

**IMPLEMENTACIÓN DE UNA INVERSIÓN DE
ONDA COMPLETA(FWI) 2-D SOBRE UNA
ARQUITECTURA GPU**

DAVID LEONARDO ABREO CARRILLO

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS
FISICO-MECANICAS
ESCUELA DE INGENIERIA ELECTRICA,
ELECTRONICA Y DE TELECOMUNICACIONES
BUCARAMANGA
2015**

**IMPLEMENTACIÓN DE UNA INVERSIÓN DE
ONDA COMPLETA(FWI) 2-D SOBRE UNA
ARQUITECTURA GPU**

DAVID LEONARDO ABREO CARRILLO

**Trabajo de Grado para optar al título de
Ingeniero Electrónico**

**Directora
ANA BEATRIZ RAMÍREZ SILVA
PhD. en Ingeniería Eléctrica**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS
FISICO-MECANICAS
ESCUELA DE INGENIERIA ELECTRICA,
ELECTRONICA Y DE TELECOMUNICACIONES
BUCARAMANGA
2015**

A C U E R D O No. 164 DE 2003
(diciembre 16)

ARTÍCULO 2º. Derogar el Acuerdo Académico No. 080 del 27 de abril de 1999.

COMUNÍQUESE Y CÚMPLASE,

Expedido en Bucaramanga, a los dieciséis (16) días del mes de diciembre de 2003.

LA PRESIDENTA DEL CONSEJO ACADÉMICO,

LUCILA NIÑO BAUTISTA
Vicerrectora Académica

LA SECRETARIA GENERAL, LILIA AMANDA PATIÑO DE CRUZ

*En primer lugar a Dios por bendecirme y protegerme
cada día y permitirme terminar una fase 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.

Todo este trabajo ha sido posible gracias a ellos.

Agradecimientos

Quiero agradecer a mis padres y mi hermano por su constante apoyo, 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.

Agradezco al cuerpo docente de la escuela de Ingeniería Electrónica de la Universidad Industrial de Santander y a todos aquellos participantes en mi formación profesional, así como a mi Director y Codirector de Proyecto los docentes Ana Beatriz Ramírez y Sergio Alberto Abreo, por su apoyo incondicional y académico en el desarrollo de mi tesis para optar al título de Ingeniero Electrónico.

Un agradecimiento especial 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 compañeros Fabián Sánchez, Edgar Alfonso, Kevin Polo y Oscar Ribero y muchos más, grandes personas, grandes amigos, amigos para toda una vida. A Smith Tarazona, por estar apoyándome en aquellos momentos en que el estudio y el trabajo ocuparon mi tiempo y esfuerzo.

A todos, muchas Gracias!!!

David Leonardo Abreo Carrillo

Contenido

Introducción	13
1 Inversión de onda completa (FWI)	14
1.1 Marco Teórico	16
2 Inversión de Onda Completa de manera serial	21
2.1 Algoritmo de programación	21
2.2 Ecuación de onda acústica isotrópica y CPML	23
2.3 Diferencias finitas	25
2.3.1Resumen diferencias finitas	31
2.4 Implementación ecuación de onda acústica 2D	32
2.5 Implementando la FWI	33
2.6 Aplicando sobre un Modelo Geofísico complejo	36
3 Inversión de Onda Completa de manera paralela	40
3.1 Arquitectura de Cómputo de Propósito General en unidades de procesamiento gráfico (GPGPU)	40
3.2 Segmentando el código	42
3.3 Algoritmo de programación	43
3.4 Aplicación de la FWI en GPU sobre un modelo geofísico	44
4 Conclusiones	48
Bibliografía	50

Lista de figuras

1.1	Problema inverso teórico	15
1.2	Minimos locales asociados a los modelos de partida.	16
1.3	Propagación de una fuente (en rojo) a diferentes intervalos de tiempo utilizando la ecuación de onda. Los geófonos se muestran en azul y la zona límites se representan en verde.	19
1.4	Retropropagación de los residuales (en rojo) en diferentes intervalos de tiempo utilizando la ecuación de onda. La zona límites se representa en verde.	20
2.1	Vector $f(x)$	25
2.2	Valores a_x y b_x	26
2.3	Valores d_x	26
2.4	Stencil para ecuación de onda 2D fuera de la zona límite.	32
2.5	Stencil para ecuación de onda 2D en la zona límite.	32
2.6	Propagación,. Figuras de la izquierda sin CPML. Figuras de la derecha con CPML	34
2.7	<i>Shot gather</i> y su modelo asociado	35
2.8	Actualización del modelo de velocidades usando FWI	38
2.9	Evolución función objetivo $\phi(m)$ de una implementación serial	39
3.1	Modelo de propagación (68x210 puntos) usando bloques de 16x32 threads.	42
3.2	Distribucion de un bloque (16x32 threads).	43
3.3	Modelo de propagación (68x210 puntos) usando un área de CPML de 20 puntos para la frontera no natural.	43
3.4	Evolución función objetivo $\phi(m)$ de una implementación paralela	45
3.5	Actualización del modelo de velocidades usando FWI	46

Lista de Tablas

2.1	Arquitectura Intel Core i7-4700HQ	37
3.1	Arquitectura GPU GTX 660	47
3.2	Tiempo ejecución de kernel en GPU	47

RESUMEN

TÍTULO: IMPLEMENTACION DE UNA INVERSION DE ONDA COMPLETA (FWI) 2-D SOBRE UNA ARQUITECTURA GPU

AUTOR: DAVID LEONARDO ABREO CARRILLO**

PALABRAS CLAVES: Inversión de onda completa, método del estado adjunto, unidades de procesamiento gráfico, capas perfectamente emparejadas.

DESCRIPCIÓN:

Una de las principales necesidades en la exploración de hidrocarburos y gas es encontrar imágenes sísmicas confiables y de alta resolución adquiridas a partir de sensores en la superficie. La adquisición de estos datos requiere también de perfiles de velocidad exactos. Recientemente, el método de inversión de onda completa (FWI, por sus siglas en inglés), usado para encontrar modelos de velocidad confiables, ha ganado atención en la comunidad geofísica. La implementación del método FWI requiere una función de costo, una ecuación de onda para encontrar el campo de presión propagado hacia adelante y hacia atrás y un modelo de velocidad inicial, para estimar un modelo de velocidades en forma iterativa. Sin embargo, un aspecto no favorable para el método FWI, es el costo computacionalmente, en especial cuando se usa para encontrar modelos de velocidades en experimentos sísmicos 2D de largo offset o en experimentos sísmicos 3D.

