Taylor hasta segundo orden [16], como sigue:

$$\Phi(\hat{\mathbf{m}}^k + \delta \hat{\mathbf{m}}) = \Phi(\hat{\mathbf{m}}^k) + (\delta \hat{\mathbf{m}}^k)^T \mathbf{g}(\hat{\mathbf{m}}^k) + \frac{1}{2} (\delta \hat{\mathbf{m}}^k)^T \mathbf{H}(\hat{\mathbf{m}}^k) \delta \hat{\mathbf{m}} + \mathbf{O}(\delta \hat{\mathbf{m}}^3), \quad (3.3)$$

donde $\mathbf{g}(\hat{\mathbf{m}}^k) = \frac{\partial \Phi(\hat{\mathbf{m}}^k)}{\partial \hat{\mathbf{m}}}$ es el gradiente de la función de costo evaluado en el punto $\hat{\mathbf{m}}^k$, $\mathbf{H}(\hat{\mathbf{m}}^k) = \frac{\partial^2 \Phi(\hat{\mathbf{m}}^k)}{\partial \hat{\mathbf{m}}^2}$ es la Hessiana de la función de costo evaluada en el punto $\hat{\mathbf{m}}^k$, y $\mathbf{O}(\delta \hat{\mathbf{m}}^3)$ representa el error asociado a los términos de orden superior de la expresión de Taylor.

Finalmente, se puede encontrar que la dirección de actualización en el método de Newton es [16]:

$$\delta \hat{\mathbf{m}} = - (\mathbf{H}(\hat{\mathbf{m}}^k))^{-1} \mathbf{g}(\hat{\mathbf{m}}^k), \quad (3.4)$$

y la actualización del modelo de velocidad está dada por:

$$\hat{\mathbf{m}}^{k+1} = \hat{\mathbf{m}}^k + \alpha_k (- (\mathbf{H}(\hat{\mathbf{m}}^k))^{-1} \mathbf{g}(\hat{\mathbf{m}}^k)), \quad (3.5)$$

donde α_k es un escalar que determina el paso de avance hacia el mínimo de la función de costo [17]. La inversa de $\mathbf{H}(\hat{\mathbf{m}}^k)$ está relacionada con la información de curvatura de la función de costo y $\mathbf{g}(\hat{\mathbf{m}}^k)$ está relacionado con la dirección de actualización del modelo. La selección del valor de α_k se convierte en un problema de ajuste mientras que el cálculo de $((\mathbf{H}(\hat{\mathbf{m}}^k))^{-1} \mathbf{g}(\hat{\mathbf{m}}^k))$ es un proceso costoso computacionalmente para el cual se plantean diversas estrategias para determinarlo.

El gradiente puede ser calculado eficientemente empleando el método del estado adjunto planteado por Plessix [18]. Plessix define el gradiente de la función de costo como:

$$\mathbf{g}(\hat{\mathbf{m}}^k) = - \frac{2}{(\hat{\mathbf{m}}^k)^3} \int_0^T \lambda \cdot \frac{\partial^2 \mathbf{p}(x, y, z, t)}{\partial t^2} dt, \quad (3.6)$$

donde T es el tiempo de grabación de los datos registrados, $\mathbf{p}(x, y, z, t)$ denota el campo de presión propagado hacia adelante y λ representa el campo de presión generado por la propagación hacia atrás de los datos residuales d_{res} ; donde d_{res} se obtiene de $d_{mod}(\hat{\mathbf{m}}) - d_{obs}(\mathbf{m})$ y se retropropaga usando la ecuación (2.9).

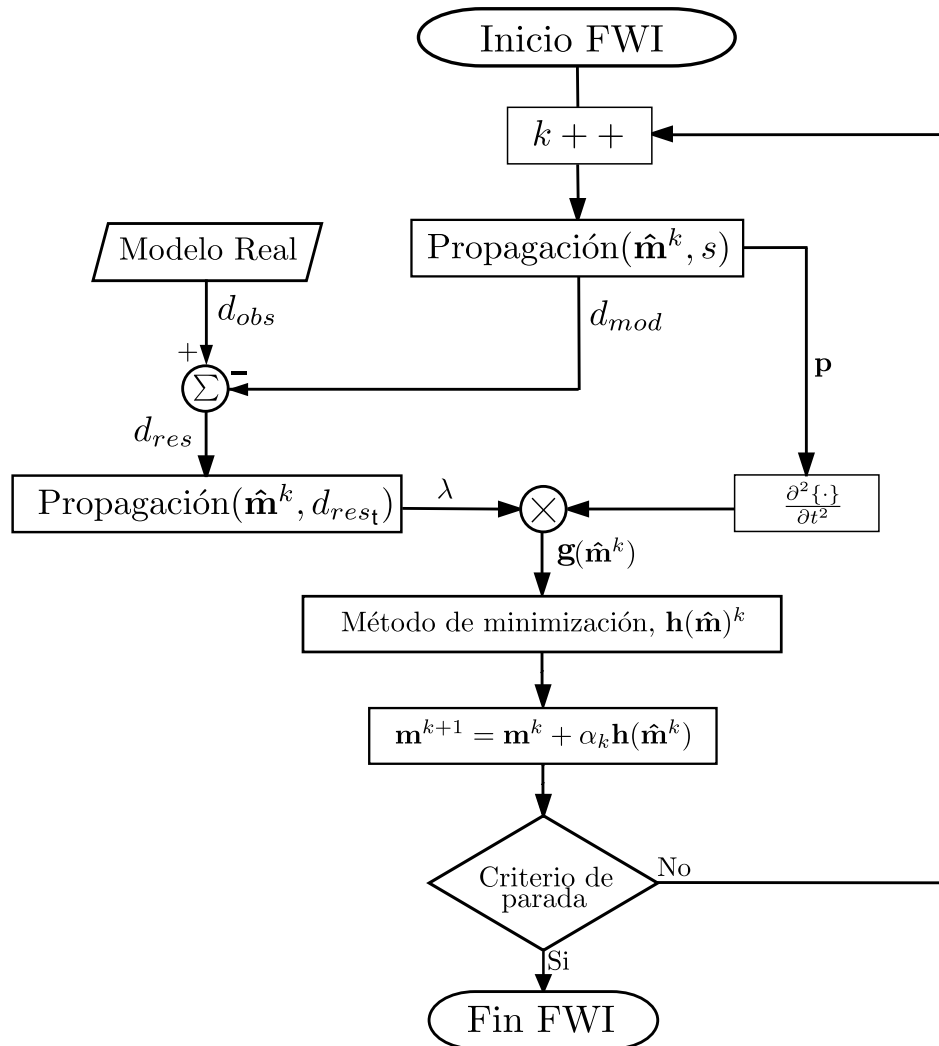
Por otra parte, la Hessiana usualmente no es calculada de manera exacta debido a su gran tamaño y alto costo computacional. Por esta razón, se plantea un método general para el cálculo del siguiente modelo de velocidad [19] como:

$$\hat{\mathbf{m}}^{k+1} = \hat{\mathbf{m}}^k + \alpha_k \mathbf{h}(\hat{\mathbf{m}}^k), \quad (3.7)$$

donde $\mathbf{h}(\hat{\mathbf{m}}^k)$ es la dirección de búsqueda, determinada por métodos de gradiente como el gradiente descendente o empleando métodos quasi-Newton como el L-BFGS [20].

Un esquema general de la implementación de la FWI es mostrado en la Figura 7. La iteración k del algoritmo inicia con un valor de cero, el cuál va incrementando en cada iteración.

Figura 7: Esquema general de una implementación FWI. Modificado de [21].



Los datos observados d_{obs} son registrados en superficie mediante una adquisición sísmica sobre el modelo real. El modelo estimado $\hat{\mathbf{m}}^k$ en la primera iteración toma el valor de un modelo inicial o punto de partida $\hat{\mathbf{m}}^0$, y a partir de este se actualiza hasta que se cumple un criterio de parada definido. Realizando una propagación hacia adelante con la ecuación (2.9) sobre $\hat{\mathbf{m}}^k$ usando una fuente en la superficie, ecuación (2.26), se obtiene d_{mod} y la derivada temporal del campo de onda $\partial^2 \mathbf{p} / \partial t^2$. Con los datos residuales d_{res} usados como fuente, se propagan en tiempo inverso (desde la última

muestra registrada hacia la primera) y se obtiene el campo retropropagado λ . Con el campo propagado y el campo retropropagado se realiza el cálculo del gradiente con la ecuación (3.6).

A partir de este punto se emplea alguna técnica para determinar la dirección de avance \mathbf{h}_k y el paso de avance α_k . Al determinar y calcular estos valores se realiza la actualización del modelo de velocidades, se evalúa el criterio de parada y se repite todo el proceso si no se cumple esta condición. Al final de la implementación de la FWI, se genera un modelo final de velocidades y una curva que determina el avance de la inversión en la función de costo, siendo esta curva típicamente descendente.

3.1.1. Método del gradiente descendente

Una vez que el gradiente de la función de costo es obtenido utilizando el método del estado adjunto [18], la forma más sencilla de obtener una dirección de búsqueda a partir del gradiente es utilizando el método del gradiente descendente, en donde la dirección de búsqueda $\mathbf{h}(\hat{\mathbf{m}}^k)$ es el gradiente de la función de costo, es decir:

$$\mathbf{h}(\hat{\mathbf{m}}^k) = -\mathbf{g}(\hat{\mathbf{m}}^k) . \quad (3.8)$$

Este método se basa en que el gradiente de una función evaluado en un punto, indica la dirección en la que la función aumenta más rápidamente desde ese punto de interés [17]. Por lo tanto, siguiendo la dirección opuesta, se espera una disminución en la función de costo. Al reemplazar la ecuación (3.8) en la ecuación (3.7), la nueva función para actualizar el modelo de velocidades está dada por:

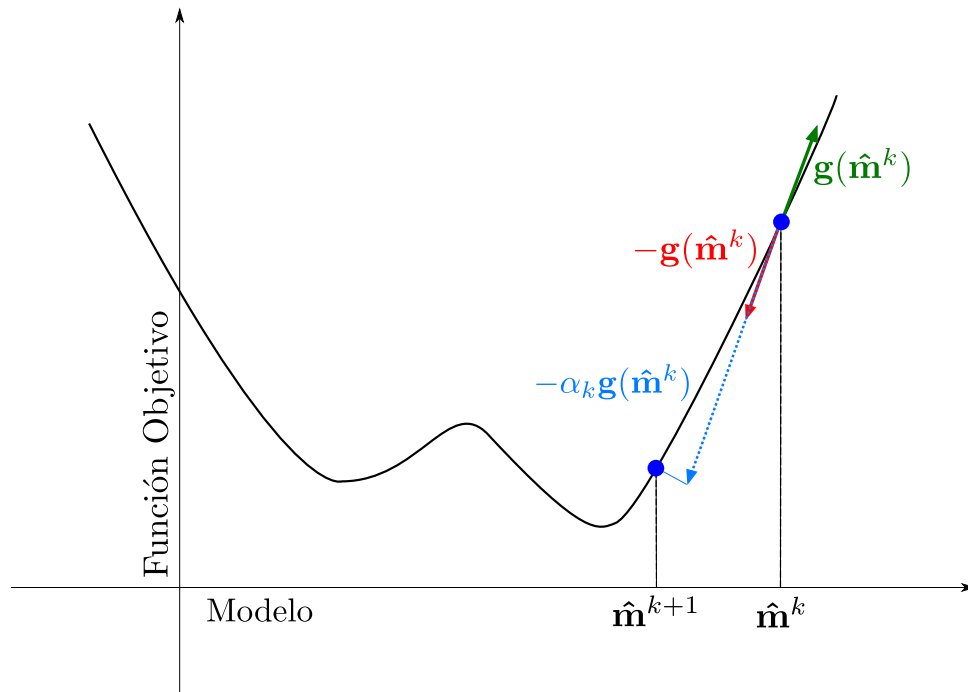
$$\hat{\mathbf{m}}^{k+1} = \hat{\mathbf{m}}^k - \alpha_k \mathbf{g}(\hat{\mathbf{m}}^k) . \quad (3.9)$$

La Figura 8 muestra el comportamiento del método del gradiente descendente, para una función de costo de un modelo con un único parámetro. El gradiente de la función objetivo en el punto $\hat{\mathbf{m}}^k$, indica la dirección de máximo crecimiento de la función y se denota con la flecha verde. En color rojo, se muestra el gradiente negativo (gradiente descendente), y luego se escala por un valor α_k (flecha en color azul) para determinar un paso de avance en el función objetivo y alcanzar $\hat{\mathbf{m}}^{k+1}$, el cuál es una proyección del avance del modelo sobre la función objetivo. En este caso ejemplo, el α_k es un valor superior a 1. Este método usa un valor de α_k constante para todas las iteraciones

El método del gradiente descendente es eficiente cuando $\hat{\mathbf{m}}^k$ aún se encuentra alejado del mínimo de la función objetivo. Cuando $\hat{\mathbf{m}}^k$ empieza a converger hacia el mínimo local más cercano, este método tiene un comportamiento oscilatorio sobre el mínimo de la función, dado que el paso de avance es un valor fijo determinado por α_k .

El valor de α_k es elegido comúnmente de manera *a priori* y se realiza algunas iteraciones para analizar los primeros pasos de avance en la FWI. Si se escoge un

Figura 8: Actualización del punto de partida usando el método del gradiente descendente.



valor de α_k pequeño, la inversión necesitará un mayor número de iteraciones para llegar a una región cercana al mínimo. En cambio, si es un valor grande, presentará un comportamiento oscilatorio rápidamente y es probable que salte entre diferentes regiones de mínimos cercanos. La elección de un α_k adecuado hace que se presente un comportamiento tipo exponencial decreciente en la función de costo, en el cuál no se presenten oscilaciones.

Se realiza la implementación de la FWI 3D para un medio isótropo con densidad constante, usando el método del gradiente descendente. El modelo escogido para realizar la implementación es un modelo sintético de velocidad constante (2000 m/s) y en su centro un cubo difractor con una velocidad superior (2500 m/s). El punto de partida para la inversión es un modelo con una velocidad constante de 2000 m/s.

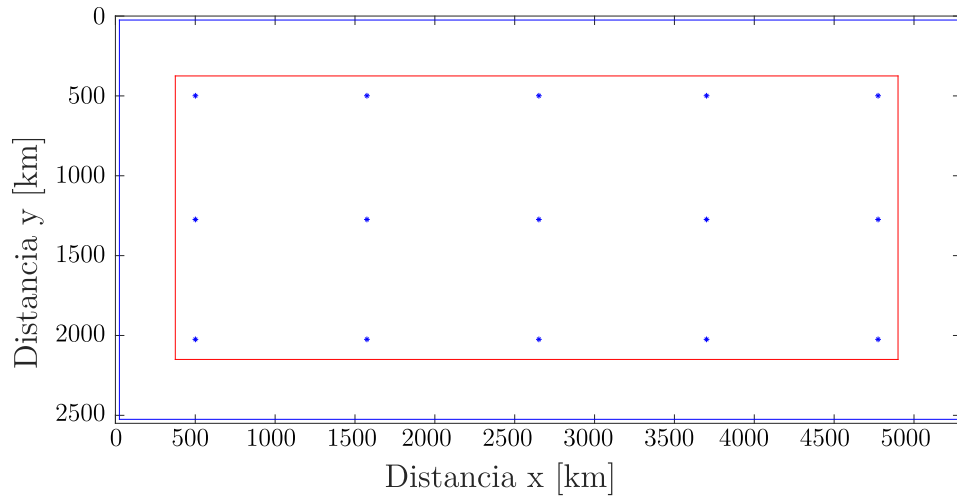
El modelo tiene una profundidad de 1700 m, y dimensiones superficiales de 5250 m y 2500 m. Se eligen los parámetros para realizar la inversión con una dispersión numérica máxima del 0,2% y que se cumpla el criterio de estabilidad de la ecuación (2.25).

Se define un muestreo espacial $dh = 25$ m, obteniendo un modelo de $211 \times 101 \times 68$ puntos. Se implementa un enfoque multiescala con frecuencias de 5, 10 y 15 Hz, donde el modelo final obtenido por cada frecuencia anterior, se emplea como punto de partida para la siguiente frecuencia, con el fin de resolver los componentes frecuenciales bajos primero, asociados a las formas del modelo y luego los componentes frecuenciales altos,

asociados detalles presentes en estas formas. Se realiza la implementación con un tiempo de grabación de 2,7 segundos con un muestreo temporal $dt = 1,8$ ms.

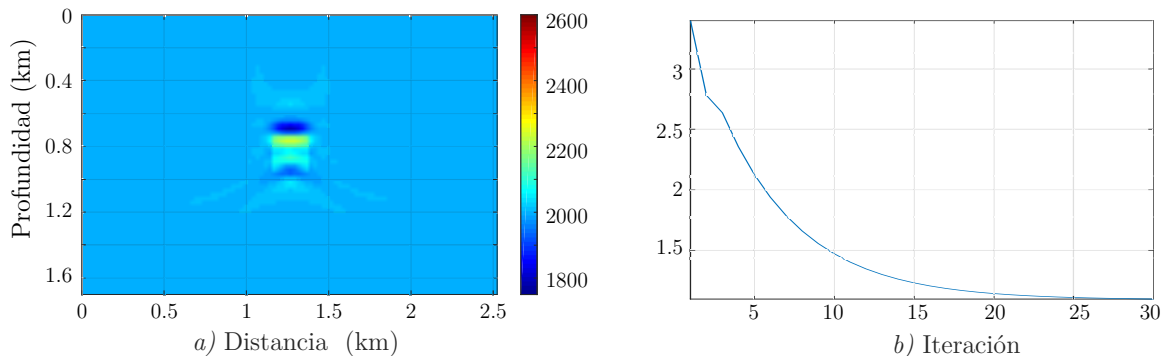
La distribución de fuentes en superficie se muestra en la Figura 9. El interior de la zona roja indica la zona de interés de reconstrucción del modelo, la zona exterior es la usada para realizar el proceso de atenuación del campo de presión y evitar las reflexiones no naturales dentro del modelo. Se distribuyen 15 fuentes en la superficie

Figura 9: Posición espacial de las fuentes en la superficie del modelo.



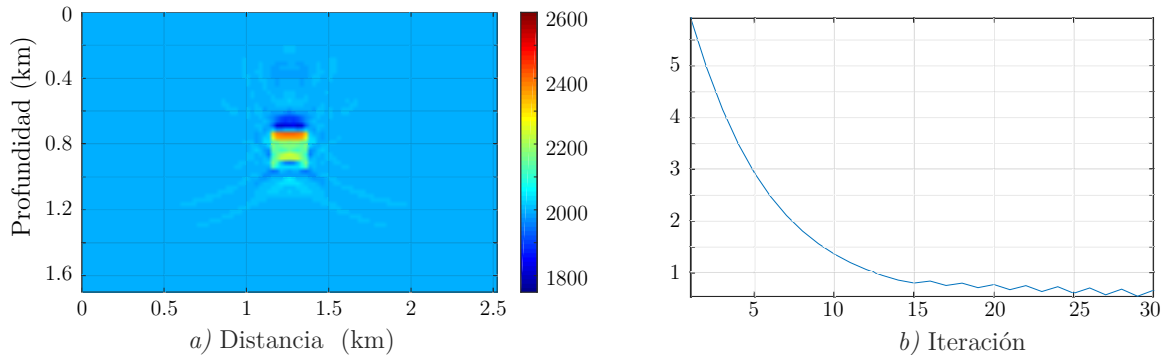
En la Figura 10 se muestra un corte en dirección del eje y del modelo final obtenido con $\alpha_k = 90$, realizando 30 iteraciones y empleando una ondícula con una frecuencia de 5Hz. La función objetivo tiene un comportamiento tipo exponencial decreciente, sin oscilaciones en el punto en el cual el comportamiento se estabiliza.