Este trabajo aborda el problema del costo computacional del método, y presenta una propuesta de la implementación de la FWI en 2D en forma paralela sobre una arquitectura de propósito general en unidades de procesamiento gráfico (GPGPU), el cual utiliza la ecuación de onda acústica e isotrópica en dos dimensiones con densidad constante para modelar la propagación de la onda de presión en el perfil de velocidades. La implementación en paralelo usa el método del estado adjunto para estimar en forma iterativa el modelo de velocidades, el cual es obtenido cuando un número de iteraciones determinado es alcanzado o cuando se alcanza un nivel de error dado.

* Trabajo de grado

** Facultad de Ingenierías Físico-mecánicas. Escuela de Ingeniería Eléctrica, Electrónica y de Telecomunicaciones, Director PhD Ana Beatriz Ramírez Silva, Codirector MIE. Sergio Alberto Abreo Carrillo

ABSTRACT

TITLE: IMPLEMENTATION OF A FULL WAVEFORM INVERSION (FWI) 2-D IN A GPU ARCHITECTURE*

AUTHOR: DAVID LEONARDO ABREO CARRILLO **

KEY WORDS: Full Waverform Inversion, adjoint state method, graphics processing units, convolutional perfect matched layers.

DESCRIPTION:

One of the main needs in the oil and gas exploration is to find reliable seismic images and high resolution acquired from sensors on the surface. The acquisition of these data may require precise speed profiles. Recently, the method of Full Waveform Inversion (FWI), used to find reliable speed models, has gained attention in the geophysical community. FWI implementing the method requires a cost function, a wave equation to find the pressure field propagated forwards and backwards and initial velocity model to estimate a velocity model iteratively. However, an unfavorable for FWI aspect method is the computational cost, especially when used to find patterns in seismic velocity experiments long offset 2D or 3D seismic experiments.

This paper addresses the problem of the computational cost of the method, and presents a proposal for the implementation of the FWI 2D parallel architecture on a General-Purpose Graphics Processing Units (GPGPU), which uses the acoustic wave equation and two dimensional isotropic constant density to model the propagation of the pressure wave velocity profile. The parallel implementation uses the adjoint state method iteratively to estimate the velocity model, which is obtained when a certain number of iterations is reached or when a given error level is reached.

* Bachelor thesis

** Physical-Mechanical Engineering Faculty. Electrical Engineering, Electronics and Telecommunications School. Director PhD Ana Beatriz Ramirez Silva. Co-director MIE. Sergio Alberto Carrillo Abreo

Introducción

La FWI es un procedimiento de ajuste de datos no lineal que tiene como objetivo la obtención de estimaciones de las propiedades del subsuelo a partir de registros sísmicos, que pueden ser provenientes de adquisiciones de sísmica pasiva o activa. Dada una estimación inicial de los parámetros del subsuelo (un modelo), los datos se predicen mediante la solución de una ecuación de onda. Luego el modelo se actualiza, a fin de reducir el desajuste entre los datos observados y los predichos. Esto se repite de manera iterativa hasta que el error entre los datos es suficientemente pequeño o se hayan realizado un número de iteraciones [19].

Particularmente, el método FWI requiere una función de costo, una ecuación de onda para encontrar el campo de presión propagado hacia adelante y hacia atrás y un modelo de velocidad inicial, para estimar un modelo de velocidades en forma iterativa. A pesar de las ventajas de la FWI, el método es costoso computacionalmente. Además de su alto costo computacional, en especial cuando se usa para encontrar modelos de velocidades en experimentos sísmicos 2D de largo offset o en experimentos sísmicos 3D; el punto de partida del método también es una restricción. El método requiere una muy buena predicción del modelo inicial para lograr convergencia y se logre reducir la brecha entre el modelo calculado y el modelo real.

Realizando un enfoque en el costo computacional, este trabajo de grado presenta una propuesta de la implementación del método FWI 2D en forma paralela sobre una GPU, el cual utiliza la ecuación de onda acústica e isotrópica para modelar la propagación de la onda de presión, usando ecuaciones diferenciales de segundo orden, mediante el método del estado adjunto [12]; para estimar en forma iterativa el modelo de velocidades, el cual es obtenido cuando un número de iteraciones es alcanzado o cuando se alcanza un determinado nivel de error.

Capítulo 1

Inversión de onda completa (FWI)

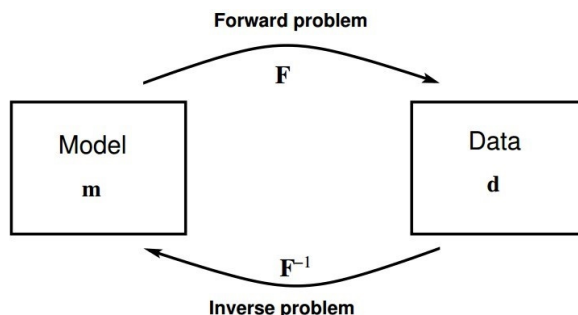
Durante los últimos 20 años, la inversión de onda completa ha estado en transición desde un enfoque defendido y usado casi exclusivamente por las instituciones académicas a ser un habitual método empleado en exploración sísmica por empresas de procesamiento de datos sísmicos en todo el mundo. A medida que logramos entender y aplicar las complejas inversiones 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. Así se incrementó la capacidad de enfoque para superar a otras técnicas de determinación de modelos de velocidades, como por ejemplo, la tomografía basada en rayos en términos de la resolución de pequeñas estructuras complejas en tierra [5].

Desde que un algoritmo FWI iterativo basado en gradiente se introdujo por primera vez a la comunidad geofísica por Lailly [7], muchas estrategias y esquemas computacionales se han desarrollado, desde implementación en el dominio del tiempo y de la frecuencia, hasta herramientas prácticas para el procesamiento sísmico 3D [15].

Básicamente, la inversión de onda completa [17] utiliza los datos sísmicos adquiridos \mathbf{d} para estimar parámetros de un modelo del subsuelo \mathbf{m} , minimizando la diferencia entre los datos registrados y los datos sintéticos $F(\mathbf{d})$, donde F es un operador no lineal. (Ver figura 1.1). En otras palabras, FWI minimiza las diferencias entre los datos registrados y los datos generados. Este enfoque distingue FWI de otros métodos.

La FWI es atractiva por su capacidad para estimar modelos del subsuelo

Figura 1.1: Problema inverso teórico



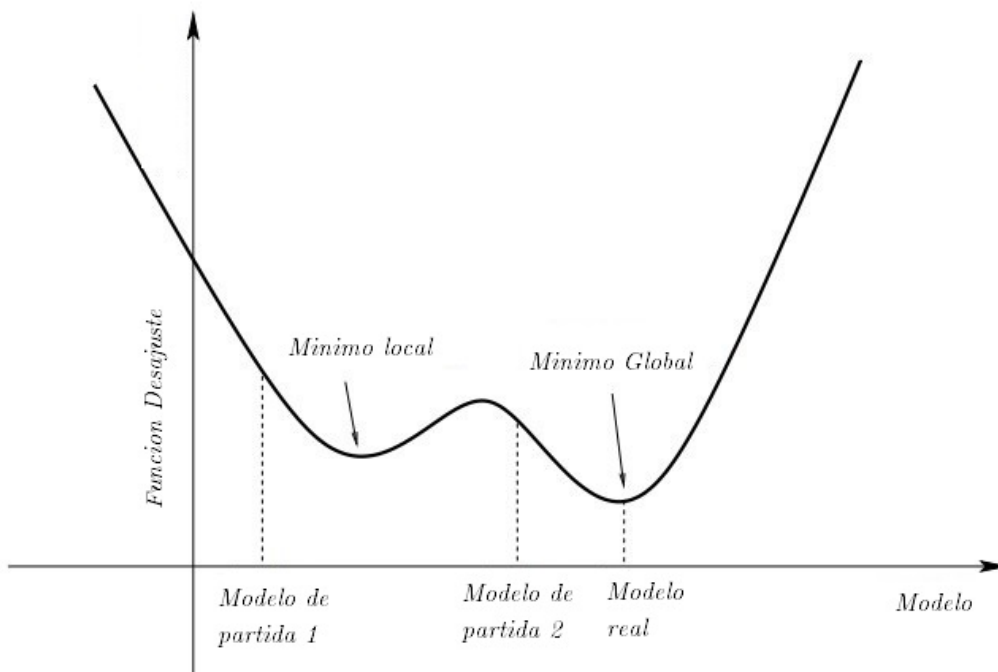
generalmente con mayor resolución [10] que muchos otros métodos, como la tomografía de tiempo de viaje y análisis de la velocidad de migración (MVA) [13].

La FWI requiere un gran número de iteraciones para minimizar el desajuste entre los datos, donde el costo computacional es proporcional al número de fuentes usadas, y al número de iteraciones requeridas. Por lo tanto, varios métodos se han aplicado para reducir el coste computacional. Uno de estos métodos consiste en aplicar técnicas de codificación de fase [6], en el que se combinan todos los disparos juntos para formar una fuente simultánea. El costo de FWI, utilizando la codificación de ese modo, se reduce por un factor idealmente igual al número de disparos codificados dividido por el número de disparos grabados. Para reducir el número de iteraciones requeridas, se puede usar una escasa representación de un modelo de espacio y reducir el número de parámetros del modelo. En la inversión, por lo tanto, uno debe elegir algunas muestras de lo posible para construir un modelo disperso, sin dejar de mantener la mayor cantidad de características geológicas como sea posible. La transformada wavelet es una técnica representativa utilizada en problemas inversos para reducir el número de parámetros [8].

Debido que FWI resuelve un problema inverso indeterminado, diferentes modelos pueden entrar en una región de modelos igualmente validos con un margen de tolerancia razonable. Este problema de no unicidad es causado principalmente por mínimos locales en una función desajuste entre los datos (ver figura 1.2), y la presencia de los mínimos locales es debido al hecho de que el operador avance F es generalmente una función no lineal del modelo de \mathbf{m} .

Para mitigar este problema, se han propuesto los enfoques multiescala [1].

Figura 1.2: Mínimos locales asociados a los modelos de partida.



Estos métodos añaden de forma recursiva los detalles de mayor frecuencia a los primeros modelos calculados a partir de datos de baja frecuencia. 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 [4].

1.1. Marco Teórico

El método FWI requiere una función de costo $\Phi(\mathbf{m})$, 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 costo $\Phi(\mathbf{m})$ es un escalar que mide el error entre los datos observados y los datos modelados. La FWI usa la norma ℓ_2 del error como función costo, el cual está dado por

$$\Phi(\mathbf{m}) = \frac{1}{2} \sum_{x=0}^{N_x} \|\mathbf{d}(\mathbf{x} = x, z = 0, \mathbf{t}|\mathbf{m}) - \mathbf{d}^{obs}(\mathbf{x} = x, z = 0, \mathbf{t})\|_2^2 \quad (1.1)$$

donde $\mathbf{d}^{obs}(\mathbf{x} = x, z = 0, \mathbf{t})$ representa los datos observados en una localización espacial dada ($\mathbf{x} = x, z = 0$), y $\mathbf{d}(\mathbf{x} = x, z = 0, \mathbf{t}|\mathbf{m})$ representa los datos modelados en la mismas coordenadas espaciales, usando un modelo de velocidad conocido \mathbf{m} .

El dato modelado usado para calcular el error en la ecuación 1.1, puede ser obtenido usando la ecuación de onda 2-D acústica e isotrópica. La solución numérica es obtenida usando diferencias finitas en el dominio del tiempo, como será explicado en la sección 2.3.