Figura 10: Resultados de la FWI 3D multiescala con una ondícula de frecuencia de 5 Hz. a) Corte en dirección del eje y del modelo final y b) evolución de la función de costo



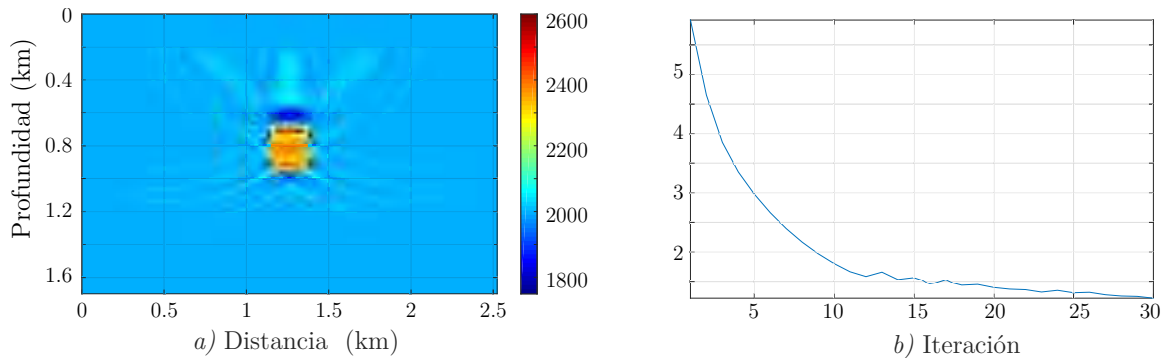
A partir del modelo final obtenido por la inversión de 5Hz, mostrado en la Figura 10, se realiza la FWI usando una frecuencia de 10Hz con $\alpha_k = 130$. Los resultados se muestran en la Figura 11. La función objetivo presenta un comportamiento oscilatorio en las ultimas iteraciones.

Figura 11: Resultados de la FWI 3D multiescala con una ondícula de frecuencia de 10 Hz. a) Corte en dirección del eje y del modelo final y b) evolución de la función de costo



De igual manera que en la frecuencia anterior, se emplea el modelo final obtenido en la inversión previa como punto de partida para la FWI con una frecuencia de 15Hz y con $\alpha_k = 100$. El modelo final obtenido por la FWI 3D con el método del gradiente descendente y usando un enfoque multiescala se muestra en la Figura 12.

Figura 12: Resultados de la FWI 3D multiescala con una ondícula de frecuencia de 15 Hz. a) Corte en dirección del eje y del modelo final y b) evolución de la función de costo



Se realizan en total 90 iteraciones, empleando un proceso multiescala de 3 frecuencias. La Figura 13 presenta los perfiles de velocidad en profundidad alcanzados por el modelo final obtenido en cada paso de frecuencia. Los perfiles de velocidad se toman en el centro de la superficie del modelo. A medida que la frecuencia aumenta y se emplea la información obtenida por la frecuencia anterior, se realiza una mejor reconstrucción.

Figura 13: Perfiles de velocidad del modelo final obtenido en cada paso de frecuencia para el modelo del cubo difractor empleando el método del gradiente descendente.

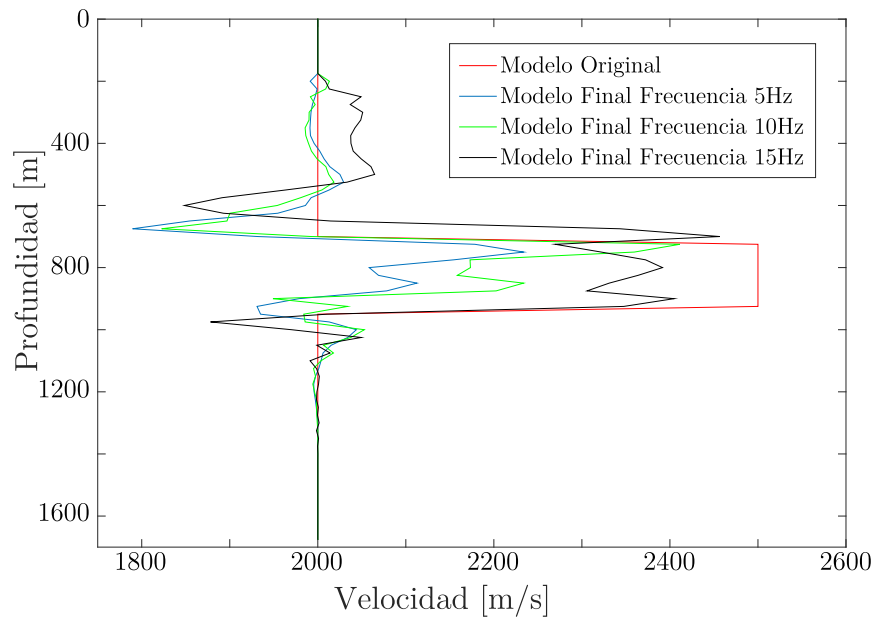
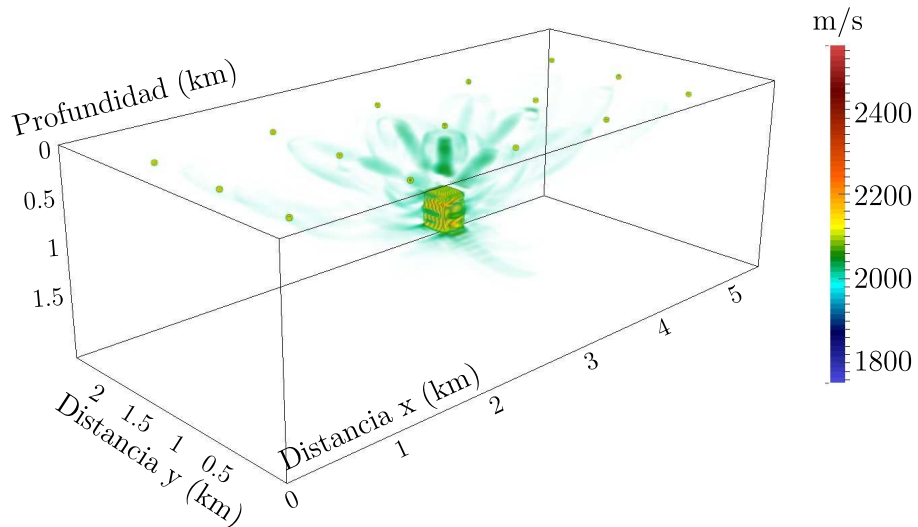


Figura 14: Modelo final de velocidad obtenido en la FWI 3D usando el método del gradiente descendente.



El modelo final reconstruido se aprecia en la Figura 14, donde para efectos de visualización, se resalta el efecto de la energía de las fuentes en el proceso de inversión y se elimina el valor de capa de velocidad constante, para mostrar la reconstrucción del cubo difractor en el interior del modelo.

3.1.2. Método de L-BFGS

Para realizar un mejor proceso de inversión se elige un método cuasi-Newton, el cual no solo da información respecto a la dirección de búsqueda, sino también aporta información sobre la curvatura de la función de costo asociado a $\hat{\mathbf{m}}^k$.

El algoritmo cuasi-Newton más popular es el método de BFGS, llamado así por sus desarrolladores Broyden, Fletcher, Goldfarb y Shanno. Este método propone guardar todos los modelos y gradientes obtenidos en cada una de las iteraciones realizadas en la inversión, y luego operarlos para obtener una dirección de búsqueda $\mathbf{h}(\hat{\mathbf{m}}^k)$.

Sin embargo en una FWI 3D, el costo computacional debido al almacenamiento de estos gradientes y modelos puede ser muy grande y difícil de manejar. A medida que se avanza en las iteraciones de la inversión y se acerca al mínimo de la función objetivo, las iteraciones iniciales dejan de aportar información relevante para el proceso de minimización y se pueden convertir en un lastre para el proceso de inversión.

Por esta razón, se emplea una versión que limita el número de gradientes y modelos de velocidad que se guardan para calcular $\mathbf{h}(\hat{\mathbf{m}}^k)$. El método L-BFGS [17] utiliza los gradientes y modelos de las l iteraciones anteriores (típicamente con $l \leq 10$) para calcular una aproximación de $(\mathbf{H}(\hat{\mathbf{m}}^k))^{-1} \mathbf{g}(\hat{\mathbf{m}}^k)$ recursivamente, tal como se presenta en el Algoritmo 1.

Algoritmo 1 Método de L-BFGS

```

1:  $\mathbf{q} \leftarrow \mathbf{g}(\hat{\mathbf{m}}^k)$ ;
2: for  $i = k - 1 : -1 : k - l$  do
3:    $\epsilon_i \leftarrow \sigma_i \mathbf{s}_i^T \mathbf{q}$ ;
4:    $\mathbf{q} \leftarrow \mathbf{q} - \epsilon_i \mathbf{y}_i$ ;
5: end for
6:  $\mathbf{r} \leftarrow \mathbf{D}_k^0 \mathbf{q}$ ;
7: for  $i = k - l : 1 : k - 1$  do
8:    $\mathbf{r} \leftarrow \mathbf{r} + \mathbf{s}_i (\epsilon_i - \sigma_i \mathbf{y}_i^T \mathbf{r})$ ;
9: end for

```

Donde \mathbf{s}_k representa la diferencia entre los modelos de velocidad, y \mathbf{y}_k sirve como un operador de escala, que trata de emular valores verdaderos de la matriz Hessiana. De esta manera, el método L-BFGS encuentra mejores y más precisas direcciones de búsqueda que el método del gradiente descendente [22]. El método de L-BFGS está determinado por el Algoritmo 1 y el conjunto de ecuaciones desde la (3.10) hasta la (3.14).

$$\mathbf{s}_k = \hat{\mathbf{m}}^{k+1} - \hat{\mathbf{m}}^k, \quad (3.10)$$

$$\mathbf{y}_k = \mathbf{g}(\hat{\mathbf{m}}^{k+1}) - \mathbf{g}(\hat{\mathbf{m}}^k), \quad (3.11)$$

$$\sigma_k = \frac{1}{\langle \mathbf{s}_k, \mathbf{y}_k \rangle}, \quad (3.12)$$

$$\mathbf{D}_k^0 = \gamma_k \mathbf{I}, \quad (3.13)$$

$$\gamma_k = \frac{\langle \mathbf{s}_k, \mathbf{y}_k \rangle}{\langle \mathbf{y}_k, \mathbf{y}_k \rangle}. \quad (3.14)$$

Este proceso se implementa en cada iteración y la nueva dirección búsqueda del siguiente modelo de velocidad está determinada por $-\mathbf{r}_k$. La ecuación que define el siguiente modelo de velocidades es:

$$\hat{\mathbf{m}}^{k+1} = \hat{\mathbf{m}}^k - \alpha_k \mathbf{r}_k, \quad (3.15)$$

donde el valor de α_k es escogido tal que cumpla la condición

$$\Phi(\hat{\mathbf{m}}^k - \alpha_k \mathbf{r}_k) < \Phi(\hat{\mathbf{m}}^k). \quad (3.16)$$

Típicamente comenzando por $\alpha_k = 1$, y disminuyendo este valor si no cumple la condición definida en la ecuación (3.16) hasta que sea cumplida, recordando que $\alpha_k \in (0, 1]$. Se podrían implementar otros criterios tales como las condiciones de Wolfe [23], pero dado el incremento en el costo computacional y que la función de costo $\Phi(\hat{\mathbf{m}})$ es siempre positiva, tales condiciones serán satisfechas por la condición de la ecuación (3.16).

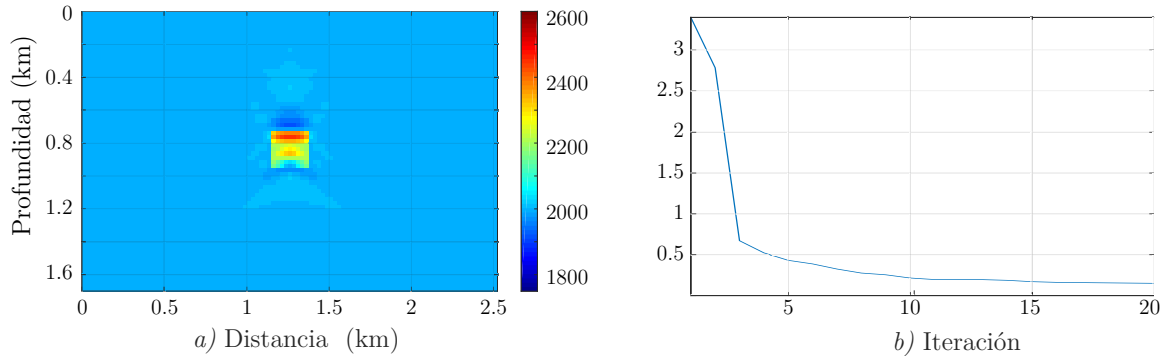
Para poder implementar este método se requiere un histórico de al menos dos modelos de velocidad y dos gradiente calculados. Por esta razón, la primera iteración se realiza usando el método del gradiente descendente, con la restricción de que el primer modelo obtenido debe tener un valor inferior en la función de costo en comparación a la función de costo del modelo de partida. Con este modelo obtenido, el modelo de partida y el gradiente de cada uno de estos modelos, se puede realizar la implementación del L-BFGS para las siguientes iteraciones de la inversión.

Se realiza el mismo experimento para implementar la FWI 3D, pero en este caso usando como paso de avance L-BFGS. El primer paso de avance en el modelo de velocidad, se obtiene por medio del método del gradiente descendente y desde allí se implementa el método L-BFGS.

El valor de α_k para la primera iteración en cada paso de frecuencia, es el mismo que el experimento anterior. Se guarda un histórico de 10 modelos y 10 gradientes, $l = 10$.

En la Figura 15 se muestra el primer resultado de la FWI 3D usando L-BFGS, con un primer paso de avance con $\alpha_k = 90$. En la función de costo obtenida, los primeros dos valores de la curva coinciden con el resultado de implementar el gradiente descendente. Se realizan 20 iteraciones por valor de frecuencia. El valor inicial para α_k empleando

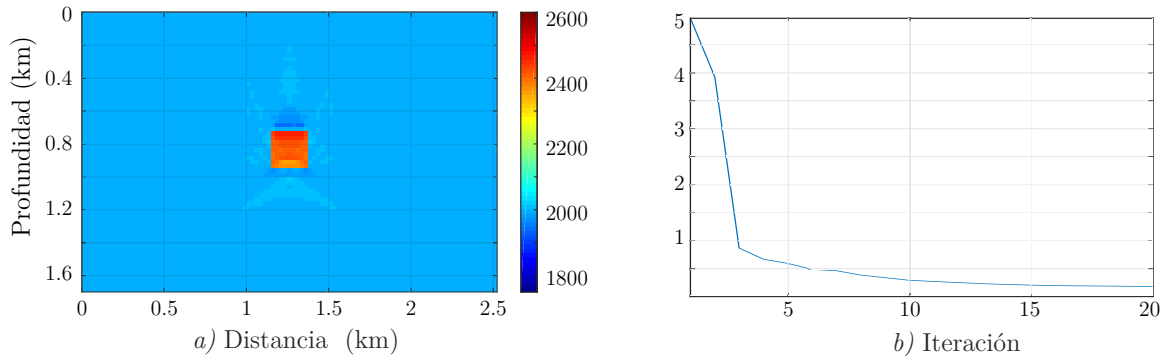
Figura 15: Resultados de la FWI 3D L-BFGS multiescala con una ondícula de frecuencia de 10 Hz. a) Corte en dirección del eje y del modelo final y b) evolución de la función de costo



el método de L-BFGS es 1 y se realiza hasta 10 intentos por iteración para verificar la condición de la ecuación (3.16).

Usando como punto de partida el modelo final de la frecuencia anterior, se ejecuta la FWI 3D con una frecuencia de 10Hz y el modelo final obtenido y su función objetivo se muestra en la Figura 16. El α_k es 130 para la primera iteración.

Figura 16: Resultados de la FWI 3D L-BFGS multiescala con una ondícula de frecuencia de 10 Hz. a) Corte en dirección del eje y del modelo final y b) evolución de la función de costo



El modelo de velocidad y la función de costo obtenidos usando L-BFGS en el proceso de inversión con la frecuencia de 15Hz, son mostrados en la Figura 17. El valor de α_k para la primera iteración es 100.

De igual manera, se realiza un comparativo con los perfiles de velocidad en profundidad obtenidos por cada paso de frecuencia contra el perfil del modelo real. En la Figura 18 se observa la evolución del modelo en cada paso de frecuencia.

Figura 17: Resultados de la FWI 3D L-BFGS multiescala con una ondícula de frecuencia de 15 Hz. a) Corte en dirección del eje y del modelo final y b) evolución de la función de costo

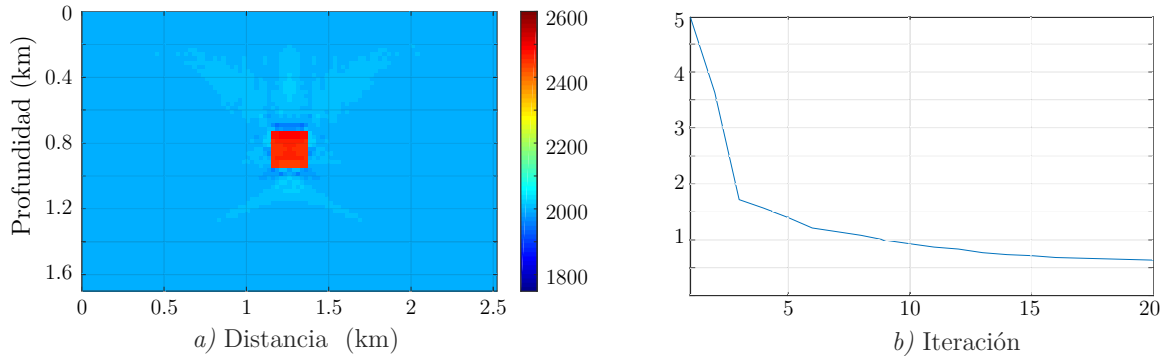
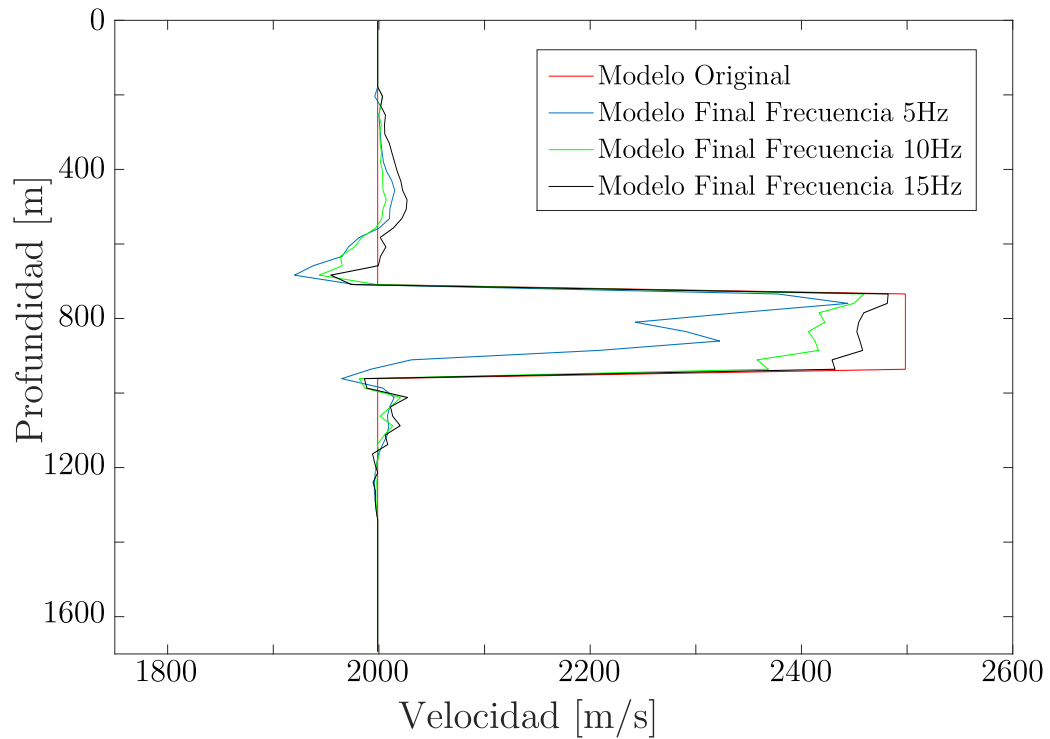
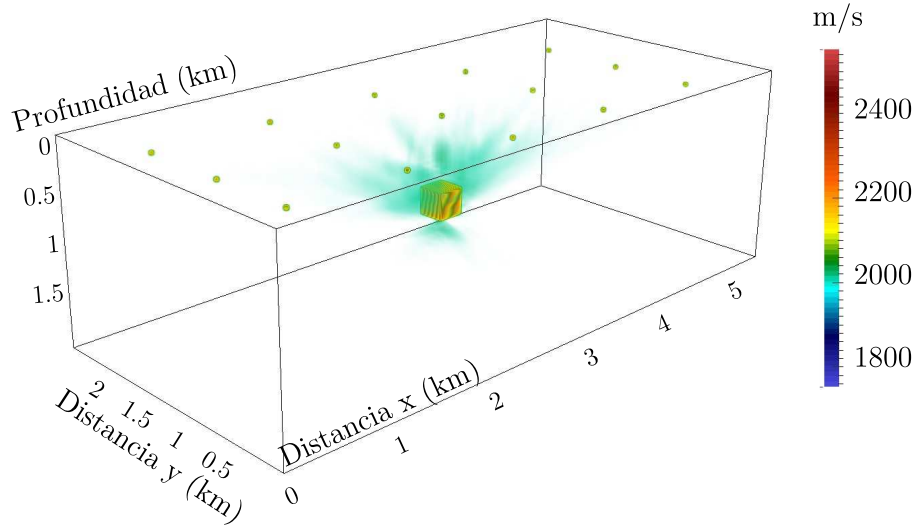


Figura 18: Perfiles de velocidad del modelo final obtenido en cada paso de frecuencia para el modelo del cubo difractor usando el método de L-BFGS.