La actualización para el modelo de velocidad puede ser obtenida como sigue,

$$\mathbf{m}^{k+1} = \mathbf{m}^k + \alpha_k \Delta \mathbf{m}, \quad (1.2)$$

donde $\Delta \mathbf{m}$ se encuentra usando el método de Newton que aproxima la función costo usando la serie de Taylor como sigue

$$\Phi(\mathbf{m}^{k+1}) = \Phi(\mathbf{m}^k) + \Delta \mathbf{m}^T \mathbf{g}(\mathbf{m}^k) + \frac{1}{2} \Delta \mathbf{m}^T \mathbf{H}(\mathbf{m}^k) \Delta \mathbf{m} + \mathbf{O}(\Delta \mathbf{m}^3); \quad (1.3)$$

donde $\mathbf{g}(\mathbf{m}^k) = \frac{\partial \Phi(\mathbf{m}^k)}{\partial \mathbf{m}}$ es el gradiente de la función costo, y $\mathbf{H}(\mathbf{m}^k) = \frac{\partial^2 \Phi(\mathbf{m}^k)}{\partial \mathbf{m}^2}$ es la Hessiana.

Finalmente, se puede encontrar que,

$$\Delta \mathbf{m} = -[\mathbf{H}(\mathbf{m}^k)]^{-1} \mathbf{g}(\mathbf{m}^k). \quad (1.4)$$

Así, la actualización del modelo de velocidad está dada por,

$$\mathbf{m}^{k+1} = \mathbf{m}^k + \alpha_k (-[\mathbf{H}(\mathbf{m}^k)]^{-1} \mathbf{g}(\mathbf{m}^k)). \quad (1.5)$$

A fin de reducir el coste computacional de la actualización del modelo de velocidad en cada iteración de la FWI, la inversa de la matriz Hessiana puede ser reemplazada por la matriz identidad, y el gradiente se puede calcular como propuso Plessix [12]. Plessix define el gradiente de la función de costo como

$$\mathbf{g}(\mathbf{m}^k) = - \sum_s \int_0^T q_s(\mathbf{x}, \mathbf{z}, T - t) \frac{\partial^2 p_s(\mathbf{x}, \mathbf{z}, t)}{\partial t^2} dt, \quad (1.6)$$

donde el campo de onda p_s es el volumen de datos obtenidos a partir de la propagación de fuentes puntuales en la superficie, y el campo de onda q_s , es el volumen de datos obtenidos a partir de la misma ecuación de propagación onda de p_s (ecuación de onda 2D acústica, ver figura 1.3), pero usando como fuente el error entre los datos modelados y observados, en el tiempo inverso, convirtiendo a cada geófono en fuentes. Por esta razón, q_s se llama el campo retropropagado del residual.

En un experimento sísmico real, los datos registrados por los geófonos se conocen como trazas sísmicas. El error entre las trazas sísmicas modeladas y observadas se utiliza como fuentes, para encontrar el campo de onda $q_s((\mathbf{x}, \mathbf{z}, t))$ que son las trazas sísmicas obtenidas por retropropagación. Este proceso se muestra en la Figura 1.4. La principal diferencia entre las Figuras 1.3 y 1.4 es que en la Figura 1.4, la fuente original desaparece y cada geófono se convierte en una fuente dada por es un residual. Al tener ambos campos de onda, la Ecuación 1.6 se utiliza entonces para calcular el gradiente $\mathbf{g}(\mathbf{m}^k)$.

Figura 1.3: Propagación de una fuente (en rojo) a diferentes intervalos de tiempo utilizando la ecuación de onda. Los geófonos se muestran en azul y la zona límites se representan en verde.

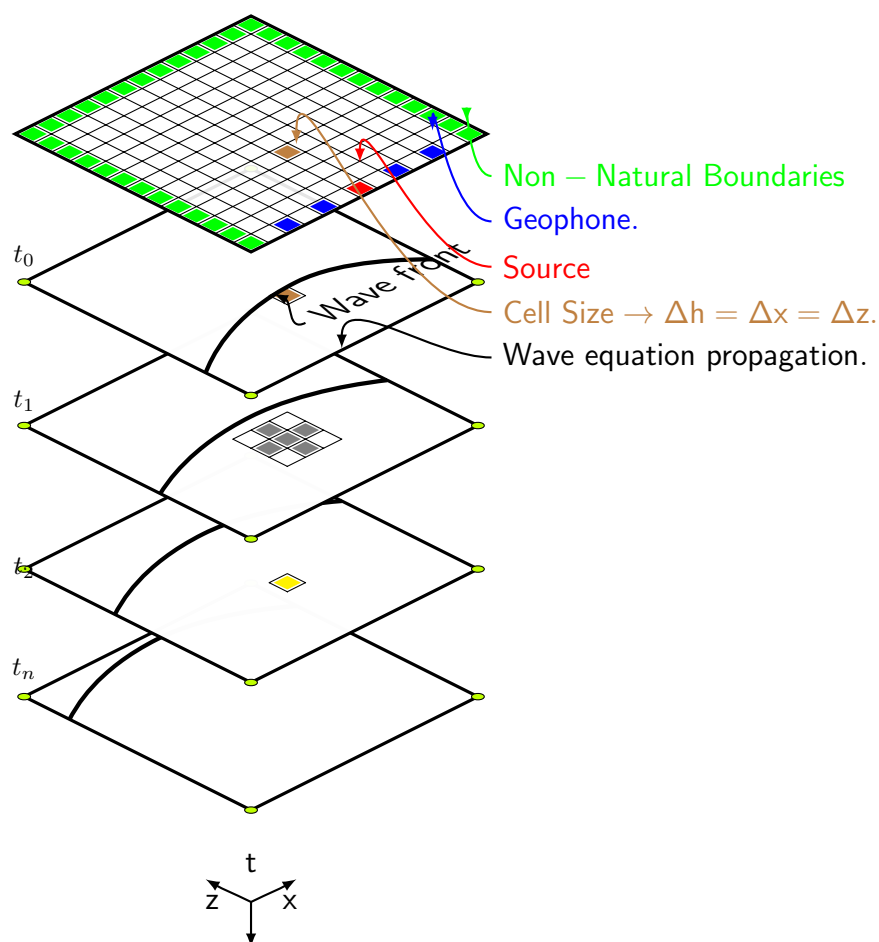
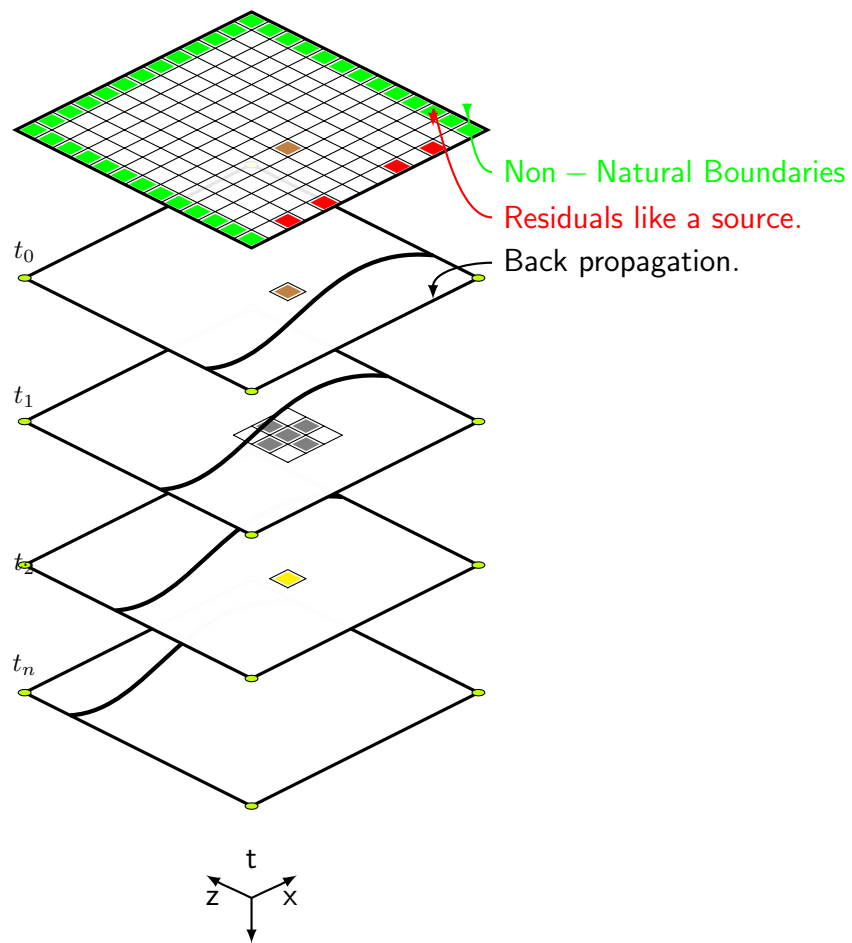


Figura 1.4: Retropropagación de los residuales (en rojo) en diferentes intervalos de tiempo utilizando la ecuación de onda. La zona límites se representa en verde.



Capítulo 2

Inversión de Onda Completa de manera serial

2.1. Algoritmo de programación

La implementación de una Inversión de Onda Completa (por sus siglas en inglés, FWI), es descrita por medio del algoritmo 1, que muestra su estructura de manera general.

Los parámetros de entrada de la función $FWI(m, Obs, S, inf, \alpha)$, son el modelo inicial de velocidades m , las trazas observadas Obs , la ondícula fuente S , la ubicación y offset de cada disparo inf y el valor de alpha, teniendo como datos de salida el modelo final de velocidades y la evolución de la función objetivo ϕ de cada una de las iteraciones de la FWI.

El núcleo de la FWI está concentrado en la correcta implementación de la ecuación de onda acústica escalar para un medio con densidad constante. La propagación consiste en la solución de la ecuación diferencial de la línea 6 con fuentes puntuales $S(sx, t)$ para cada uno de los disparos Ns , cuya solución genera el campo de onda $p_s(x, z, t)$, usado para obtener las trazas modeladas, que son la primera fila del campo de onda p_s en cada capa de tiempo (línea 7).

La norma L2 de los residuales, diferencia entre las trazas observadas y las trazas modeladas (línea 14), determinará la función objetivo ϕ del problema, buscando sea cada vez menor esta diferencia.

La retropropagación, usa la misma ecuación para determinar el campo de onda p_s , pero ahora se toma como fuente, cada uno de los residuales pero

Algorithm 1 Inversión de onda Completa $FWI(m, obs, s, inf, \alpha)$

```
1: for  $i \leftarrow 0, iG$  do ▷ iG, numero de iteraciones de FWI
2:   for  $j \leftarrow 0, Ns$  do ▷ Ns, numero de fuentes en superficie
3:     for  $t \leftarrow 0, Nt$  do ▷ Nt, numero de pasos de tiempo
4:       For doble con  $Nx * Nz$ , dimensiones del modelo
5:       for  $x \leftarrow 0, Nx; z \leftarrow 0, Nz$  do
6:          $\frac{1}{m(x,z)^2} \frac{\partial^2 p_s}{\partial t^2} = \frac{\partial^2 p_s}{\partial x^2} + \frac{\partial^2 p_s}{\partial z^2} + S(sx, t)$ 
7:          $Mod_j(x, t) \leftarrow p_s(x, 0, t)$ 
8:       end for
9:     end for
10:  end for
11:   $acum \leftarrow 0$ 
12:  for  $j \leftarrow 0, Ns$  do
13:    for  $x \leftarrow 0, Nx; t \leftarrow 0, Nt$  do
14:       $Residual(x, t)_j \leftarrow Mod_j(x, t) - Obs_j(x, t)$ 
15:    end for
16:     $norm \leftarrow \|Residual(x, t)_j\|_2$  ▷ Norma L2
17:     $acum \leftarrow \frac{1}{2}norm + acum$ 
18:  end for
19:   $\phi(i) \leftarrow acum$ 
20:  for  $j \leftarrow 0, Ns$  do
21:    for  $t \leftarrow 0, Nt$  do
22:      for  $x \leftarrow 0, Nx; z \leftarrow 0, Nz$  do
23:         $\frac{1}{m(x,z)^2} \frac{\partial^2 q_s}{\partial t^2} = \frac{\partial^2 q_s}{\partial x^2} + \frac{\partial^2 q_s}{\partial z^2} + Residual_j(x, Nt - t)$ 
24:      end for
25:    end for
26:  end for
27:  for  $x \leftarrow 0, Nx; z \leftarrow 0, Nz$  do
28:     $g(x, z) = -\sum_{Ns} \int_0^T q_s(x, z, T - t) \frac{\partial^2 p_s(x, z, t)}{\partial t^2} dt$  ▷ Gradiente
29:  end for
30:  for  $x \leftarrow 0, Nx; z \leftarrow 0, Nz$  do
31:     $m(x, z) \leftarrow m(x, z) - \alpha_k [\mathbf{I}]g(x, z)$  ▷ Actualización del modelo
32:  end for
33: end for
34:  $m_{end} \leftarrow m(x, z)$ 
35: return  $\phi, m_{end}$  ▷ Resultados FWI
```

partiendo desde la ultima fila hacia la primera, es decir, iniciando desde el tiempo final, convirtiendo cada uno de los geófonos en fuentes puntuales; la

solución de esta ecuación genera el campo de onda q_s

Una vez obtenido el campo de onda propagado p_s y el retropropagado q_s , se implementa la ecuación de la línea 28 [12], que determina el gradiente del modelo de velocidades. Finalmente se realiza la actualización del modelo (línea 31).