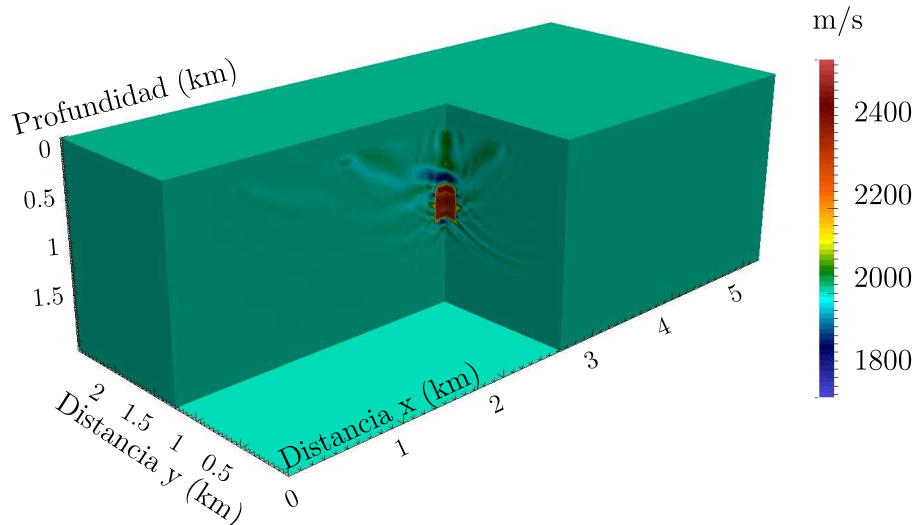
El modelo final reconstruido se aprecia en la Figura 19, donde para efectos de visualización, se resalta el efecto de la energía de las fuentes en el proceso de inversión y se elimina el valor de capa constante de fondo, para mostrar la reconstrucción del cubo difractor en el interior del modelo. Las fuentes se resaltan para ser observadas en la superficie del modelo reconstruido.

Figura 19: Modelo final de velocidad obtenido en la FWI 3D usando el método de L-BFGS.



Las Figuras 20 y 21 muestra la reconstrucción interna de los cubos difractores obtenidos por cada uno de los métodos de decremento en la función objetivo. Visualmente se nota una mejor reconstrucción obtenida por el método de L-BFGS.

Figura 20: Corte interno modelo final obtenido por el método de gradiente descendente.



Se emplea la métrica de salto de fase (CS por sus siglas en ingles) para medir cual de los dos modelos obtenidos es mejor. El CS ocurre cuando existe un desfase mayor a la mitad del periodo de la fuente [24]. Por ejemplo en la Figura 22-a se presenta un desfase mayor a $T/2$ con respecto a la Figura 22-b es decir con CS. El lóbulo n de la

Figura 21: Corte interno modelo final obtenido por el método de L-BFGS.

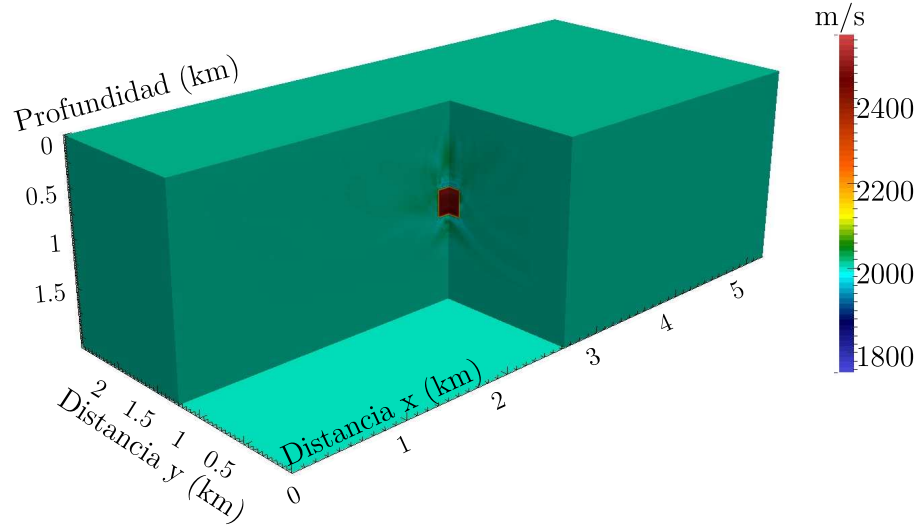


Figura 22-a se ajusta al lóbulo $n-1$ de la Figura 22-b. Por otra parte la Figura 22-c presenta un desfase menor a $T/2$ con respecto a la Figura 22-b es decir, sin CS. El lóbulo n de la Figura 22-c se ajusta al lóbulo n de la Figura 22-b.

Se realizan dos propagaciones en ambos modelos obtenidos, con fuentes situadas en la posición $[500m, 500m]$ (en un esquina de la superficie del modelo dentro de la zona de interés) y $[2600m, 1250m]$ (en el centro de la superficie del modelo). Se extrae el conjunto de trazas agrupadas por disparo común de cada propagación en el modelo final y el modelo original, y se realiza la comparación de CS traza a traza.

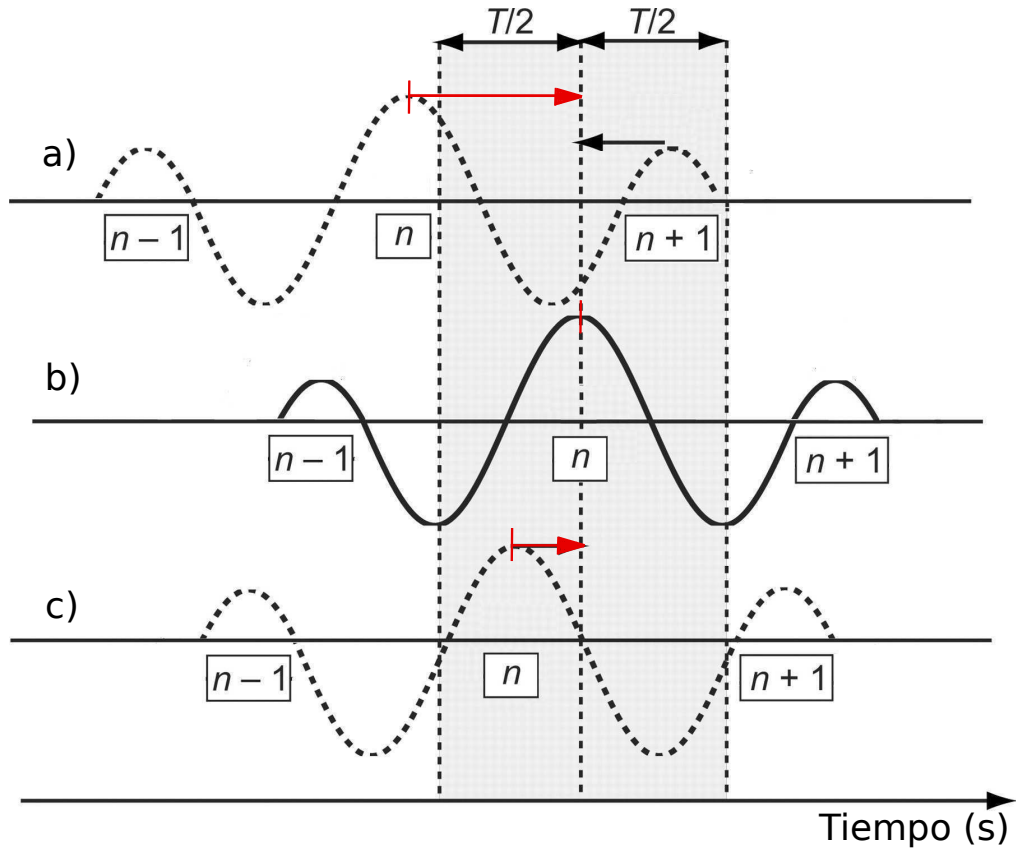
La Figura 23 se muestra un corte del CS de las propagaciones usando la fuente situada en una esquina de los modelos. Esta gráfica representa el desfase entre las trazas del modelo original y de los modelos obtenidos por el método del gradiente descendente (Figura 23 a) y el método L-BFGS (Figura 23 b). El desfase está medido entre 0 y π .

De igual manera se aprecia el mismo análisis en la Figura 24 para la propagación con la fuente situada en el centro de la superficie del modelo.

Para apreciar de manera clara el CS de todo el conjunto de trazas por disparo común de cada uno de los modelos, se hace un conteo del numero de veces que se presenta CS en cada traza, y se realiza una gráfica en base al número obtenido, proyectada en la superficie de cada modelo.

La figura 25 y 26 muestran el CS de todas las trazas cuando la fuente está situada en la esquina superior izquierda ($[500m, 500m]$) y en el centro ($[2600m, 1250m]$), respectivamente. Las gráficas de la izquierda representan el CS del modelo obtenido por el método gradiente descendente, mientras que las derecha por el método L-BFGS. Es

Figura 22: Problema de salto de fase. Adaptada de [24]



posible obtener un porcentaje de CS para cada gráfica si se cuentan la cantidad de fases instantáneas que tienen CS y se dividen sobre el número total de fases instantáneas.

El porcentaje de CS obtenido en la Figura 25-a es 6,68 % y en la Figure 25-b es 3,07%. En la propagación con la fuente centrada es 12,26 % (Figura 26-a y 1,46 % (Figura 26-b.

El contraste de resultados de la implementación de ambos métodos se compila en la Tabla 2.

Tabla 2: Comparación de tiempos e iteraciones en la inversión usando el método del gradiente descendente contra L-BFGS

	Tiempo	Iteraciones
Grad. desc.	608,1[<i>min</i>]	90
L-BFGS	634,8[<i>min</i>]	60

El método de L-BFGS requiere de una mayor cantidad de tiempo por iteración, debido al cálculo de un α_k que cumpla la condición de la ecuación (3.16), la cuál se verifica realizando una propagación adicional en cada iteración. De esta manera, en el

Figura 23: Salto de fase presentado en un corte en dirección del eje y con una fuente situada en una esquina de la superficie de los modelos

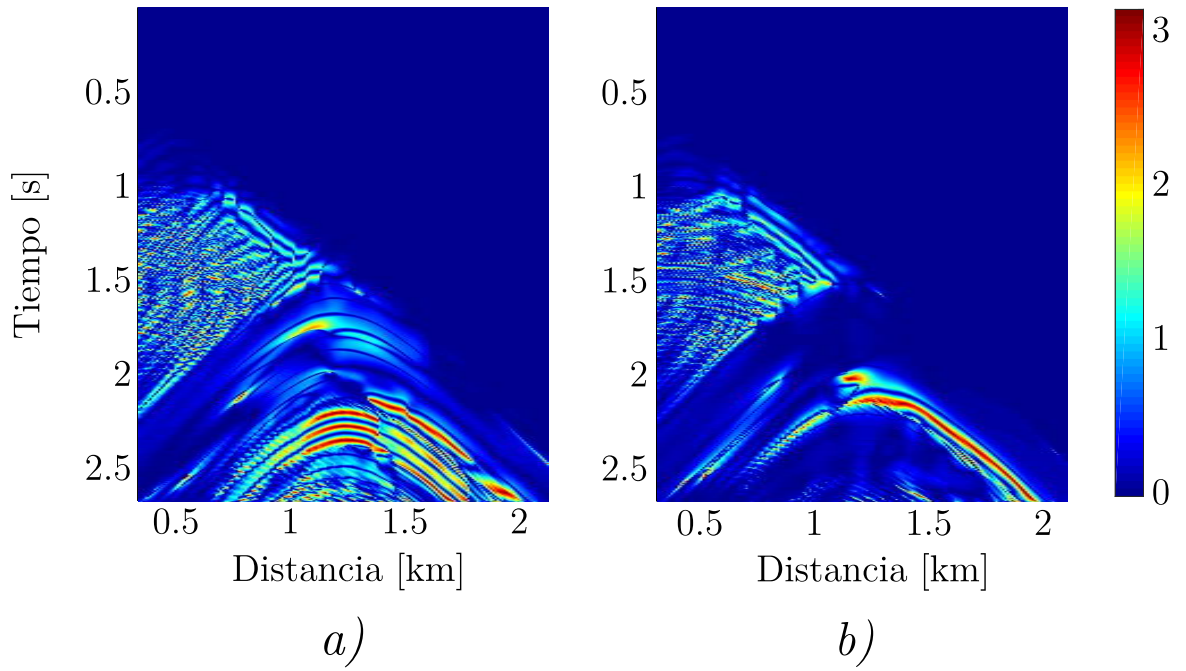


Figura 24: Salto de fase presentado en un corte en dirección del eje y con una fuente situada en el centro de la superficie de los modelos

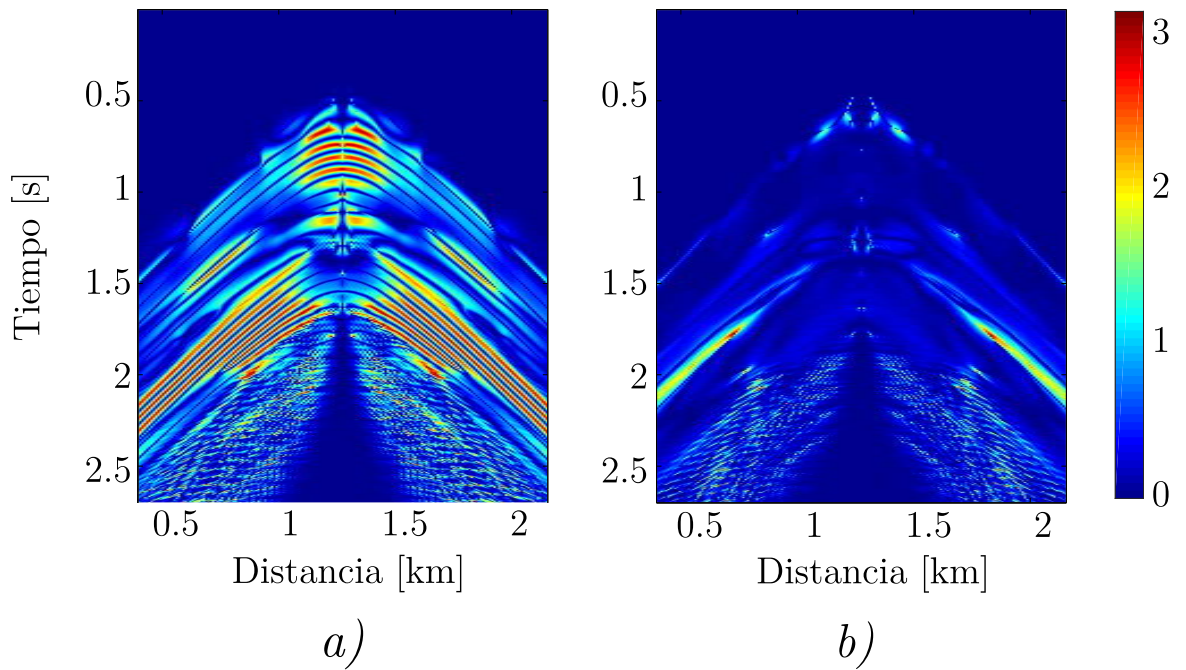


Figura 25: Comparación del salto de fase del modelo final obtenido usando el método del gradiente descendente(a) y el método de L-BFGS(b) para una fuente situada en la esquina de la superficie del modelo del cubo difractor reconstruido por cada método, con un CS de 6,68 % (a) y 3,07 % (b).

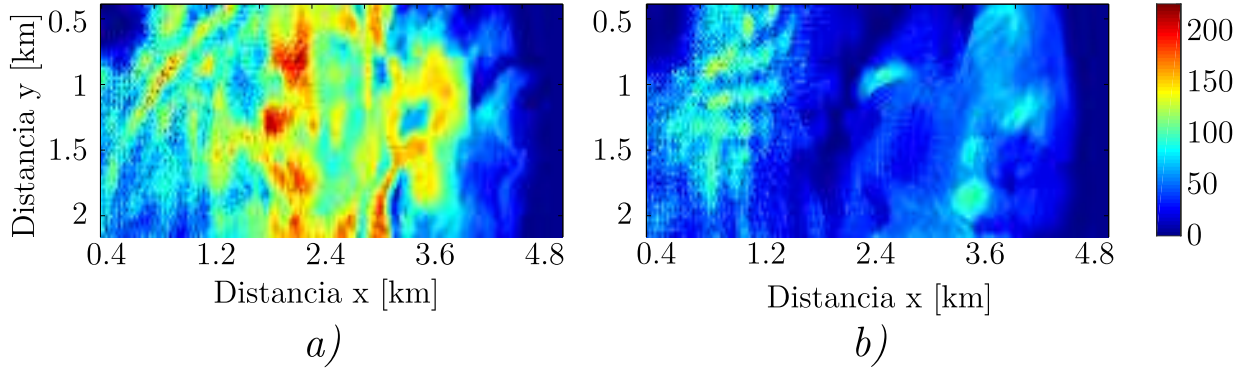
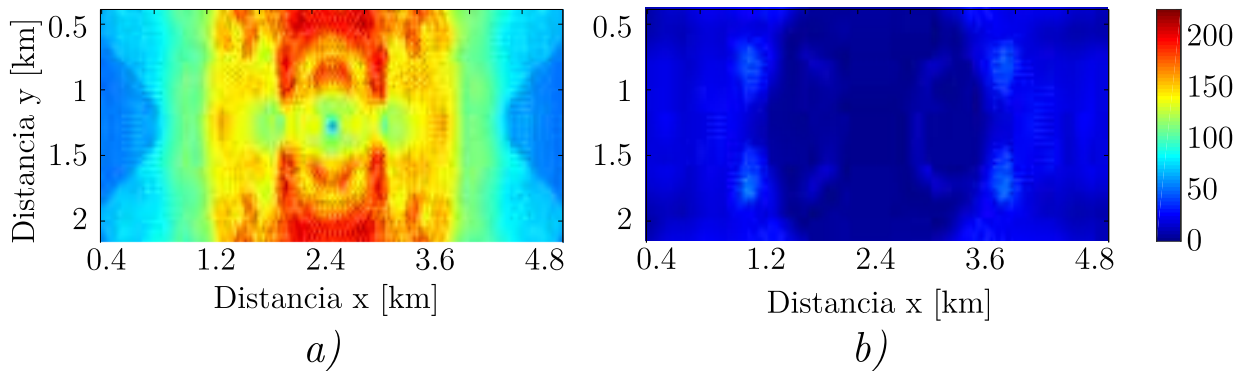


Figura 26: Comparación del salto de fase del modelo final obtenido usando el método del gradiente descendente(a) y el método de L-BFGS(b) para una fuente situada en el centro de la superficie del modelo del cubo difractor reconstruido por cada método, con un CS de 12,26 % (a) y 1,46 % (b).



mejor de los casos, L-BFGS requiere de 3 iteraciones para calcular un paso de avance en el modelo (para verificar el cumplimiento de la condición de la ecuación (3.16)), pero así mismo, se requiere un menor número de iteraciones para obtener un mejor modelo que en el método del gradiente descendente, como se puede verificar a partir del análisis de CS en las Figuras 23, 24, 25 y 26.

4. ESTRATEGIAS DE IMPLEMENTACIÓN

Para llevar la inversión de onda completa 3D en el dominio del tiempo a una implementación en una plataforma de cómputo paralelo como las GPUs, se requiere evaluar las limitantes que conlleva este proceso. En Abreo et al. [7] se implementa la FWI 2D en el dominio temporal usando GPUs para reducir el costo computacional. En este trabajo se destaca que aunque las GPUs son eficientes en el procesamiento masivo de datos (ideal para procesamiento tridimensional), presentan dificultades para almacenar grandes cantidades de información.

De esta manera, es necesario plantear estrategias para poder implementar la inversión 3D en GPUs. Así, en este capítulo se muestran las estrategias implementadas, enfocadas en reducir la memoria requerida para la implementación de la FWI 3D, y el uso de un cluster multi-GPU para reducir el tiempo de cómputo de la inversión.

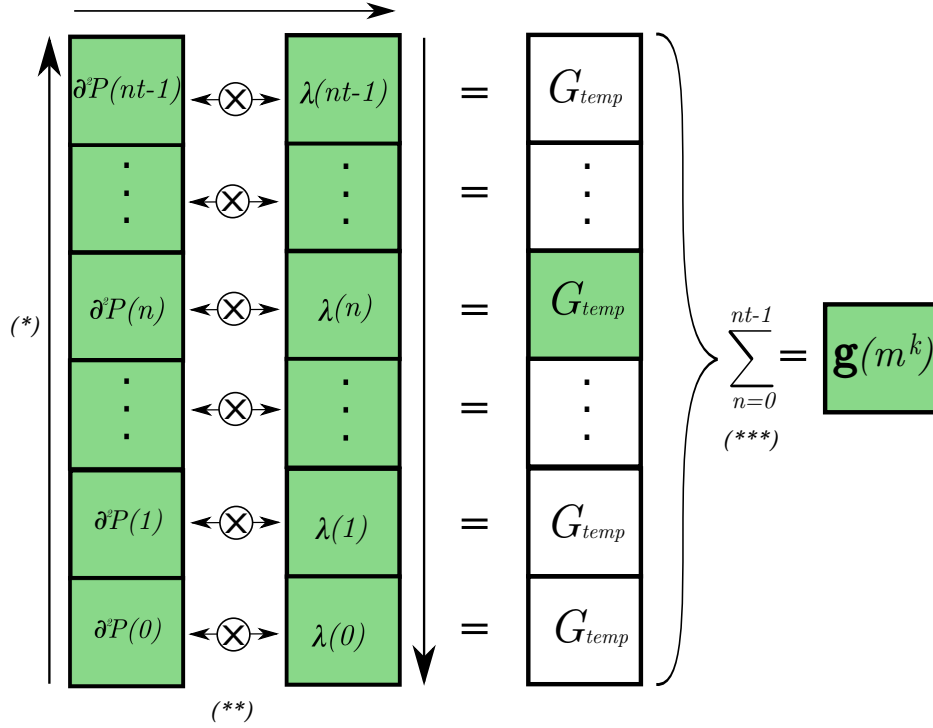
4.1. Implementación de referencia

Realizando el cálculo del gradiente como lo plantea Plessix [18], en la Figura 27 se muestra la manera como se calcula el gradiente siguiendo el procedimiento planteado en Abreo et al. [7], donde se almacenan y usan el campo de propagación y el de retropropagación. Esta forma de calcular el gradiente, la llamaremos *implementación de referencia*. Las flechas negras indican el flujo de las operaciones y los cuadrados en verde representan la memoria guardada en la GPU.

Esta forma de implementación sigue el flujo de cálculo planteado en la Figura 7. Primero (*), se realiza una propagación hacia adelante del campo de onda sobre el modelo actual, usando una fuente puntual en superficie. En esta propagación, se calcula y almacena la derivada del campo de onda $\partial^2 \mathbf{p} / \partial t^2$, denominada $\partial^2 \mathbf{p}(n)$ en la figura; el dato modelado d_{mod} también se computa en este paso. Se determina d_{res} y se realiza la retropropagación de estos datos para calcular y almacenar el campo de onda en cada paso de tiempo $\lambda(n)$. El campo de onda λ es calculado desde el tiempo $n = nt - 1$ hasta el tiempo $n = 0$.

Una vez obtenido y almacenado estos dos campos de onda, se recorren simultáneamente desde el tiempo discreto $n = 0$ hasta el tiempo $n = nt - 1$ y se realiza una multiplicación punto a punto (**) en cada paso de tiempo i para calcular un G_{temp} de la talla del modelo. En cada paso de tiempo G_{temp} es acumulado (***) para determinar

Figura 27: Cálculo del gradiente usando la implementación de referencia [7].



el gradiente $\mathbf{g}(\hat{\mathbf{m}}^k)$ asociado a cada disparo en superficie. El gradiente total de cada iteración, es la suma de todos los gradientes obtenidos por cada disparo.

La memoria requerida para poder realizar esta implementación en una GPU es aproximadamente de:

$$Ram_{3D} \approx 2 \cdot \beta \cdot (Nx \cdot Ny \cdot Nz \cdot nt), \quad (4.1)$$

donde las unidades de la expresión son [GiB]. La variable β está asociado al tipo de precisión usada para almacenar la información. En este trabajo de investigación se usará precisión flotante (32 bits), y el valor de β es $4/2^{30}$. Las variables Nx , Ny , Nz , son las dimensiones del modelo y nt son los pasos de tiempo usados en la propagación.

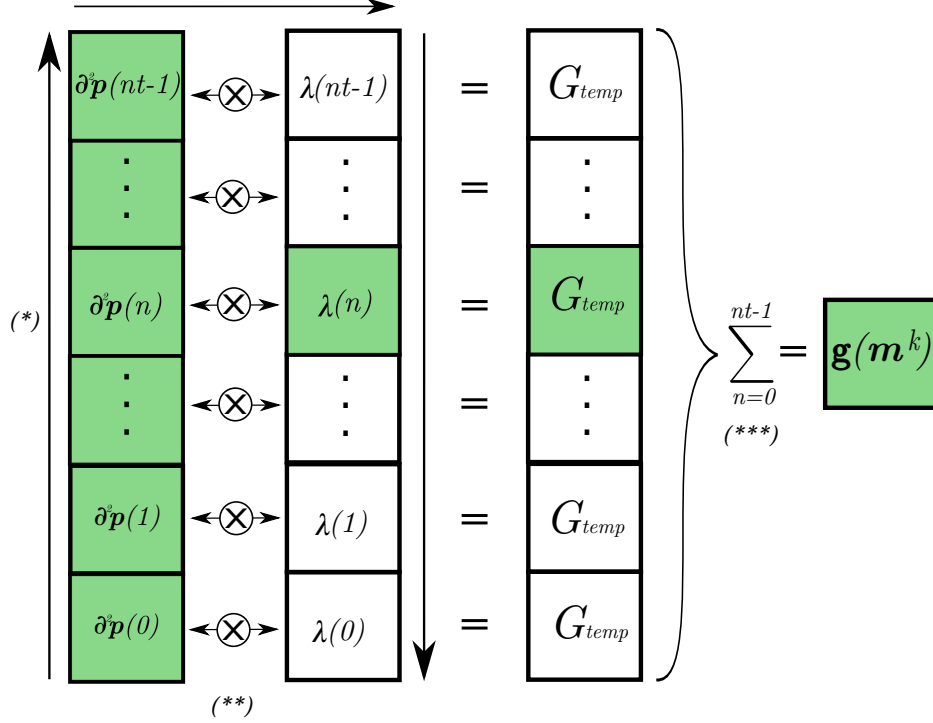
Para implementar la FWI 3D en un experimento sísmico con un modelo como el planteado en la sección 3.1.1, la memoria requerida sería aproximadamente de 16,2 [GiB], superando los 12GiB de las GPUs Tesla K40c usadas para estas pruebas [25].

4.2. Primera estrategia de implementación

La primera estrategia de implementación para reducir la memoria requerida por el cálculo del gradiente se muestra en la Figura 28. Esta estrategia consiste en almacenar en la GPU la derivada del campo de onda \mathbf{p} en todos los instantes de tiempo, $\partial^2 \mathbf{p}(n)$. El procedimiento es similar a la implementación anterior hasta el punto en el que se realiza

la retropropagación de los datos d_{res} . Mientras se calcula $\lambda(n)$, de manera simultánea se determina el G_{temp} asociado a ese instante de tiempo. Cada G_{temp} calculado es acumulado para hallar $\mathbf{g}(\hat{\mathbf{m}}^k)$.

Figura 28: Cálculo del gradiente guardando un solo campo [21].



De esta manera, no es necesario almacenar el campo de propagación λ , reduciendo en aproximadamente la mitad de la memoria requerida por el proceso de inversión, y manteniendo en igual medida la cantidad de operaciones realizadas para calcular el gradiente, por lo que el tiempo de computo es similar en ambos procedimientos.

Siguiendo el planteamiento de la ecuación (4.1), la memoria requerida para el cálculo del gradiente es aproximadamente

$$Ram_{3D} \approx \beta \cdot (Nx \cdot Ny \cdot Nz \cdot nt) . \quad (4.2)$$

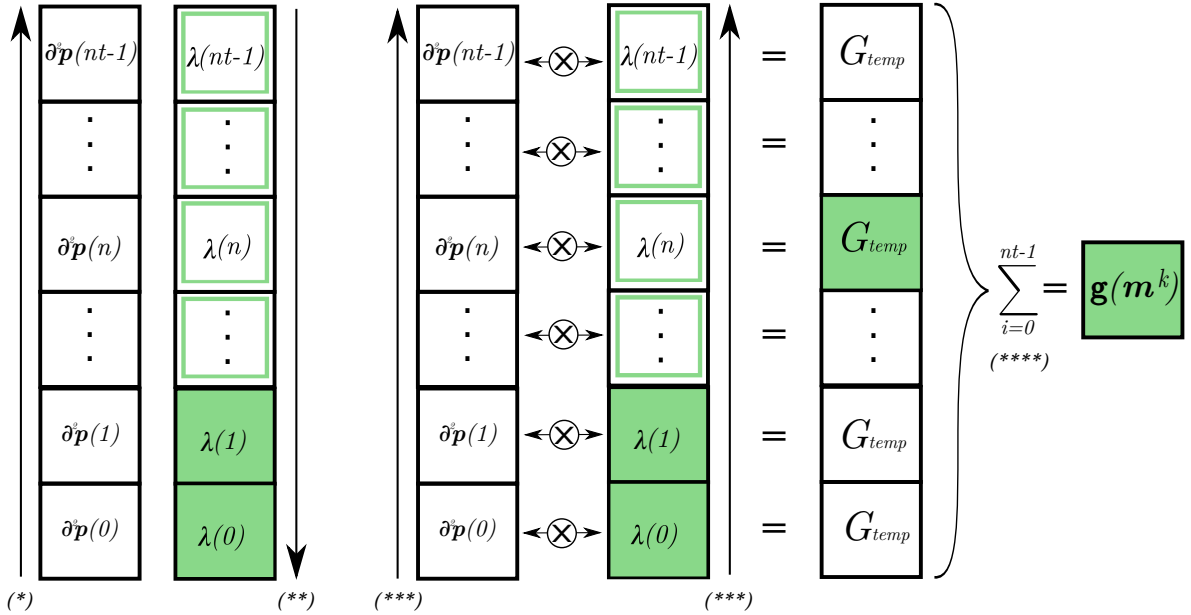
Para implementar la FWI 3D en un experimento sísmico con en un modelo como el planteado en la sección 3.1.1, la memoria requerida sería aproximadamente de 8,1 [GiB]; haciendo viable su implementación en una GPU Tesla K40 con 12GiB de memoria DRAM [25].

4.3. Segunda estrategia de implementación

La segunda estrategia consiste en no almacenar ninguno de los dos campos de propagación $\partial^2 \mathbf{p}$ y λ . Esta estrategia solo requiere almacenar una pequeña porción de

la información del campo λ . La Figura 29 muestra el flujo del proceso para realizar el cálculo del gradiente por disparo. De manera general, esta estrategia se basa en la reconstrucción de λ , desde el último *slice* de tiempo generado hacia el primero.

Figura 29: Cálculo del gradiente guardando la información de fronteras [21].



Primero, se realiza una propagación hacia adelante (*) para determinar d_{mod} , en esta propagación no se almacena $\partial^2 \mathbf{p}(i)$. Con d_{mod} se obtiene d_{res} , el cual es retropropagado (**), y solo se guarda la información del campo λ ubicada en las fronteras no naturales del modelo, justo en el borde antes de aplicar *CPML* (ver Figura 30) para realizar la reconstrucción del campo. Luego, se realiza nuevamente una propagación hacia adelante (***) y de manera simultánea se genera λ , reconstruyendo el campo de onda a partir de la información de frontera guardada en la retropropagación. Tanto $\partial^2 \mathbf{p}(i)$ como $\lambda(i)$ son calculadas desde $i = 0$ hasta $i = n - 1$. De esta manera, se calcula G_{temp} mediante la multiplicación punto a punto en cada instante de tiempo y se va acumulando para determinar $\mathbf{g}(\hat{\mathbf{m}}^k)$.

Esta estrategia reduce en gran medida la memoria requerida por la FWI 3D pero requiere una propagación adicional para calcular el gradiente en comparación con las dos implementaciones anteriores, lo que implica en un aumento significativo en al menos 50% del tiempo de ejecución.

4.3.1. Reconstrucción de campo

Para realizar la reconstrucción del campo λ es necesario modificar la forma de realizar la propagación. Recordando la ecuación (2.8)

$$\mathbb{U}_{i,j,k}^{n+1} = 2 \cdot \mathbb{U}_{i,j,k}^n + \frac{(\mathbf{m}_{i,j,k})^2 \cdot \Delta t^2}{\Delta h^2} \left(\sum_{c=-4}^4 C_s[c] \cdot (\mathbb{U}_{i+c,j,k}^n + \mathbb{U}_{i,j+c,k}^n + \mathbb{U}_{i,j,k+c}^n) \right) - \mathbb{U}_{i,j,k}^{n-1}.$$

Se puede apreciar que la estructura de la ecuación no se modifica para realizar una propagación en un sentido o en otro, dado que solo es necesario intercambiar los términos de futuro y pasado, calculando el anterior paso de tiempo a partir de los pasos de tiempo presente y futuro ya conocidos, como sigue:

$$\mathbb{U}_{i,j,k}^{n-1} = 2 \cdot \mathbb{U}_{i,j,k}^n + \frac{(\mathbf{m}_{i,j,k})^2 \cdot \Delta t^2}{\Delta h^2} \left(\sum_{c=-4}^4 C_s[c] \cdot (\mathbb{U}_{i+c,j,k}^n + \mathbb{U}_{i,j+c,k}^n + \mathbb{U}_{i,j,k+c}^n) \right) - \mathbb{U}_{i,j,k}^{n+1}. \quad (4.3)$$

Pero se presenta un inconveniente al querer reconstruir el campo de propagación de manera inversa, dado que la onda de presión al llegar a las fronteras no naturales del modelo, es atenuada usando *CPML* (ver sección 2.2). Debido a esta atenuación la energía que llega al límite se elimina, por lo cual usando la ecuación (4.3), no es posible lograr que esta energía se colapse hacia el inicio de la propagación.

Se plantea almacenar la información de frontera en todos los instantes de tiempo de la retropropagación hacia adelante, justo en el borde antes de iniciar la atenuación por *CPML*. Al realizar la reconstrucción hacia atrás, se usará la ecuación (4.3) y en cada paso de tiempo n se inyectará en los límites del modelo la información almacenada, en su respectivo paso de tiempo. De esta manera, es posible colapsar toda la energía propagada hasta el origen de ella. Para minimizar los errores también se guarda la información de la frontera superior, para asegurar que la energía llegará a cero en el instante $n = 0$. En la Figura 30 se muestra los lugares del modelo en los cuales se almacena la información de frontera de la propagación del campo de onda.

En esta forma de propagación no es necesario implementar las condiciones de frontera *CPML* porque la energía de la propagación de onda solo fluirá en dirección contraria a los bordes del modelo.

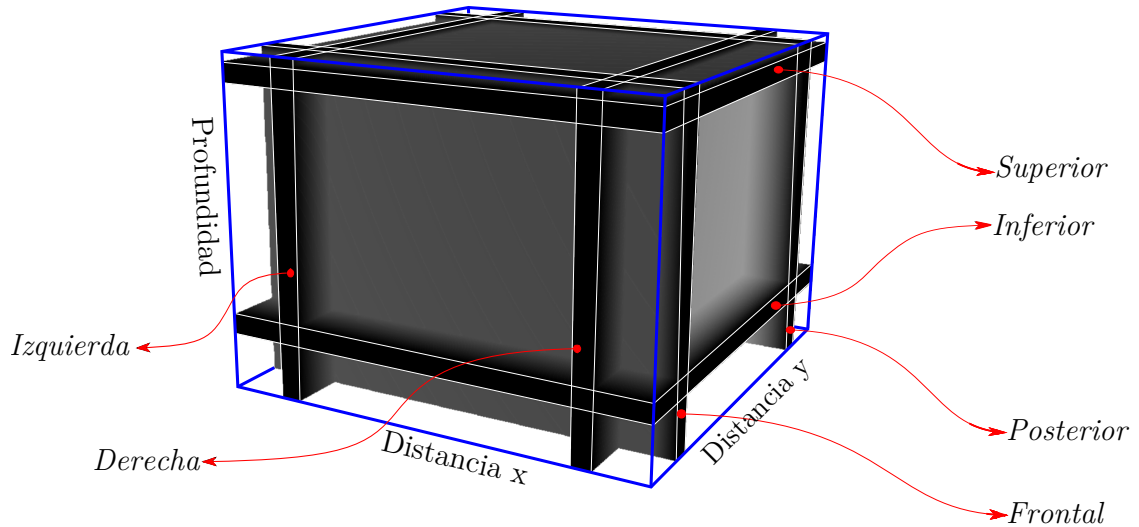
Al usar una aproximación de octavo orden para las diferencias finitas de las derivadas espaciales, es necesario almacenar un borde de 4 puntos de grosor. Así la memoria requerida para guardar la información de frontera y poder implementar la FWI 3D es aproximadamente:

$$Ram_{3D} \approx 4 \cdot \beta \cdot nt \cdot (2 \cdot Nx \cdot Ny + 2 \cdot Nx \cdot Nz + 2 \cdot Ny \cdot Nz), \quad (4.4)$$

donde el 4 representa el ancho de cada una de las secciones (superior, inferior, frontal, posterior, izquierda y derecha) y el 2 la forma de agrupar estos volúmenes por parejas.

El volumen de datos de cada una de estas secciones es de 4 veces el área de cada cara. El área de las secciones superior e inferior es $Nx \cdot Ny$, de las secciones frontal y posterior es $Nx \cdot Nz$ y de las secciones izquierda y derecha es $Ny \cdot Nz$.

Figura 30: Secciones del modelo donde se almacena la información de frontera.



4.4. División por disparos

Las estrategias planteadas anteriormente están totalmente enfocadas en la reducción de la memoria requerida para la implementación de la inversión de onda completa 3D. Para reducir el tiempo de ejecución de la inversión, se hace uso de la implementación en múltiples GPUs al mismo tiempo, repartiendo la carga de trabajo entre todas ellas.

En el cálculo del gradiente final para realizar la actualización del modelo, no importa el orden en el que se hagan los aportes por disparo, el gradiente final obtenido es el mismo. De la misma manera, no afecta si se agrupan los aportes por disparo para hacer al final una suma de estos grupos de aportes.

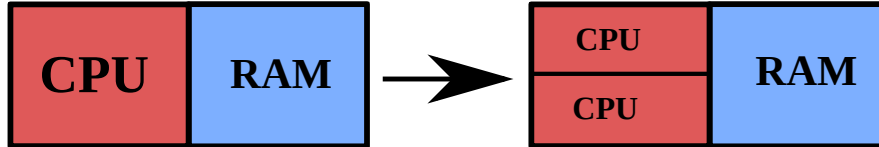
Usando esta característica se puede dividir la carga de trabajo entre múltiples GPUs, a partir de la división del proceso de inversión en grupos de disparos distribuidos equitativamente entre las GPUs dentro del cluster. De esta manera cada GPU se encarga de calcular los aportes de este grupo de disparos y acumularlos, para luego sumarlos con el resto de gradientes parciales obtenidos por las otras GPUs.