2.2. Ecuación de onda acústica isotrópica y CPML

La ecuación de onda acústica 2D con densidad constante para un medio isotrópico está definida por

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

Donde $m(x, z)$ es la velocidad acústica del medio en función de las variables x y z , p denota el campo de presión escalar, y t la variable de tiempo.

Las condiciones de frontera deben tenerse en cuenta en la solución de la ecuación de onda acústica utilizando alguna técnica de análisis numérico. Se requieren las condiciones de frontera para minimizar los reflejos artificiales en el interior del área de interés. se han propuesto varios métodos para incluir las condiciones de frontera en la solución de la ecuación de onda acústica, entre ellos *Perfectly Matched Layer* (PML) y *Convolutional Perfectly Matched Layer* (CPML) [11]. La ecuación de onda acústica isotrópica incluyendo CPML requiere de dos variables auxiliares adicionales en cada dimensión espacial. Modificando la ecuación 2.1 tenemos que

$$\frac{1}{m^2(x, z)} \frac{\partial^2 p}{\partial t^2} = \frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial z^2} + \frac{\partial \psi_x}{\partial x} + \frac{\partial \psi_z}{\partial z} + \zeta_x + \zeta_z. \quad (2.2)$$

Segun Pasalic, en [11], las variables auxiliares que minimizan las reflexiones artificiales en las fronteras vienen dadas por

$$\psi_q^n = b_q \psi_q^{n-1} + a_q \frac{\partial p}{\partial q} \quad (2.3)$$

$$\zeta_q^n = b_i \zeta_q^{n-1} + a_q \left[\left(\frac{\partial^2 p}{\partial q^2} \right)^n + \left(\frac{\partial \psi_q}{\partial q} \right)^n \right] \quad (2.4)$$

donde q puede ser x o z , ψ_q^0 y ζ_q^0 son cero en la primera iteración, y a_q y b_q pueden ser calculados usando las ecuaciones presentadas en [2]. A continuación se calcularán a modo de ejemplo los parámetros para la dimensión x .

$$R = 0,001$$

$$L_x = CPMLimit * \Delta h$$

$$d_0 = \frac{-3}{2L_x} \log(R)$$

Estos tres valores, son constantes para cualquier posición y momento de la propagación. Δh , es el valor del paso espacial, y $CPMLimit$ es la cantidad de pasos sobre la cual se desea realizar la atenuación, así, L_x es el ancho de la región de CPML.

Se define un vector $f(x)$ como:

$$f(x) = \begin{cases} L_x : \Delta h : 0 & x \in (0, CPML] \\ 0 & x \in (CPML, Nx - CPML) \\ 0 : \Delta h : L_x & x \in (Nx - CPML, Nx) \end{cases}$$

Y se muestra en la figura 2.1.

Usando las definiciones de [2], a_q y b_q pueden ser calculados usando las ecuaciones 2.5 al 2.8.

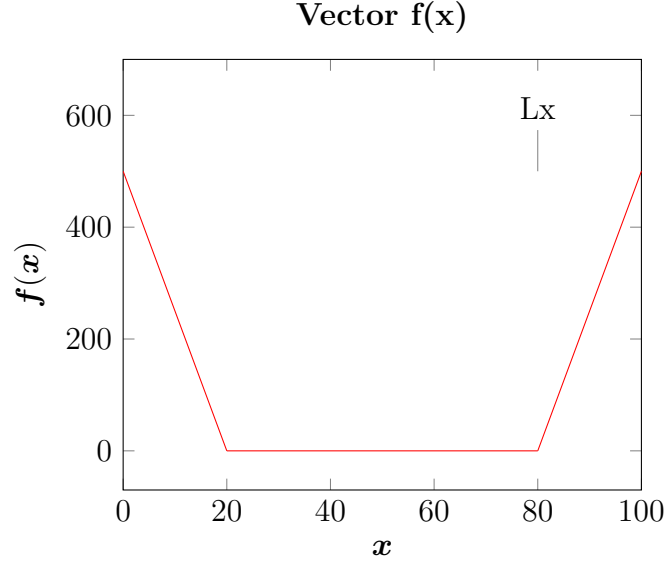
$$d_x = d_0 V_{max} \left(\frac{f(x)}{L_x} \right)^2, \quad (2.5)$$

$$\alpha_x = \pi f \left(\frac{L_x - f(x)}{L_x} \right), \quad (2.6)$$

$$b_x = e^{-(d_x + \alpha_x) dt}, \quad (2.7)$$

$$a_x = \frac{d_x}{d_x - \alpha_x} (b_x - 1), \quad (2.8)$$

Figura 2.1: Vector $f(x)$, con $Nx=100$, $CPMLimit=20$, $\Delta h=25$, $Lx=500$, $f=3Hz$



Las figuras 2.2 y 2.3 muestra los parametros a_x , b_x y d_x , como función de x . Al igual que $f(x)$, estos parámetros se definen como 0 para regiones donde **NO** se aplique atenuación CPML, es decir, $20 < x < 80$, lo que implica que las variables ψ_x^n y ζ_x^n también son cero, y por lo tanto la actualización para el campo escalar está dada por la solución de la ecuación 2.1. Por otro parte, la ecuación 2.2 y las ecuaciones de CPML, sólo serán valida en las fronteras atenuantes, es decir, cuando $x < 20$ y $x > 80$ [16].

La forma como se definen a_x y b_x , se extrapola a la dimensión restante(z). Para el cálculo de a y b en la dimensión restante(z) se hace el mismo análisis.

2.3. Diferencias finitas

Se usará diferencias finitas de segundo orden para realizar la discretización de las ecuaciones 2.1 y 2.2. Se usará la siguiente notación para definir la posición espacial [columna, fila], sabiendo que las filas indican la profundidad.

$$p_{posicion} = \begin{cases} posicion = center & posicion = [i, j] \\ posicion = left & posicion = [i - 1, j] \\ posicion = right & posicion = [i + 1, j] \\ posicion = top & posicion = [i, j - 1] \\ posicion = bottom & posicion = [i, j + 1] \end{cases}$$

Figura 2.2: Valores a_x (azul) y b_x (rojo), con $Nx=100$, $CPMLimit=20$, $dt=4e-3$.

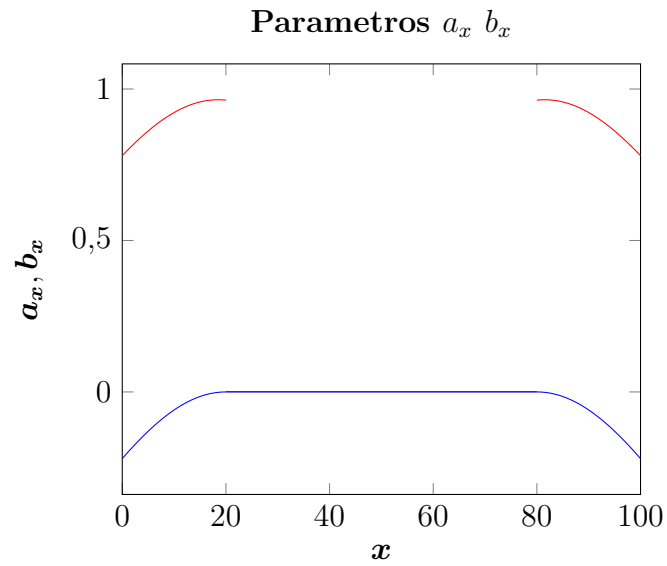
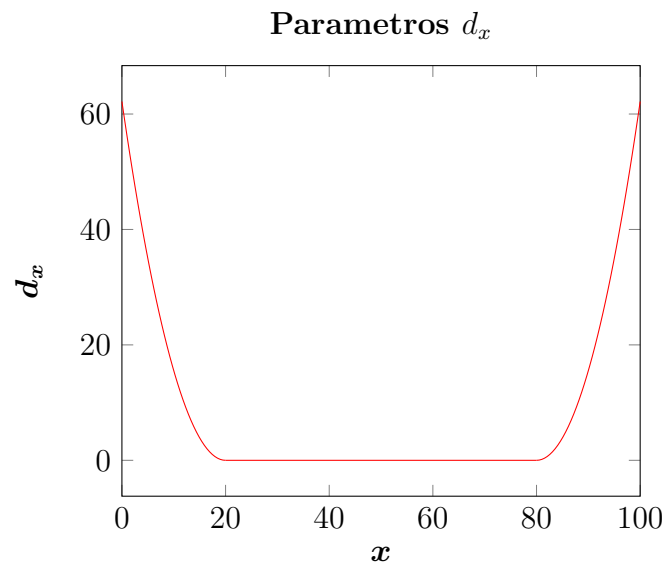


Figura 2.3: Valores d_x , con $Nx=100$, $CPMLimit=20$, $dt=4e-3$.



Solo se tomará la letra inicial, es decir, la posición será referenciada por las letras c , l , r , t y b . Las variables auxiliares, ψ_q y ζ_q , son de igual tamaño a la matriz p , dado que dependen de derivadas de ésta última.

Llevaremos a diferencias finitas la parte espacial de la ecuación 2.2, tomando solo la dimensión x y una vez obtenida una expresión, extrapolaremos su definición a la dimensión restante.

Se iniciará por encontrar expresiones discretas de las variables auxiliares. La malla espacial, será equidistante entre sus puntos, es decir, $dx = dz = dh$, por lo tanto

$$\frac{\partial p_c}{\partial x} = \frac{p_c - p_l}{dh} \quad (2.9)$$

En la ecuación 2.3 se define el valor de ψ_x , este depende de él mismo en un tiempo anterior. Dado que este termino no será usado nuevamente, podemos reasignar el valor en cada iteración. Aplicando esta modificación a la ecuación 2.3 tenemos

$$\psi_x = b_x \psi_x + a_x \frac{\partial p}{\partial x}, \quad (2.10)$$

y reemplazando 2.9 en 2.10, obtenemos

$$\psi_{x,c} = b_x \psi_{x,c} + a_x \left(\frac{p_c - p_l}{dh} \right) \quad (2.11)$$

Definimos un nuevo termino.

$$\frac{\partial p'}{\partial x} = \frac{\partial p}{\partial x} + \psi_x \quad (2.12)$$

Si se toma la expresión 2.12 y se deriva respecto a x , nos queda

$$\frac{\partial^2 p'}{\partial x^2} = \frac{\partial^2 p}{\partial x^2} + \frac{\partial \psi_x}{\partial x}. \quad (2.13)$$

Discretizando la ecuación 2.13 por diferencias finitas de segundo orden, nos lleva a

$$\frac{\partial^2 p'_c}{\partial x^2} = \left(\frac{p_r - 2p_c + p_l}{dh^2} \right) + \left(\frac{\psi_{x,r} - \psi_{x,c}}{dh} \right). \quad (2.14)$$

En la ecuación 2.4 que define el valor de ζ_x , este depende de él mismo en un tiempo anterior, pero dado que este termino no será usado nuevamente, solo reasignamos el valor de cada posición. Modificando ecuación 2.4,

$$\zeta_x = b_x \zeta_x + a_x \left[\frac{\partial^2 p}{\partial x^2} + \frac{\partial \psi_x}{\partial x} \right] \quad (2.15)$$

y usando la ecuación 2.13 llegamos a

$$\zeta_x = b_x \zeta_x + a_x \left[\frac{\partial^2 p'}{\partial x^2} \right] \quad (2.16)$$