Para realizar esta división de disparos usando multiples GPUs acopladas a un procesador, se usa un protocolo de comunicación mediante MPI. Este estándar de comunicación paralela fue lanzado en 1994 y desarrollado de manera comunitaria por más de 40 organizaciones y cuyo objetivo era crear una librería de paso de mensajes que permi-

tiera crear programas que pudieran ser migrados a diferentes computadores conectados en paralelo [26].

MPI funciona usando un modelo de memoria distribuida, es decir gestiona los recursos CPU para crear procesos independientes compartiendo los recursos de memoria física, pero teniendo su propia organización de punteros. El diagrama de la distribución de recursos físicos de una implementación MPI se muestra en la Figura 31.

Figura 31: Representación de los recursos físicos al usar MPI.



Un esquema general de la implementación de división por disparos usando un cluster de GPUs es mostrado en la Figura 32. Se realiza la división equitativa de los disparos en función de las GPUs disponibles. En nuestro caso se dividen por segmentos continuos de disparos, es decir, si se tuvieran 20 disparos y 3 GPUs, la primera GPU realizaría el proceso para los primeros 7 disparos, la segunda GPU para los 7 disparos siguientes y la última GPU para los 6 disparos restantes. Una vez definidos los disparos por GPU se envía la información necesaria a cada una de ellas, como los modelos de velocidad y la posición de las fuentes.

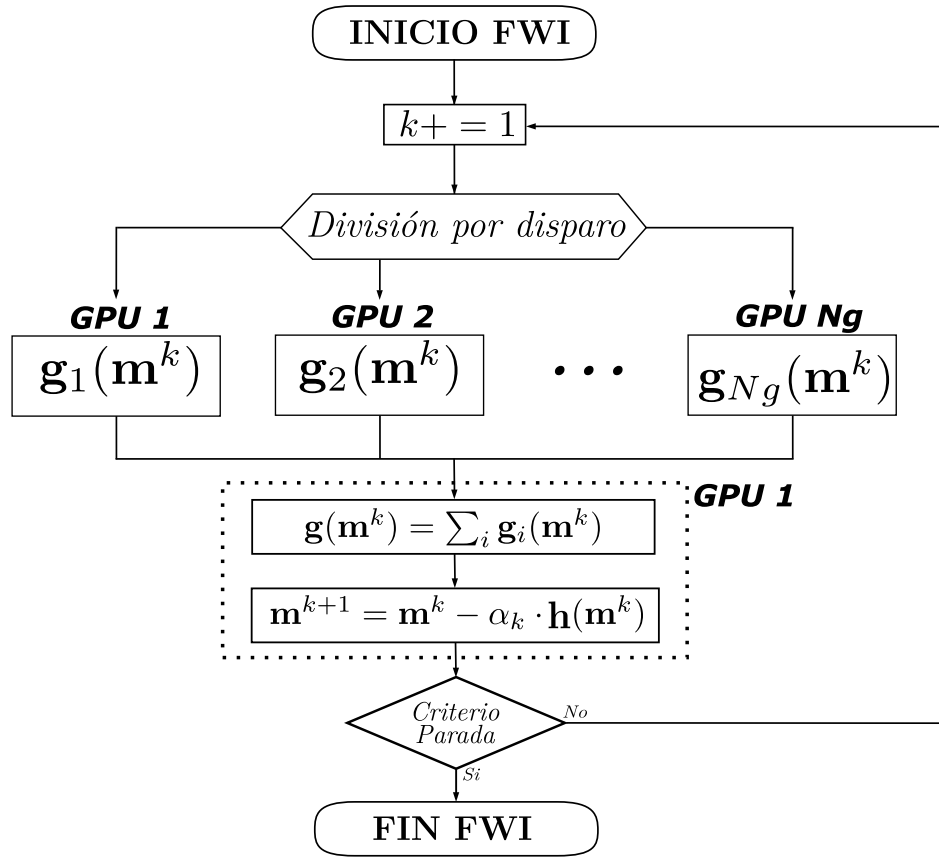
En el caso ejemplo, la primera GPU se encarga de calcular el gradiente acumulado para los primeros 7 disparos, y así sucesivamente para las otras 2 GPUs. Al finalizar de calcular todos los gradientes parciales, son enviados al procesador y desde allí enviados a la primera GPU para calcular el gradiente total y el modelo siguiente \mathbf{m}^{k+1} . Si se elige como método para la minimización de la función de costo el gradiente descendente, luego del cálculo del gradiente, se determina el paso de actualización del modelo. Si se usa el L-BFGS, la GPU 1 calcula el valor \mathbf{r}_k y se realiza la actualización del modelo, requiriendo un paso adicional para verificar el cumplimiento de la condición de la ecuación (3.16). Esta propagación adicional se realiza usando la división de disparos.

El modelo \mathbf{m}^{k+1} es enviado desde la GPU 1 al host y desde allí distribuido a todas las GPUs para realizar el proceso nuevamente si no se ha cumplido el criterio de parada de la inversión.

La ecuación (4.5) muestra una aproximación del factor de aceleración al comparar el trabajo con una sola GPU para realizar la FWI 3D respecto a la implementación en múltiples GPUs.

$$\text{Aceleración} = \frac{N_S \cdot T_{shot} + T_{serial_1}}{[(N_S/N_G)] \cdot T_{shot} + T_{serial_{MPI}}}, \quad (4.5)$$

Figura 32: Esquema FWI usando división por disparos a través de MPI. Modificado de [21].



donde T_{shot} es el tiempo requerido en calcular el gradiente asociado a un solo disparo y N_G es el numero de GPUs usadas en el proceso de inversión. El valor T_{serial_1} representa el tiempo necesario para calcular el gradiente total y la actualización del modelo cuando se calcula completamente en una sola GPU (lo que evita la transferencia de memoria entre varias GPUs). $T_{serial_{MPI}}$ tiene en cuenta el tiempo involucrado en el envío de los gradientes entre todas las GPUs hacia la GPU 1 para el cálculo del gradiente total y a su vez el tiempo para transferir este último a todas las GPUs para la actualización del modelo.

La Tabla 3 expone los resultados de implementar las estrategias 1 y 2 de reducción de memoria en combinación a la división de disparos con 1, 2, 3 y 6 GPUs. Estos resultados son obtenidos a partir de la inversión del cubo difractor partiendo de un modelo de capa constante, con pasos multifrecuencia y usando el método L-BFGS (sección 3.1.2).

De acuerdo con la Tabla 3, el tiempo de ejecución de la primera estrategia es aproximadamente un 35 % más rápido que la segunda estrategia, y esto es porque la última requiere realizar una propagación adicional; así mismo la segunda estrategia sólo necesita un 30 % de la memoria requerida por la primera estrategia. Esto se debe a que en

Tabla 3: Tiempo de ejecución y memoria requerida para cada una de las estrategias implementadas en múltiples GPUs

Estrategia	# GPUs	Tiempo [Horas]	Memoria [GiB]	T_1/T_{NG}	Aceleración
1	1	10.58	8.64	100%	1×
	2	5.68	8.64	53,69%	1,86×
	3	3.61	8.64	34,12%	2,93×
	6	2.09	8.64	17,75%	5,63×
2	1	16.40	2.61	100%	1×
	2	8.78	2.61	53,54%	1,86×
	3	5.53	2.61	33,72%	2,96×
	6	3.32	2.61	18,24%	5,48×

la primera estrategia se guarda un campo y en la segunda estrategia sólo se guardan los datos en las fronteras del modelo.

Haciendo uso de múltiples GPUs se muestra un decremento casi lineal en función de la relación de la cantidad de disparos y las GPUs usadas (T_1/T_{NG}). La memoria requerida por cada GPU es indiferente de la cantidad que sean usadas en conjunto en la inversión.

5. RESULTADOS

5.1. Efectos de la dispersión numérica en la FWI 3D

De igual manera al análisis que se realizó para determinar la dispersión numérica en la sección 2.3 y su efecto en la propagación; en esta sección se muestra el efecto que tiene ésta sobre el modelo final obtenido en el proceso de inversión.

En este caso se usa como modelo sintético de prueba, el cubo difractor con una profundidad de 1700m, y dimensiones superficiales de 5250m y 2500m. Se analizan los resultados de la inversión cuando se presenta una dispersión numérica de 0,2% y 1,0%. El primer grupo de parámetros requiere usar un $dh = 25m$ y un modelo de $Nx = 211$, $Ny = 101$ y $Nz = 68$, con un $dt = 1,8ms$. El segundo conjunto de parámetros es $Nx = 131$, $Ny = 70$ y $Nz = 45$, con $dh = 45m$ y $dt = 3,0ms$. En ambas inversiones se plantea un enfoque multiescala usando 3 frecuencias (5Hz, 10Hz y 15Hz) y 30 iteraciones por cada paso de frecuencia. Se realiza la implementación en una sola GPU usando el método del gradiente descendente y la primera estrategia de implementación (sección 4.2).

Los modelos finales obtenidos se muestran en la Figura 33, donde para efectos de visualización se elimina el valor de la capa constante de fondo, para mostrar la reconstrucción del cubo difractor en el interior de los dos modelos. El modelo de la izquierda es el obtenido con un menor valor de dispersión numérica, mostrando un cubo más definido aunque con un nivel mayor en la huella de las fuentes en la reconstrucción.

Un corte en dirección del eje x en la mitad del modelo final de cada experimento se aprecia en la Figura 34. La Figura superior denota la inversión con 0,2% de dispersión numérica y permite ver de manera más clara una mejor reconstrucción en la parte interna del modelo, mientras que en la grafica inferior, aunque logra identificar la zona de actualización, no es capaz de reconstruir adecuadamente esta zona.

Se compara los valores de velocidad alcanzados por cada modelo respecto al valor real del modelo original (ver Figura 35) usando un perfil de velocidad ubicado en el centro de la superficie de los modelos. Esta gráfica muestra que el modelo con mayor dispersión presenta una mayor dificultad para la reconstrucción del cubo, dado que para la misma cantidad de iteraciones solo alcanza un valor pico de $2232 m/s$ en comparación de su contra-parte de $2485 m/s$

Figura 33: Efectos de la dispersión numérica en el modelo final de velocidad .

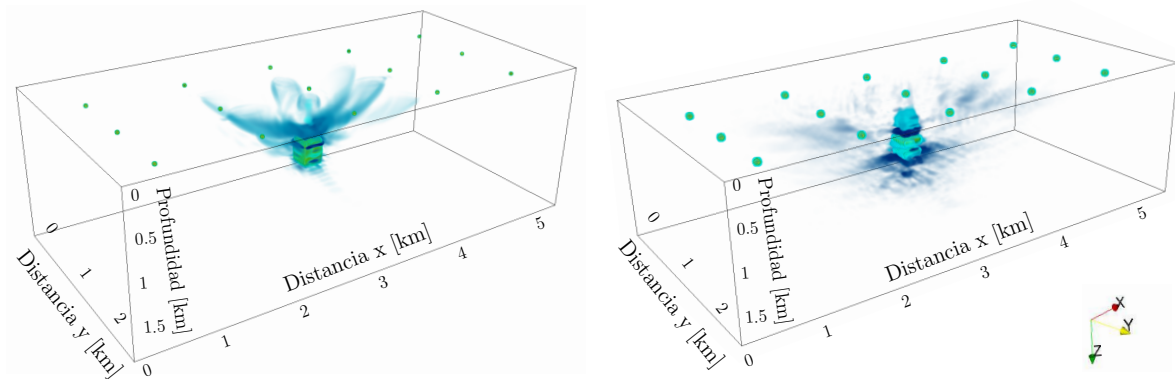
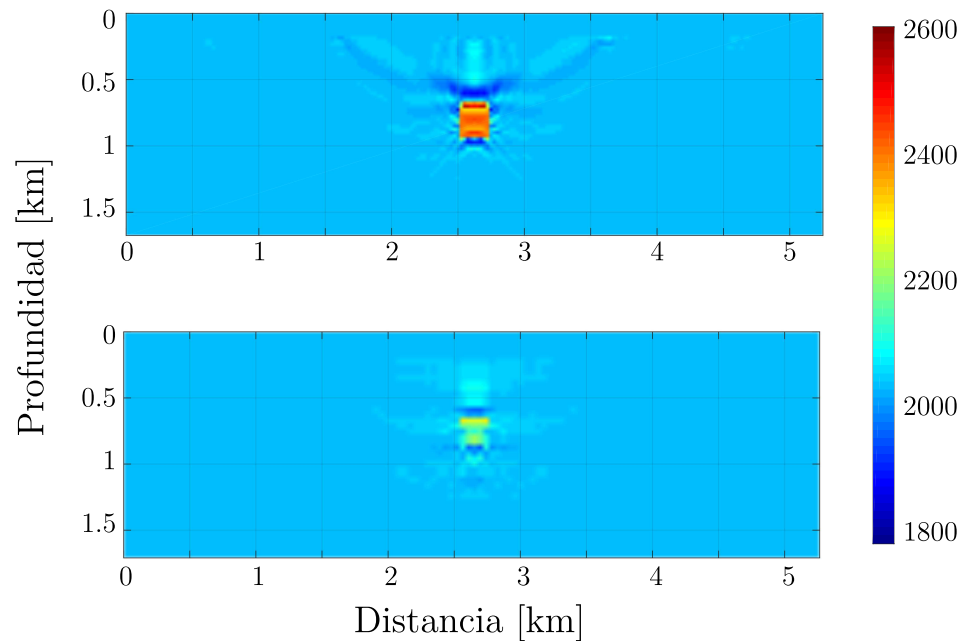


Figura 34: Sección 2D del modelo final de velocidad obtenido usando una dispersión numérica de 0,2% y 1%.

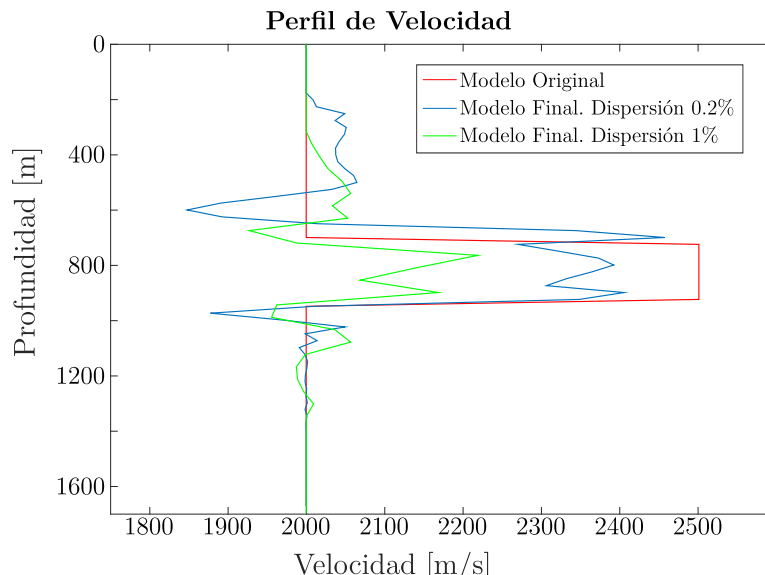


La Tabla 4 muestra los valores de RAM en cada implementación y comparados con dos valores de dispersión numérica y el valor teórico de cada experimento.

Tabla 4: Dispersión numérica vs Memoria requerida

RAM \ % dispersión numérica	1.0 %	0.2 %
Memoria Teórica [GiB]	2.23	8.56
Memoria Real [GiB]	2.15	8.64

Figura 35: Perfiles de velocidad del modelo original y de los modelos obtenidos usando dos diferentes valores de dispersión numérica.



Aunque ambos experimentos son numéricamente estables, usar un conjunto de parámetros que reduzca notablemente la memoria requerida no es aconsejable porque como se muestra en los resultados obtenidos se llega a obtener reconstrucciones erróneas, con artefactos inexistentes dentro de estos modelos. Se debe lograr un equilibrio entre un bajo valor de dispersión numérica pero sin elevar demasiado el costo computacional de la implementación.

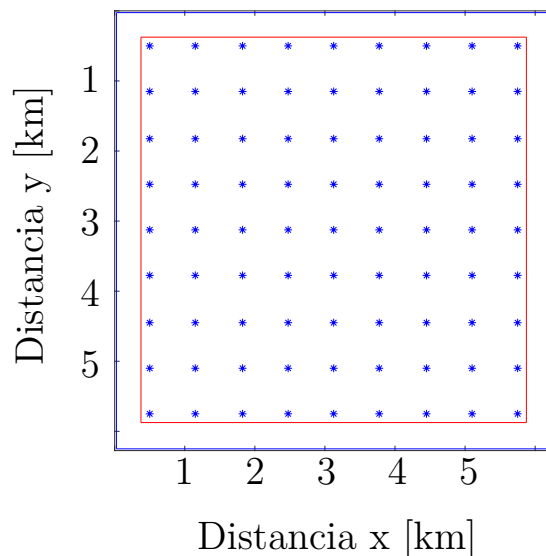
5.2. FWI 3D en modelos complejos

Se usan dos modelos sintéticos estructuralmente más complejos que el modelo del cubo-difractor, para validar la FWI 3D, llevándola a los límites de su implementación. Todas las FWI de esta sección son implementadas en multi-GPU usando la segunda estrategia de reducción de memoria, el método de L-BFGS y usando como punto de partida un modelo suavizado. Las pruebas son realizadas con un dispersión numérica de hasta 0,2 %.

5.2.1. Modelo sintético *overthrust*

Este modelo de velocidad tiene una profundidad de 3200 m, y dimensiones superficiales de 6250m \times 6250m. Se toma un paso espacial $dh = 25m$ lo que genera un modelo de $250 \times 250 \times 128$. El tiempo de grabación de los geófonos es de 3,15s con un tiempo de muestreo de 1,8ms. Se usan 81 fuentes en superficie y su disposición se detalla en la Figura 36.

Figura 36: Posición espacial de las fuentes en la superficie del modelo.



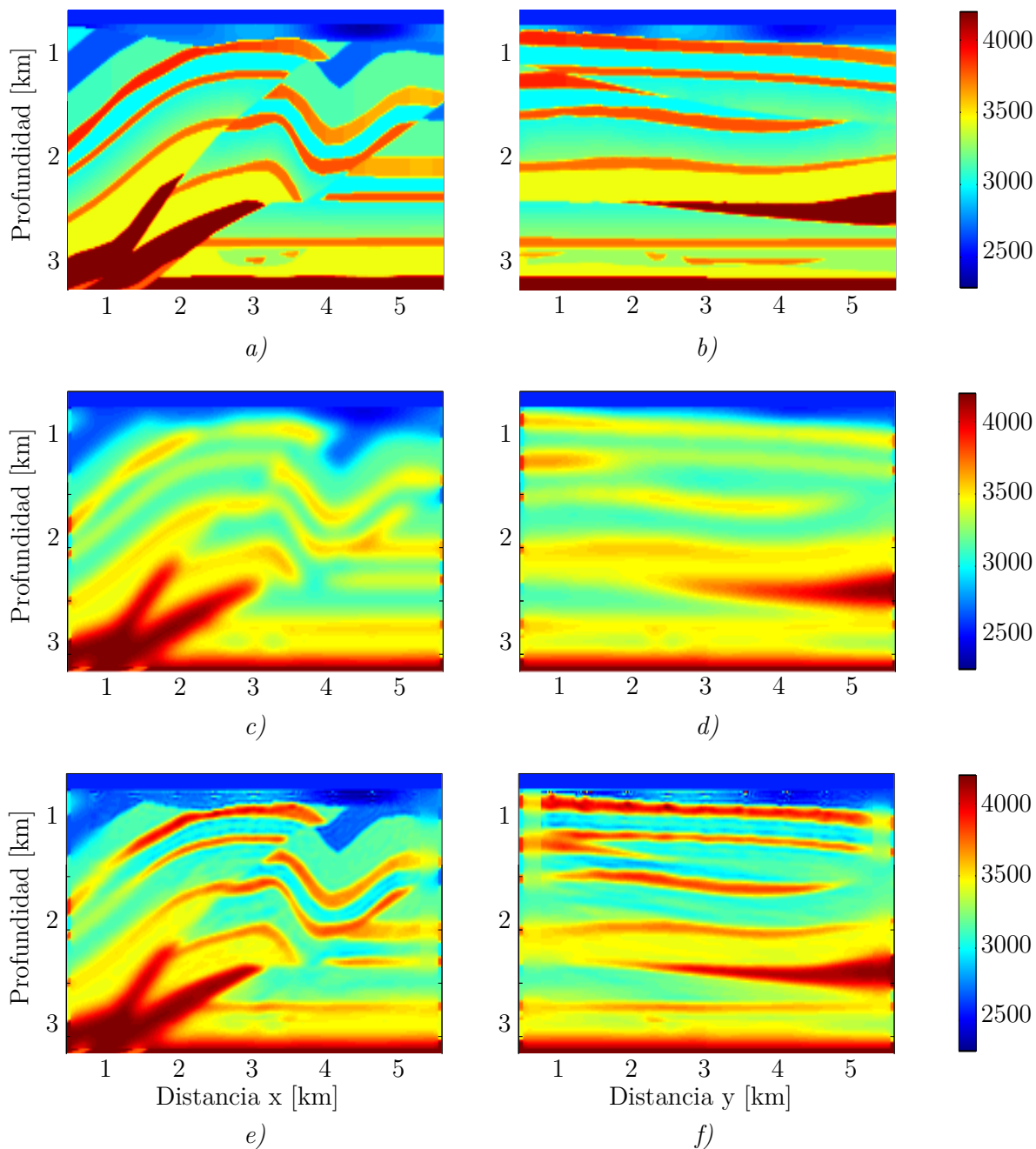
En la Figura 37 se detallan dos cortes del modelos centrados en superficie, en dirección del eje x (columna de la izquierda) y en dirección del eje y (columna de la derecha). Los cortes del modelo real se muestran en las Figura 37-a y 37-b. Para fijar el modelo de partida, se usa un filtro promediador de $7 \times 7 \times 7$, y se suaviza el modelo 5 veces. El modelo inicial en los mismos cortes de la gráfica anterior se muestran en la Figura 37-c y 37-d.

La inversión se realiza con el enfoque multiescala con frecuencias de 3Hz, 6Hz y 10Hz, usando un $\alpha_0 = 7000$ con 15 iteraciones para cada una de las frecuencias. El modelo final obtenido se visualiza en la Figura 37-e) y 37-f).

Visualmente se aprecia una mejora en el modelo final obtenido en comparación a su punto de partida, pero la métrica de salto de fase (CS) permite cuantificar esta mejora. Se realiza la propagación de una fuente puntual ubicada en el centro de la superficie del modelo original, inicial y final. Con el conjunto de trazas agrupadas por disparo común asociado a cada propagación se hacen las gráficas de CS. En la Figura 38 se muestra una comparación entre el desfase de las trazas inicial-original y final-original. Se toma un corte centrado en superficie en dirección del eje x para mostrar el efecto antes y después de realizar la FWI 3D.

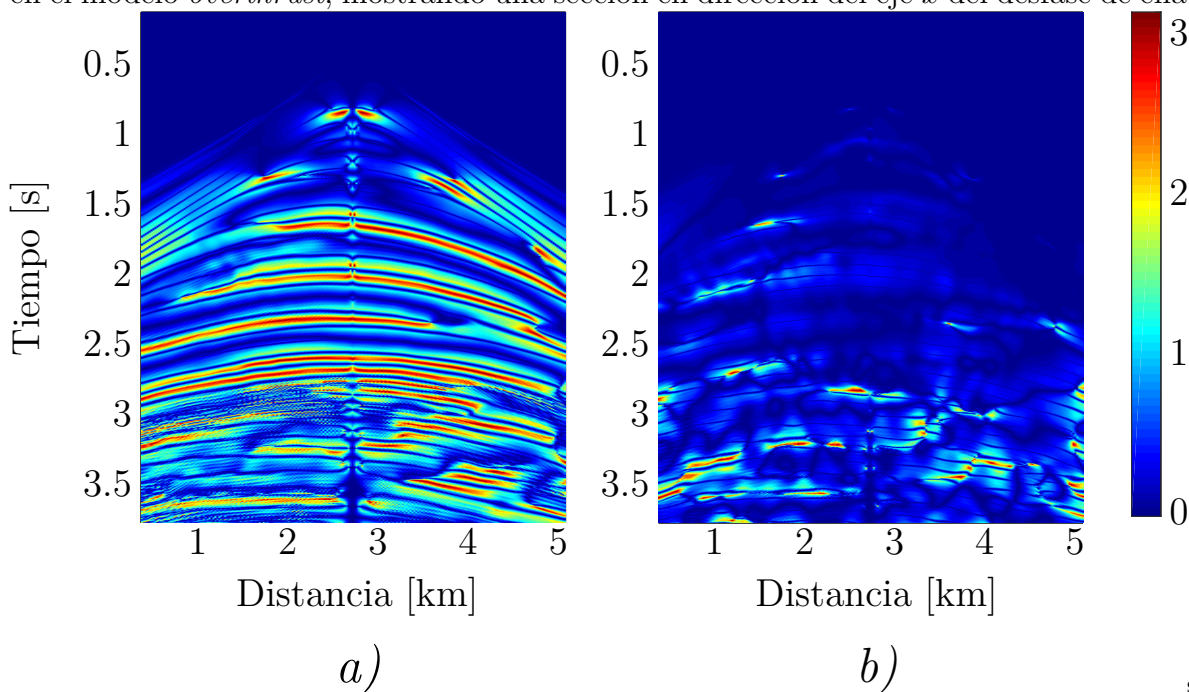
Esta comparación deja más claro el resultado positivo de la implementación de la FWI 3D. El efecto total de la inversión se muestra en la Figura 39, mostrando el acumulado de las veces que hubo CS por traza; valores proyectados a la superficie del modelo. El modelo inicial presenta un porcentaje de CS de 15,45 % (Figura 39-a) y el modelo final obtenido un valor de 1,7 % (Figura 39-b).

Figura 37: Secciones centradas en dirección del eje x y del eje y en el proceso de inversión para el modelo *overthrust* (a, b) usando un modelo suavizado como punto de partida (c, d), obteniendo el modelo final mostrado en e) y f).



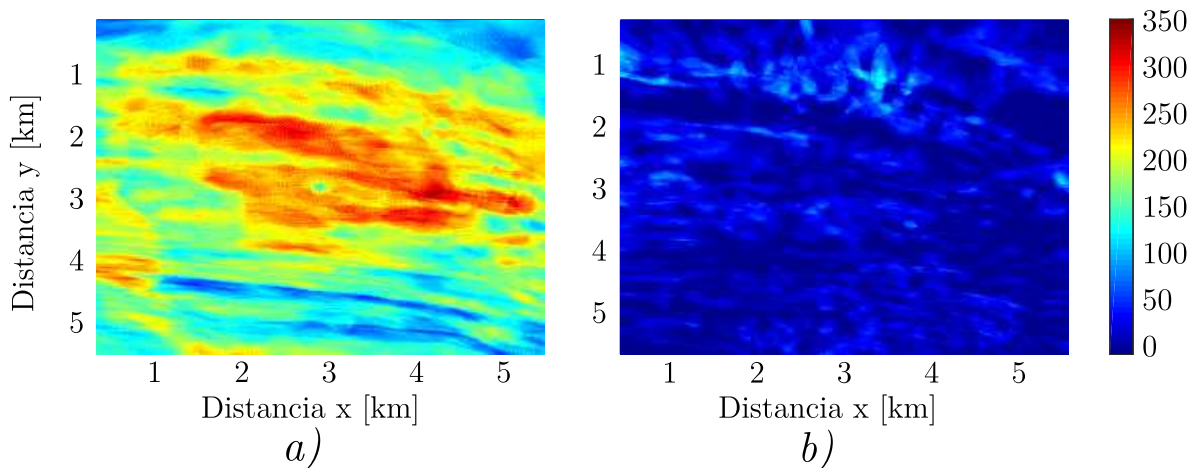
Se toma un perfil de velocidad para mostrar y comparar la reconstrucción del modelo obtenido en la FWI 3D (Figura 40). Luego de los tres pasos de frecuencia, el modelo

Figura 38: Desfase del conjunto de trazas del modelo inicial (a) y modelo final (b) en comparación al dato observado para una fuente centrada en superficie en el modelo *overthrust*, mostrando una sección en dirección del eje x del desfase de ella



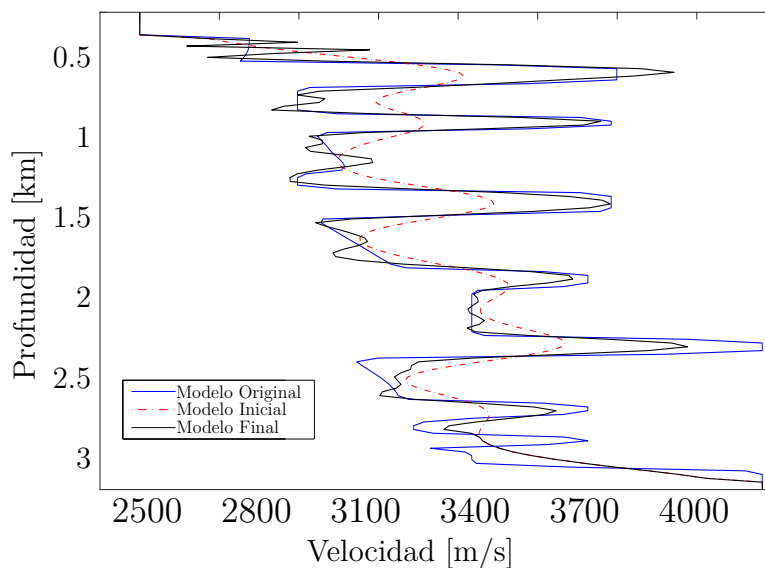
s.

Figura 39: Comparación del salto de fase modelo inicial (a) y modelo final (b) de la FWI 3D para una propagación de una fuente centrada en superficie en el modelo *overthrust*, con un CS de 15,47 % (a) y 1,7 % (b).



final (línea negra) logra seguir y acercarse a los verdaderos valores de velocidad (línea azul), partiendo de valores alejados y de contraste suavizado (línea roja) excepto en la zona inferior del modelo, dado que esta zona se encuentra dentro del CPML, por lo que no se toma en cuenta los últimos 400m aproximadamente.

Figura 40: Perfiles de velocidad del modelo *overthrust*, modelo inicial y modelo final obtenido en la FWI 3D.



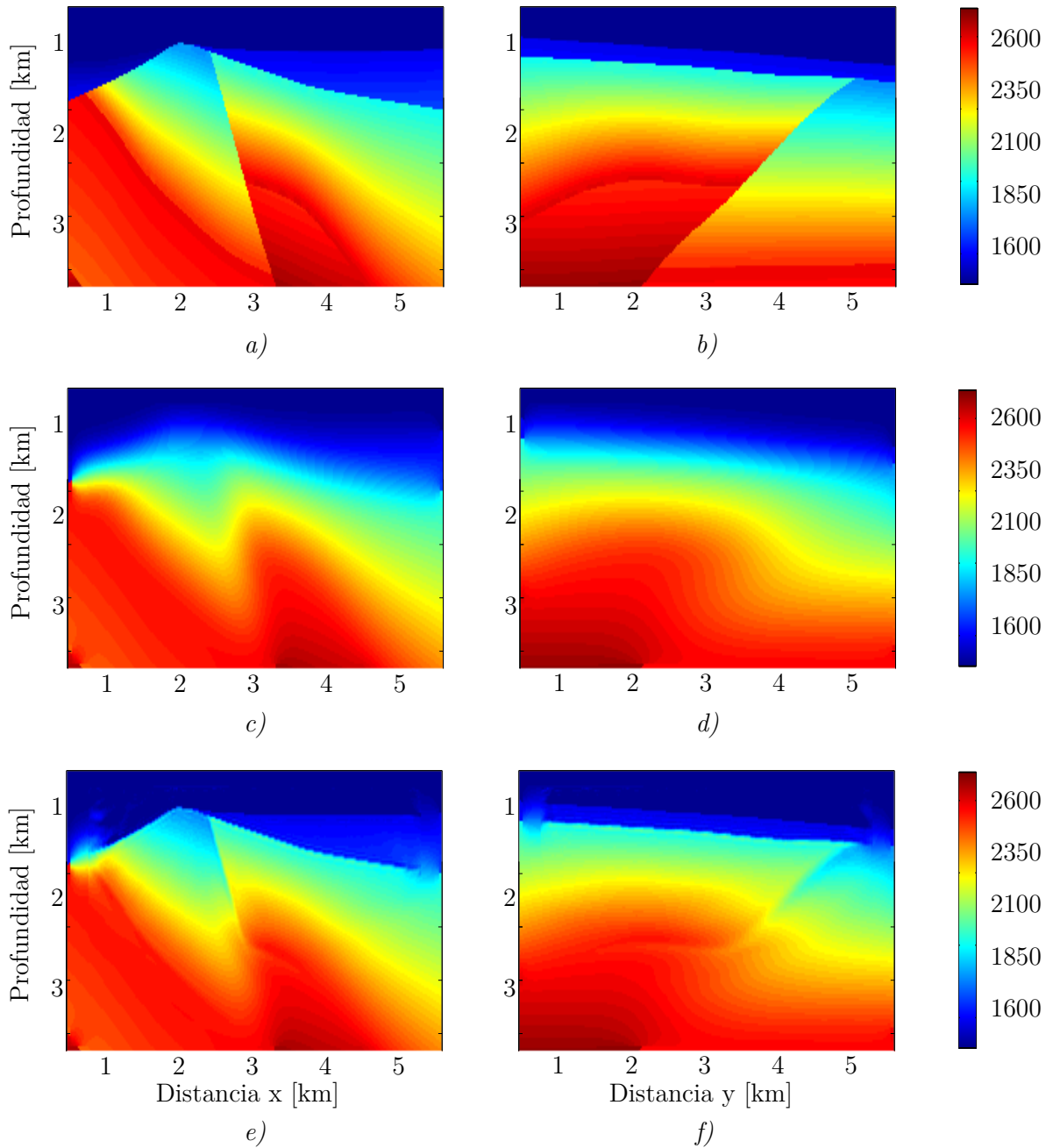
5.2.2. Modelo sintético *Fuerte*

Este modelo de velocidad tiene una profundidad de 3375m, y dimensiones superficiales de 6250m \times 6250m. Se toma un paso espacial $dh = 25m$ lo que genera un modelo de $250 \times 250 \times 135$. Es ligeramente más grande que el modelo anterior. Se emplea un tiempo mayor de grabación con 3,6s manteniendo el tiempo de muestreo, $dt = 1,8ms$.

El modelo real se detalla en la Figura 41-a y 41-b, con dos cortes centrados en direcciones del eje x y del eje y , respectivamente. En este caso el punto de partida aunque se aplica el mismo filtro promediador de $7 \times 7 \times 7$, en esta ocasión el proceso de suavizado se realiza 20 veces. El resultado se muestra en la Figura 41-c y 41-b en los mismo cortes del modelo real. Se aprecia que la falla geológica en el centro de las figuras es casi enteramente removida y reemplazada por un suave cambio de velocidades.

El resultado final es obtenido con un enfoque multiescala de 3 frecuencias (3, 6 y 10Hz), con 20 iteraciones por paso de frecuencia y con $\alpha_0 = 7000, 5000$ y 2000 respectivamente. El modelo obtenido por la FWI 3D se muestra en la Figura 41-e y 41-f. Esta inversión reconstruye en muy buena medida la capa somera, pero presenta mayor dificultad en la zona profunda del modelo.

Figura 41: Secciones centradas en dirección del eje x y del eje y en el proceso de inversión para el modelo *fuerte* (a, b) usando un modelo suavizado como punto de partida (c, d), obteniendo el modelo final mostrado en e) y f).



Se realiza el mismo análisis de CS que se hizo con el modelo anterior, con una fuente centrada en superficie. En la Figura 42 se muestra un corte en dirección del eje x del desfase del conjunto de trazas antes y después de la inversión.

Figura 42: Desfase del conjunto de trazas del modelo inicial (a) y modelo final (b) en comparación al dato observado para una fuente centrada en superficie en el modelo *fuerte*, mostrando una sección en dirección del eje x del desfase de ellas.

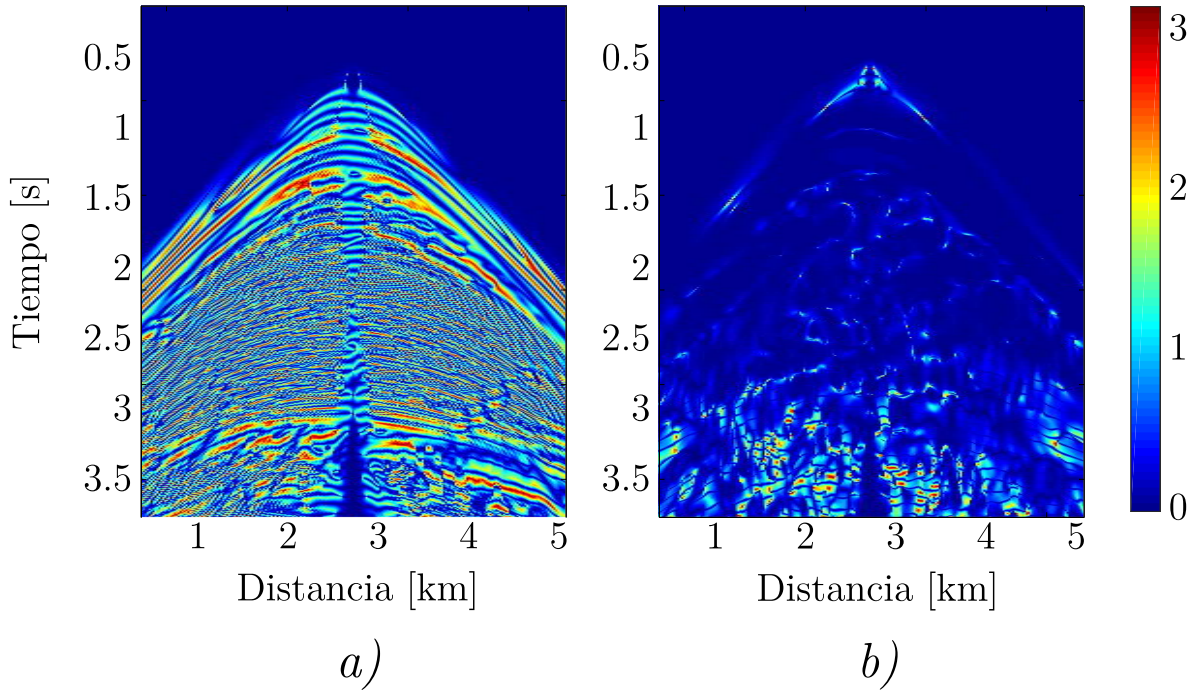
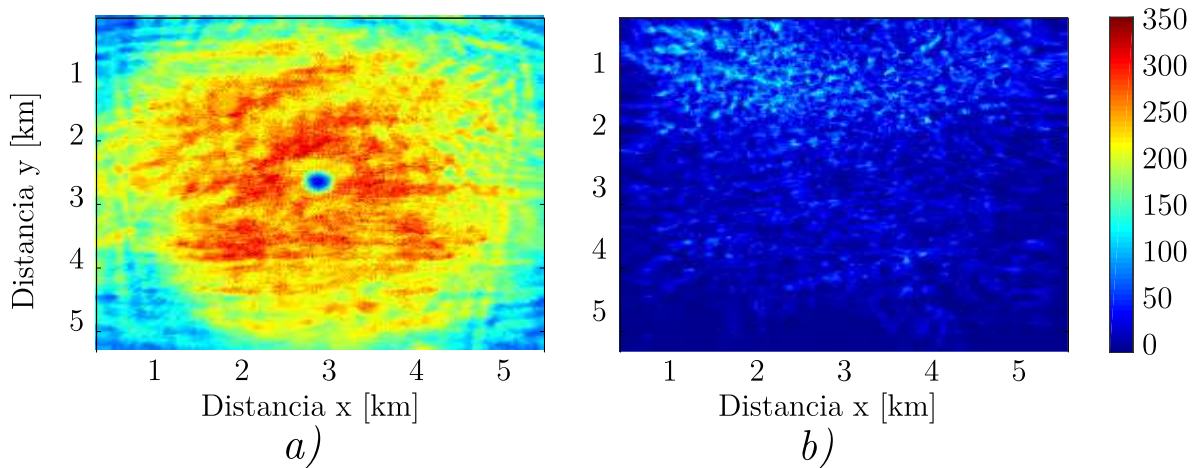


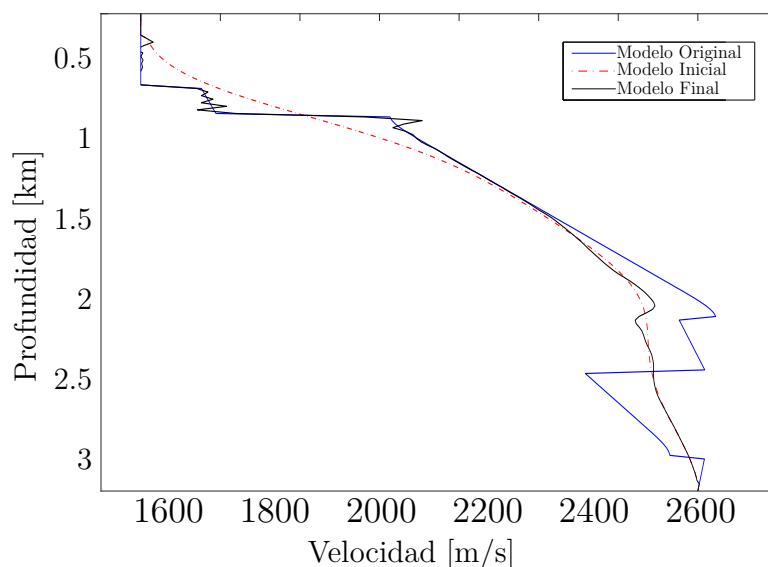
Figura 43: Comparación del salto de fase modelo inicial (a) y modelo final (b) de la FWI 3D para una propagación de una fuente centrada en superficie en el modelo *fuerte*, con un CS de 22,53 % (a) y 2,87 % (b).



Este resultado muestra que el modelo obtenido es mucho mejor que el modelo de partida. En la Figura 43 se detalla el valor de CS proyectado en la superficie del modelo en la posición de las trazas. El porcentaje de CS para el modelo de partida y el modelo final es 22,53 % (Figura 43-a) y 2,87 % (Figura 43-b), respectivamente.

Finalmente se muestra una comparativa en un perfil de velocidades en profundidad centrado en superficie, para los tres modelos, mostrando un seguimiento del valor de velocidad esperado hasta aproximadamente los 2km de profundidad (Figura 44).

Figura 44: Perfiles de velocidad del modelo *fuerte*, modelo inicial y modelo final obtenido en la FWI 3D.



Para cada uno de los modelos planteados, la memoria usada se muestra en la Tabla 5.

Tabla 5: Memoria requerida en modelos de velocidad complejos usando la estrategia 2 de implementación.

Modelo de velocidad	Memoria
<i>Overthrust</i>	8,77[GiB]
<i>Fuerte</i>	10,85[GiB]

La cantidad de memoria mostrada en la Tabla 5 hace notorio que incluso usando la estrategia que menos memoria consume, aún es necesario plantear estrategias adicionales para reducir en mayor medida estos valores que permiten hacer viable la implementación de la FWI 3D en un caso real.

6. DISCUSIÓN Y CONCLUSIONES

El núcleo para realizar la implementación de la inversión de onda completa en el dominio temporal, es la parte que se encarga de modelar la propagación de la onda de presión a través del medio. Los parámetros asociados al modelado (frecuencia, paso temporal y espacial, tiempo de adquisición, dimensiones del modelo, etc) definen la cantidad de memoria requerida para realizar la FWI 3D y su viabilidad en la implementación en una GPU. Así mismo, estos valores determinan la presencia o no de inestabilidad en el proceso de propagación y un aspecto fundamental como lo es la dispersión numérica.

Los parámetros no sólo se deben elegir para cumplir con la restricción de la inestabilidad numérica (ecuación (2.25)), también deben reducirse los valores de dispersión numérica presentes en el modelado, sin superar los requisitos de *hardware*. Un balance es necesario entre estos conceptos al momento de realizar la implementación y determinar si es viable la aplicación en algún modelo determinado. Los parámetros que determinan un cierto valor de dispersión numérica están estrictamente relacionados con el tipo de aproximación usada en la FDTD. La aproximación usada en este trabajo es de octavo orden para las derivadas espaciales y de segundo orden para las derivadas temporales. En este trabajo de investigación se realiza el análisis para el comportamiento de la dispersión y como esta afecta en el proceso final de reconstrucción.

Un valor de 0,2% de dispersión numérica en el modelado de la propagación de onda es aceptable en el proceso de inversión. Con este valor se obtienen resultados favorables para cada uno de los modelos implementados en este proyecto (cubo difractor, *overthrust*, *fuerte*). Una dispersión numérica del 1% genera reflexiones inexistentes dentro del modelado como se mostró en la sección 2.3, y debido a esta dispersión se afecta la inversión reconstruyendo “artefactos” que no están presentes en el modelo real, pero que permiten el decremento en la función objetivo. La reducción del porcentaje de dispersión numérica de 1% a 0,2% implica un aumento en cerca de 4 veces la cantidad de memoria requerida para la inversión.

Se hace necesario plantear un análisis de sensibilidad que permita relacionar la dispersión numérica en el modelado con los recursos de *hardware* necesarios para el almacenamiento de la información. Con este análisis se podría determinar un balance que haga viable definir los parámetros de una FWI 3D con el más bajo valor de dispersión numérica y la menor cantidad de memoria requerida para su implementación.

Para implementar la FWI es necesario elegir el tipo de método para realizar la búsqueda de un modelo que minimice la función de costo de la inversión. Se analiza

y se compara dos métodos de minimización, el gradiente descendente y el método de L-BFGS.

La implementación del gradiente descendente es más sencilla, dado que su algoritmo se basa únicamente en el cálculo del gradiente del modelo actual. Pero esta método no aporta información acerca de la curvatura en la función desajuste. Por otro lado, el método L-BFGS implica un algoritmo mucho más complejo, agregando propagaciones adicionales dentro del cada iteración para verificar la restricción del método (ecuación (3.16)) pero lo compensa con la obtención de un mejor modelo en cada iteración.

Estos métodos de minimización en el cubo difractor generan un porcentaje de salto de fase (CS) de 3,07% y 6,68% para L-BFGS y gradiente descendente, usando una fuente en una esquina de la superficie del modelo. De igual manera se obtiene un porcentaje de CS de 1,46% para L-BFGS y 12,26% para el gradiente descendente usando una fuente centrada en superficie. La memoria requerida en L-BFGS es hasta 5% superior en este modelo, dado la necesidad de almacenar la evolución de los últimos modelos de velocidad y sus gradientes. Este valor varía dependiendo de la cantidad de modelos y gradientes a guardar y de la estrategia implementada.

Teniendo una cantidad de tiempo fija para realizar la inversión, el método de gradiente descendente realiza más iteraciones (90 iteraciones) que el método de L-BFGS (60 iteraciones), dado que este último hace propagaciones adicionales (en el mejor de los casos una). Para todos los experimentos, L-BFGS siempre obtuvo un mejor modelo de velocidad que el obtenido usando el gradiente descendente dado que su porcentaje de CS siempre fue menor.

La inversión de onda completa 3D a nivel de ejecución, ve limitada su aplicación por la memoria requerida en el experimento sísmico y el tiempo de ejecución del algoritmo en GPU. Así, se define como estrategia de referencia la implementada en Abreo et al. [7] y alrededor de ella se construyen las estrategias para abordar estos problemas.

La primera estrategia requiere un 50% menos de memoria y su tiempo de ejecución es similar. Los resultados obtenidos no difieren salvo por el problemas asociados a la precisión numérica del dispositivo, dado que el orden en el que se calcula el gradiente es diferente, aunque usa los mismos factores. Estos errores son inferiores al 1×10^{-3} % del valor de referencia.

La segunda estrategia disminuye en mayor medida la cantidad de memoria requerida, haciendo uso de la reconstrucción de forma inversa del campo. Inicialmente se realiza la reconstrucción del campo propagado \mathbf{p} , pero al calcular la derivada asociada a este campo $\partial^2 \mathbf{p}$, los errores de la reconstrucción de la propagación se hacen más notorios y afectan el gradiente calculado, generando discrepancias con la implementación de referencia. Esta es la razón por la que se reconstruye el campo retropropagado λ , lo que implica una propagación adicional. Pero sólo requiere almacenar la información en las fronteras del modelo en lugar del campo total.

La memoria requerida por la segunda estrategia de implementación es solo un 15 % de la memoria de la implementación de referencia. Realizando una comparativa entre las estrategias implementadas, la primera es aproximadamente un 35 % más rápido que la segunda estrategia, pero este costo en tiempo lo compensa con la reducción de la memoria usada. La relación del costo de memoria entre la estrategia de referencia y la primera es constante, en cambio esta relación con la segunda estrategia varía en función del tamaño del modelo y los parámetros usados, teniendo un comportamiento inversamente proporcional; a mayor talla del modelo, menor es la relación de memoria entre la estrategia 2 y la de referencia.

Aunque la segunda estrategia presenta un decremento notorio en la RAM necesaria para la FWI 3D, es necesario realizar un análisis que permita medir e identificar el error numérico asociado a la reconstrucción del campo λ a partir de las fronteras y determinar el momento en que este error podría llegar a perjudicar el cálculo del gradiente y a su vez de la inversión.

Para reducir el tiempo de ejecución de la FWI 3D, se realiza la implementación en un cluster multi-GPU, aprovechando la independencia del orden de los disparos para calcular el gradiente total. La división del trabajo en el cluster se realiza en función de los disparos de la inversión, repartiendo equitativamente el total de los disparos entre todas las GPUs. Para un número elevado de disparos (> 60 , usando un cluster de 6 GPUs) se presenta un decremento casi lineal en el tiempo de ejecución en función de las GPUs usadas (Tabla 3).

Esta linealidad no se logra dar en la implementación dado que todo el proceso de inversión no se puede dividir equitativamente entre las GPUs. La actualización del modelo y el cálculo total del gradiente se realiza en una sola GPU, adicionalmente las transferencias de memoria entre las GPUs y el *host*, penalizan agregando tiempo en la inversión. La segmentación del trabajo y las transferencias se realizan a través del protocolo de comunicación MPI.

Se usan dos modelos para validar la implementación de la FWI 3D. El primero es un modelo sintético *overthrust* y el segundo es una sección de un modelo real de la costa colombiana llamado *fuerte*. Al realizar la inversión en estos modelos se obtiene una reducción de CS desde el punto de partida hasta el modelo final de velocidad de 15,47 % a 1,7 % en *overthrust* y una reducción de 22,53 % a 2,87 % en el caso de *fuerte*. Este CS es calculado usando una fuente centrada en la superficie de los modelos.

La ejecución del algoritmo de la FWI 3D gasta aproximadamente 7 días para la inversión de *overthrust* y 9 en la inversión de *fuerte*, usando el cluster y sus 6 GPUs exclusivamente para la implementación.

Realizando la inversión en modelos de mayor costo tamaño y complejidad sísmica se hace notorio que aún usando la estrategia que menos memoria requiere, los recursos de *hardware* quedan cerca del límite necesitando 11GiB para el modelo más grande (*fuerte*). En una inversión de un dato real con una talla de $960 \times 650 \times 420$ usando la

primera estrategia de implementación sería necesario tener disponible aproximadamente 4,8TiB de RAM para almacenar la información, mientras que la memoria requerida por la segunda estrategia es de 260GiB. Sin embargo, en la actualidad la GPU que ofrece la mayor cantidad de RAM es la GPU K80 con 24GiB de RAM, lo cual solo permitiría almacenar aproximadamente una décima parte de lo que se requeriría.

Este ejemplo muestra que para hacer viable la inversión de un dato real usando GPUs, es necesario plantear estrategias adicionales, como la segmentación de dominio [27] en combinación a la reconstrucción del campo, que reduzcan en mayor medida la cantidad de RAM requerida por la FWI 3D.

Adicionalmente, al elevar la complejidad de la ecuación de propagación de onda, la memoria requerida por la FWI 3D aumenta notablemente. Llevar la inversión a un caso con densidad variable, la RAM necesaria usando la estrategia uno se mantendría similar, pero aumentaría aproximadamente 4 veces para la segunda estrategia. Agregar variaciones a esta ecuación como la presencia en el modelo de anisotropía VTI, HTI, TTI entre otras, implicaría una mayor cantidad de RAM.

A. CAPAS CONVOLUCIONADAS PERFECTAMENTE ACOPLADAS

Según Pasalic en [10], las variables que minimizan las reflexiones en las fronteras no naturales, vienen dadas por

$$\psi_q = b_q \psi_q + a_q \frac{\partial \mathbf{p}}{\partial q}, \quad (\text{A.1})$$

$$\zeta_q = b_q \zeta_q + a_q \left(\frac{\partial^2 \mathbf{p}}{\partial q^2} + \frac{\partial \psi_q}{\partial q} \right), \quad (\text{A.2})$$

donde q es x , y o z que representa las variables espaciales del modelo. Las variables ψ_q y ζ_q se calculan iterativamente en cada paso de tiempo, con $\psi_q = \mathbf{0}$ y $\zeta_q = \mathbf{0}$ para la primera iteración. Los coeficientes b_q y a_q determinan la atenuación que generan las variables auxiliares y son calculados a partir de la zona del modelo, la frecuencia de la fuente y la velocidad del medio. Antes de mostrar como se calculan los vectores b_q y a_q , se muestran un par de parámetros necesarios definidos en [28]:

$$\begin{aligned} R &= 0,001, \\ L_q &= att \cdot \Delta q, \\ d_0 &= \frac{-3}{2L_q} \log(R), \end{aligned}$$

y un un vector F_q como:

$$F_q = \begin{cases} L_q : \Delta q : 0 & q \in [0, att) \\ 0 & q \in [att, Nq - att) \\ 0 : \Delta q : L_q & q \in [Nq - att, Nq) \end{cases} \quad (\text{A.3})$$

donde att es la cantidad de puntos usados para realizar la atenuación; Δq y N_q son el paso espacial y la cantidad de puntos del modelo en dirección de la dimensión q , respectivamente. Esta definición se implementa para las dimensiones x y y , dado que se debe de tener en cuenta la atenuación en ambas direcciones. Para la dimensión z (profundidad) la atenuación solo se presenta en la zona inferior del modelo.

$$F_z = \begin{cases} 0 & z \in [0, Nz - att) \\ 0 : \Delta z : L_z & z \in [Nz - att, Nz) \end{cases} \quad (\text{A.4})$$

Usando L_q se calcula los vectores d_q , α_q ,

$$d_q = d_0 V_{max} \left(\frac{F_q}{L_q} \right)^2, \quad (\text{A.5})$$

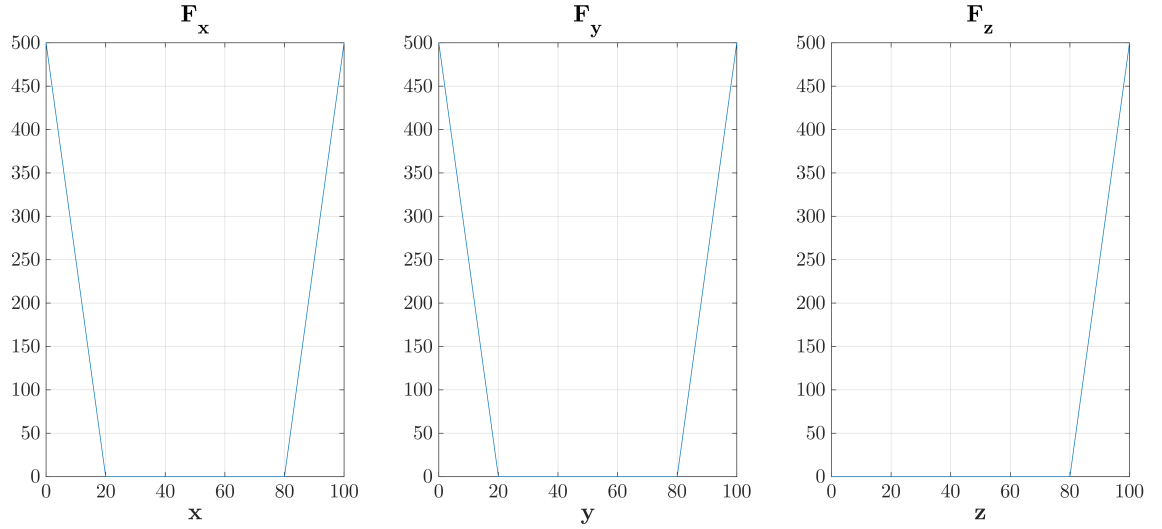
$$\alpha_q = \pi f \left(\frac{L_q - F_q}{L_q} \right), \quad (\text{A.6})$$

donde V_{max} es la velocidad máxima definida por la ecuación (2.25), y f es la frecuencia de la fuente $s(x, y, z, t)$. Finalmente b_q y a_q se calculan como:

$$b_q = e^{-(d_q + \alpha_q)dt}, \quad (\text{A.7})$$

$$a_q = \frac{d_q}{d_q + \alpha_q} (b_q - 1). \quad (\text{A.8})$$

Figura 45: Vector F_q usado en CPML para la atenuación de las reflexiones no naturales para cada una de las direcciones de propagación.



De la Figura 45 a la Figura 48 se muestran el valor de los vectores F_q , d_q , b_q y a_q , para cada una de las variables espaciales, usando un modelo de 100x100x100 puntos con 20 puntos para realizar la atenuación y una frecuencia de 5 Hz. Un paso espacial de 25 m, un paso temporal de 2 ms y una velocidad máxima de 5625 m/s.

La ecuación de onda acústica con densidad constante usando FDTD sin incluir las fronteras absorbentes es:

$$\mathbb{U}_{i,j,k}^{n+1} = 2 \cdot \mathbb{U}_{i,j,k}^n + \frac{(\mathbf{m}_{i,j,k} \cdot \Delta t)^2}{\Delta h^2} \left(\sum_{c=-4}^4 C_s[c] \cdot (\mathbb{U}_{i+c,j,k}^n + \mathbb{U}_{i,j+k+c}^n + \mathbb{U}_{i,j,k+c}^n) \right) - \mathbb{U}_{i,j,k}^{n-1}, \quad (\text{A.9})$$

donde $\mathbb{U}_{i,j,k}^n$ es el campo de presión discreto.

La ecuación (A.1), para cada una de las variables espaciales usando diferencias finitas centradas de octavo orden de aproximación es:

$$\psi_q = \begin{cases} \Psi_{\mathbf{x}(i,j,k)} = b_{x(i)} \Psi_{\mathbf{x}(i,j,k)} + a_{x(i)} \sum_{c=-4}^4 \left(\frac{1}{\Delta h} C_f[c] \cdot \mathbb{U}_{i+c,j,k}^n \right);, \\ \Psi_{\mathbf{y}(i,j,k)} = b_{y(j)} \Psi_{\mathbf{y}(i,j,k)} + a_{y(j)} \sum_{c=-4}^4 \left(\frac{1}{\Delta h} C_f[c] \cdot \mathbb{U}_{i,j+k+c}^n \right);, \\ \Psi_{\mathbf{z}(i,j,k)} = b_{z(k)} \Psi_{\mathbf{z}(i,j,k)} + a_{z(k)} \sum_{c=-4}^4 \left(\frac{1}{\Delta h} C_f[c] \cdot \mathbb{U}_{i,j,k+c}^n \right);, \end{cases} \quad (\text{A.10})$$

Figura 46: Vector d_q usado en CPML para la atenuación de las reflexiones no naturales para cada una de las direcciones de propagación.

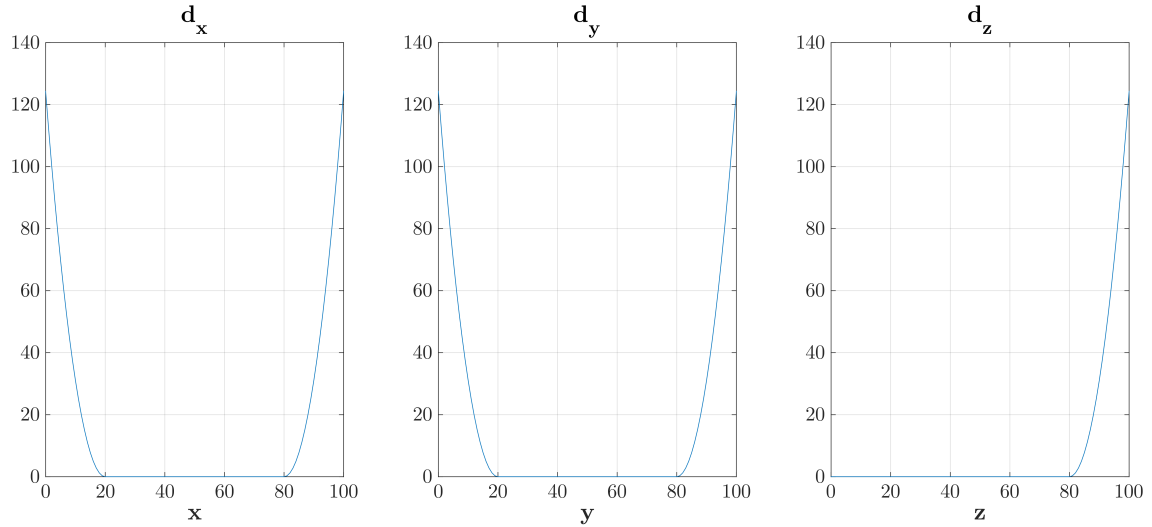
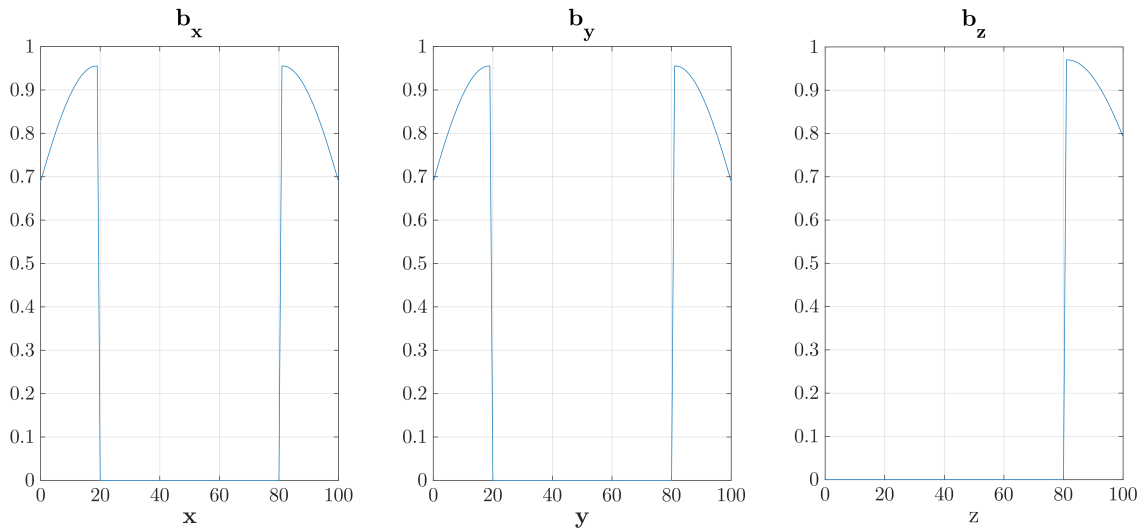
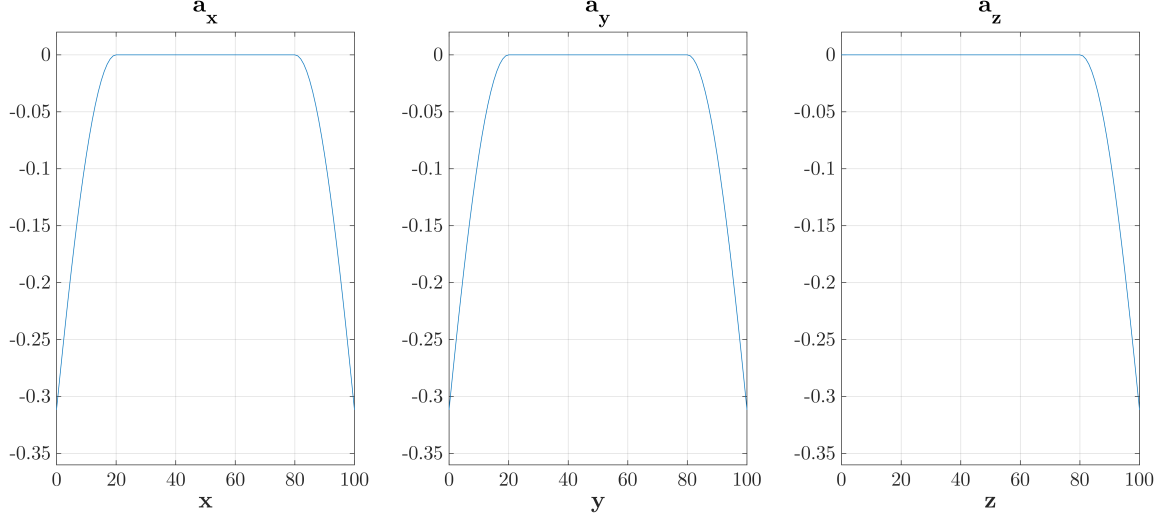


Figura 47: Valor de b_q usado en las variables auxiliares de las ecuaciones de CPML para la atenuación de las reflexiones no naturales para cada una de las direcciones de propagación.



donde $C_f = [\frac{1}{280}, \frac{-4}{105}, \frac{1}{5}, \frac{-4}{5}, 0, \frac{4}{5}, \frac{-1}{5}, \frac{4}{105}, \frac{-1}{280}]$ representa los coeficientes de la aproximación de octavo orden en diferencias finitas centradas para la derivada de primer orden; Las posiciones del vector C_f van desde -4 hasta 4 .

Figura 48: Valor de a_q usado en las variables auxiliares de las ecuaciones de CPML para la atenuación de las reflexiones no naturales para cada una de las direcciones de propagación



De igual manera se calcula la aproximación por diferencias finitas para la ecuación (A.2)

$$\zeta_q = \begin{cases} \mathbf{Z}_{\mathbf{x}(i,j,k)} = b_{x(i)} \mathbf{Z}_{\mathbf{x}(i,j,k)} + a_{x(i)} \sum_{c=-4}^4 \left(\frac{1}{\Delta h^2} C_s[c] \cdot \mathbb{U}_{i+c,j,k}^n + \frac{1}{\Delta h} C_f[c] \cdot \Psi_{\mathbf{x}(i+c,j,k)} \right); \\ \mathbf{Z}_{\mathbf{y}(i,j,k)} = b_{y(j)} \mathbf{Z}_{\mathbf{y}(i,j,k)} + a_{y(j)} \sum_{c=-4}^4 \left(\frac{1}{\Delta h^2} C_s[c] \cdot \mathbb{U}_{i,j+c,k}^n + \frac{1}{\Delta h} C_f[c] \cdot \Psi_{\mathbf{y}(i,j+c,k)} \right); \\ \mathbf{Z}_{\mathbf{z}(i,j,k)} = b_{z(k)} \mathbf{Z}_{\mathbf{z}(i,j,k)} + a_{z(i)} \sum_{c=-4}^4 \left(\frac{1}{\Delta h^2} C_s[c] \cdot \mathbb{U}_{i,j,k+c}^n + \frac{1}{\Delta h} C_f[c] \cdot \Psi_{\mathbf{z}(i,j,k+c)} \right); \end{cases} \quad (\text{A.11})$$

donde $C_s = \left[\frac{-1}{560}, \frac{8}{315}, \frac{-1}{5}, \frac{8}{5}, \frac{-205}{72}, \frac{8}{5}, \frac{-1}{5}, \frac{8}{315}, \frac{-1}{560} \right]$ representa los coeficientes de la aproximación de octavo orden en diferencias finitas centradas para la derivada de segundo orden; Las posiciones del vector C_s van desde -4 hasta 4 .

Partiendo de la ecuación (A.9), se agregan los dos conjuntos de variables auxiliares que realizan la atenuación en las fronteras no naturales usando el operador de onda definido en la ecuación (2.9). La expresión en diferencias finitas en el dominio temporal de la ecuación (2.9) es:

$$\begin{aligned} \mathbb{U}_{i,j,k}^{n+1} = & 2 \cdot \mathbb{U}_{i,j,k}^n + \frac{(\mathbf{m}_{i,j,k} \cdot \Delta t)^2}{\Delta h^2} \left(\sum_{c=-4}^4 C_s[c] \cdot (\mathbb{U}_{i+c,j,k}^n + \mathbb{U}_{i,j+c,k}^n + \mathbb{U}_{i,j,k+c}^n) \right) \\ & + \frac{(\mathbf{m}_{i,j,k} \cdot \Delta t)^2}{\Delta h} \left(\sum_{c=-4}^4 C_f[c] \cdot (\Psi_{\mathbf{x}(i+c,j,k)} + \Psi_{\mathbf{x}(i,j+c,k)} + \Psi_{\mathbf{x}(i,j,k+c)}) \right) \quad (\text{A.12}) \\ & + (\mathbf{m}_{i,j,k} \cdot \Delta t)^2 \left(\mathbf{Z}_{\mathbf{x}(i,j,k)} + \mathbf{Z}_{\mathbf{y}(i,j,k)} + \mathbf{Z}_{\mathbf{z}(i,j,k)} \right) - \mathbb{U}_{i,j,k}^{n-1} + \mathbb{S}_{i,j,k}^n. \end{aligned}$$

El operador de onda final, que se implementa para modelar la propagación de onda en un medio acústico e isótropo usando diferencias finitas de octavo orden en espacio y segundo orden en tiempo, es el conjunto de ecuaciones acopladas (A.10), (A.11) y A.12, para calcular el campo de onda en el siguiente paso de tiempo.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Jones I, “Tutorial: Incorporating near-surface velocity anomalies in pre-stack depth migration models,” *First Break*, pp. 47–58, 2012.
- [2] Lailly P, “The seismic inverse problem as a sequence of beforestack inversions. conference on inverse scattering: Theory and applications.” pp. 206–220, 1983.
- [3] Métivier Ludovic, Brossier Romain, Mérigot Q, Oudet E, and Virieux Jean, “Overcoming cycle skipping in fwi - an optimal transport approach,” 05 2016.
- [4] Pan Wenyong, Innanen Kirs, Margrave Gary, and Keating Scott, “Mitigate cycle-skipping for full-waveform inversion by bandlimited impedance inversion and pocs,” 03 2016.
- [5] Boonyasiriwat, C., Valasek P. Routh P. Cao W. Schuster G. T. and Macy B., “An efficient multiscale method for time-domain waveform tomography,” *Geophysics*, vol. 74, 2009.
- [6] Hicks G. J. and Pratt R. G., “Reflection waveform inversion using local descent methods: Estimating attenuation and velocity over a gas-sand deposit,” *Geophysics*, vol. 66, no. 2, pp. 598–612, 2001.
- [7] Abreo S., Ramirez A., Abreo D., Reyes O., and Gonzales H, “A practical implementation of acoustic full waveform inversion on graphical processing units,” *Ciencia, Tecnología y Futuro*, vol. 6, no. 2, pp. 5–16, 2015.
- [8] Virieux J., Ribodetti A., Mercerat D., Chaljub E., Tago J., Coutant O., Brossier R., Garambois S., Operto S., Cruz-Atienza V., *et al.*, *Modelling Seismic Wave Propagation for Geophysical Imaging*. INTECH Open Access Publisher, 2012.
- [9] Sullivan Dennis M, *Electromagnetic simulation using the FDTD method*. John Wiley & Sons, 2013.
- [10] Pasalic Damir, McGarry Ray, *et al.*, “Convolutional perfectly matched layer for isotropic and anisotropic acoustic wave equations,” 2010.
- [11] Finkelstein Bezalel and Kastner Raphael, “Finite difference time domain dispersion reduction schemes,” *J. Comput. Phys.*, vol. 221, pp. 422–438, Jan. 2007.
- [12] Sen K. and Stoffa P.L., *Global Optimization Methods in Geophysical Inversion*. Advances in Exploration Geophysics, Elsevier Science, 1995.

- [13] Serrano Jheyston, “Evaluación de métricas en dominios transformados usadas en las metaheurísticas para generar un punto de partida favorable a la inversión de onda completa 2d,” Master’s thesis, Universidad Industrial de Santander, 2017.
- [14] Tarantola A, “Inversion of seismic-reflection data in the acoustic approximation,” *Geophysics*, vol. 48, no. 8, pp. 1259–1266, 1984.
- [15] Goldstein Allen A, “On newton’s method,” *Numerische Mathematik*, vol. 7, no. 5, pp. 391–393, 1965.
- [16] Pratt R. Gerhard, Shin Changsoo, and Hick G. J., “Gauss–newton and full newton methods in frequency–space seismic waveform inversion,” *Geophysical Journal International*, vol. 133, no. 2, p. 341, 1998.
- [17] Nocedal J. and Wright S. J., *Numerical Optimization*. New York: Springer, 2nd ed., 2006.
- [18] Plessix R.-E., “A review of the adjoint-state method for computing the gradient of a functional with geophysical applications,” *Geophysical Journal International*, vol. 167, no. 2, pp. 495–503, 2006.
- [19] Ma Yong and Hale Dave, “Quasi-newton full-waveform inversion with a projected hessian matrix,” *Geophysics*, vol. 77, no. 5, pp. R207–R216, 2012.
- [20] Liu D. C. and Nocedal J., “On the limited memory bfgs method for large scale optimization,” *Math. Program.*, vol. 45, pp. 503–528, dec 1989.
- [21] Abreo D., Ramirez A., and Abreo S, “A hybrid methodology for a 3d full waveform inversion in time domain using gpus,” *15th International Congress of the Brazilian Geophysical Society EXPOGEf*, 2017.
- [22] dos Santos Adriano W. G. and Pestana Reynam C., “Time-domain multiscale full-waveform inversion using the rapid expansion method and efficient step-length estimation,” *Geophysics*, vol. 80, no. 4, pp. R203–R216, 2015.
- [23] Wolfe P., “Convergence conditions for ascent methods,” *SIAM Review*, vol. 11, no. 2, pp. 226–235, 1969.
- [24] Y. Ma, *Waveform-based velocity estimation from reflection seismic data*. No. 1, Colorado School of Mines, 2012.
- [25] Tesla K40 GPU Accelerator - Nvidia, “www.nvidia.com/content/PDF/kepler/nvidia-tesla-k40.pdf.”
- [26] Gropp William, Hoefler Torsten, Thakur Rajeev, and Lusk Ewing, *Using advanced MPI: Modern features of the message-passing interface*. MIT Press, 2014.

- [27] Villarreal Alberto and Scales John A, “3d finite difference modeling via domain decomposition,” en *SEG Technical Program Expanded Abstracts 1996*, pp. 1231–1234, Society of Exploration Geophysicists, 1996.
- [28] Perez Carlos, *Two-dimensional near-surface seismic imaging with surface waves: alternative methodology for waveform inversion*. PhD thesis, MINES ParisTech, 2013.
- [29] “CUDA C Programming Guide.” <http://docs.nvidia.com/cuda/>, 2015. [Online; accedido 20-06-2016].

BIBLIOGRAFÍA

Abreo D., Ramirez A., and Abreo S, “A hybrid methodology for a 3d full waveform inversion in time domain using gpus,” *15th International Congress of the Brazilian Geophysical Society EXPOGEf*, 2017

Boonyasiriwat, C., Valasek P. Routh P. Cao W. Schuster G. T. and Macy B., “An efficient multiscale method for time-domain waveform tomography,” *Geophysics*, vol. 74, 2009

“CUDA C Programming Guide.” <http://docs.nvidia.com/cuda/>, 2015. [Online; accedido 20-06-2016]

dos Santos Adriano W. G. and Pestana Reynam C., “Time-domain multiscale full-waveform inversion using the rapid expansion method and efficient step-length estimation,” *Geophysics*, vol. 80, no. 4, pp. R203–R216, 2015

Finkelstein Bezalel and Kastner Raphael, “Finite difference time domain dispersion reduction schemes,” *J. Comput. Phys.*, vol. 221, pp. 422–438, Jan. 2007

Goldstein Allen A, “On newton’s method,” *Numerische Mathematik*, vol. 7, no. 5, pp. 391–393, 1965

Gropp William, Hoefler Torsten, Thakur Rajeev, and Lusk Ewing, *Using advanced MPI: Modern features of the message-passing interface*. MIT Press, 2014

Hicks G. J. and Pratt R. G., “Reflection waveform inversion using local descent methods: Estimating attenuation and velocity over a gas-sand deposit,” *Geophysics*, vol. 66, no. 2, pp. 598–612, 2001

Jones I, “Tutorial: Incorporating near-surface velocity anomalies in pre-stack depth migration models,” *First Break*, pp. 47–58, 2012

Lailly P, “The seismic inverse problem as a sequence of beforestack inversions. conference on inverse scattering: Theory and applications.” pp. 206–220, 1983

Liu D. C. and Nocedal J., “On the limited memory bfgs method for large scale optimi-

zation,” *Math. Program.*, vol. 45, pp. 503–528, dec 1989

Ma Yong and Hale Dave, “Quasi-newton full-waveform inversion with a projected hessian matrix,” *Geophysics*, vol. 77, no. 5, pp. R207–R216, 2012

Métivier Ludovic, Brossier Romain, Mérigot Q, Oudet E, and Virieux Jean, “Overcoming cycle skipping in fwi - an optimal transport approach,” 05 2016

Nocedal J. and Wright S. J., *Numerical Optimization*. New York: Springer, 2nd ed., 2006

Pan Wenyong, Innanen Kirs, Margrave Gary, and Keating Scott, “Mitigate cycle-skipping for full-waveform inversion by bandlimited impedance inversion and pocs,” 03 2016

Pasalic Damir, McGarry Ray, *et al.*, “Convolutional perfectly matched layer for isotropic and anisotropic acoustic wave equations,” 2010

Perez Carlos, *Two-dimensional near-surface seismic imaging with surface waves: alternative methodology for waveform inversion*. PhD thesis, MINES ParisTech, 2013

Plessix R.-E., “A review of the adjoint-state method for computing the gradient of a functional with geophysical applications,” *Geophysical Journal International*, vol. 167, no. 2, pp. 495–503, 2006

Pratt R. Gerhard, Shin Changsoo, and Hick G. J., “Gauss–newton and full newton methods in frequency–space seismic waveform inversion,” *Geophysical Journal International*, vol. 133, no. 2, p. 341, 1998

Sen K. and Stoffa P.L., *Global Optimization Methods in Geophysical Inversion*. Advances in Exploration Geophysics, Elsevier Science, 1995

Serrano Jheyston, “Evaluación de métricas en dominios transformados usadas en las metaheurísticas para generar un punto de partida favorable a la inversión de onda completa 2d,” Master’s thesis, Universidad Industrial de Santander, 2017

Sullivan Dennis M, *Electromagnetic simulation using the FDTD method*. John Wiley & Sons, 2013

Tarantola A, “Inversion of seismic-reflection data in the acoustic approximation,” *Geophysics*, vol. 48, no. 8, pp. 1259–1266, 1984

Villarreal Alberto and Scales John A, “3d finite difference modeling via domain decomposition,” en *SEG Technical Program Expanded Abstracts 1996*, pp. 1231–1234, Society

of Exploration Geophysicists, 1996

Virieux J., Ribodetti A., Mercerat D., Chaljub E., Tago J., Coutant O., Brossier R., Garambois S., Operto S., Cruz-Atienza V., *et al.*, *Modelling Seismic Wave Propagation for Geophysical Imaging*. INTECH Open Access Publisher, 2012

Wolfe P., “Convergence conditions for ascent methods,” *SIAM Review*, vol. 11, no. 2, pp. 226–235, 1969

Y. Ma, *Waveform-based velocity estimation from reflection seismic data*. No. 1, Colorado School of Mines, 2012