Para discretizar la ecuación 2.16 reemplazamos 2.14 en 2.16

$$\zeta_{x,c} = b_{x,i} \zeta_{x,c} + a_{x,i} \left[\frac{p_r - 2p_c + p_l}{dh^2} + \frac{\psi_{x,r} - \psi_{x,c}}{dh} \right] \quad (2.17)$$

Adicionalmente, si usamos un nuevo término, para agrupar la parte espacial en x de la ecuación de onda acústica obtenemos

$$\frac{\partial^2 p'_x}{\partial x^2} = \frac{\partial^2 p}{\partial x^2} + \frac{\partial \psi_x}{\partial x} + \zeta_x. \quad (2.18)$$

Con la ecuación 2.13, podemos compactar la discretización espacial en x como

$$\frac{\partial^2 p'_{x,c}}{\partial x^2} = \frac{\partial^2 p'_c}{\partial x^2} + \zeta_{x,c}, \quad (2.19)$$

y si reemplazamos la ecuación 2.14 en 2.19, nos queda

$$\frac{\partial^2 p'_{x,c}}{\partial x^2} = \left(\frac{p_r - 2p_c + p_t}{dh^2} \right) + \left(\frac{\psi_{x,r} - \psi_{x,c}}{dh} \right) + \zeta_{x,c}. \quad (2.20)$$

Ahora si llevamos las ecuaciones 2.11, 2.17 y 2.20, a la dimensión en z , obtenemos

$$\psi_{z,c} = b_{z,j} \psi_{z,c} + a_{z,j} \left(\frac{p_c - p_t}{dh} \right), \quad (2.21)$$

$$\zeta_{z,c} = b_{z,j} \zeta_{z,c} + a_{z,j} \left[\frac{p_b - 2P_c + p_t}{dh^2} + \frac{\psi_{z,b} - \psi_{z,c}}{dh} \right], \quad (2.22)$$

$$\frac{\partial^2 p'_{z,c}}{\partial z^2} = \left(\frac{p_b - 2p_c + p_t}{dh^2} \right) + \left(\frac{\psi_{z,b} - \psi_{z,c}}{dh} \right) + \zeta_{z,c}. \quad (2.23)$$

Por lo tanto la ecuacion de onda acústica con CMPL puede ser escrita como

$$\frac{1}{m^2} \frac{\partial^2 p}{\partial t^2} = \frac{\partial^2 p'_x}{\partial x^2} + \frac{\partial^2 p'_z}{\partial z^2}. \quad (2.24)$$

Sumando y organizando las ecuaciones 2.20 y 2.23, nos queda

$$\left(\frac{p_t + p_r + p_b + p_t - 4p_c}{dh^2} \right) + \left(\frac{\psi_{x,r} - \psi_{x,c} + \psi_{z,b} - \psi_{z,c}}{dh} \right) + (\zeta_{x,c} + \zeta_{z,c}). \quad (2.25)$$

De ahora en adelante se usará la siguiente notación para definir la posición temporal de la matriz p . Esta notación también sera usada con la notación espacial de c, l, r, t y b .

$$p_c^{tiempo} = \begin{cases} tiempo = n, & Present_c \\ tiempo = n - 1, & Past_c \\ tiempo = n + 1, & Future_c \end{cases}$$

Usando las nuevas variables

$$Present_{st} = p_l + p_r + p_b + p_t, \quad (2.26)$$

$$\psi_{st} = \psi_{x,r} - \psi_{x,c} + \psi_{z,b} - \psi_{z,c}, \quad (2.27)$$

$$\zeta_{st} = \zeta_{x,c} + \zeta_{z,c}. \quad (2.28)$$

y reemplazando en 2.25 tenemos

$$\frac{\partial^2 p'_{x,c}}{\partial x^2} + \frac{\partial^2 p'_{z,c}}{\partial z^2} = \left(\frac{Present_{st} - 4Present_c}{dh^2} \right) + \left(\frac{\psi_{st}}{dh} \right) + (\zeta_{st}) \quad (2.29)$$

$$\frac{\partial^2 p'_{x,c}}{\partial x^2} + \frac{\partial^2 p'_{z,c}}{\partial z^2} = \left(\frac{Present_{st}}{dh^2} \right) - \left(\frac{4Present_c}{dh^2} \right) + \left(\frac{\psi_{st}}{dh} \right) + (\zeta_{st}). \quad (2.30)$$

Llevando a diferencias finitas de segundo orden la parte temporal de la ecuación de onda acústica y reemplazando en la ecuación 2.30

$$\frac{\partial^2 p}{\partial t^2} = \left(\frac{Future_c - 2Present_c + Past_c}{dt^2} \right) \quad (2.31)$$

$$\left(\frac{Future_c - 2Present_c + Past_c}{dt^2} \right) = m^2 * \left[\frac{\partial^2 p'_{x,c}}{\partial x^2} + \frac{\partial^2 p'_{z,c}}{\partial z^2} \right] \quad (2.32)$$

$$Future_c - 2Present_c + Past_c = (m \cdot dt)^2 \cdot \left[\frac{\partial^2 P'_{x,c}}{\partial x^2} + \frac{\partial^2 P'_{z,c}}{\partial z^2} \right] \quad (2.33)$$

$$Future_c - 2Present_c + Past_c = (m \cdot dt)^2 \cdot \left[\left(\frac{Present_{st}}{dh^2} \right) - \left(\frac{4Present_c}{dh^2} \right) + \left(\frac{\psi_{st}}{dh} \right) + (\zeta_{st}) \right] \quad (2.34)$$

Despejando $Future_c$ y usando la siguiente definición, resolvemos la ecuación 2.34.

$$G = \frac{m \cdot dt}{dh}$$

$$Future_c = G^2 \cdot Present_{st} + (2 - 4G^2) \cdot Present_c + G^2 dh \cdot \psi_{st} + (G \cdot dh)^2 \zeta_{st} - Past_c \quad (2.35)$$

2.3.1. Resumen diferencias finitas

Las ecuaciones de aplicación de CMPL en un modelo de propagación por medio de la ecuación de onda acústica 2D, una vez calculado a_x , b_x , a_z y b_z , son:

$$G = \frac{m * dt}{dh},$$

en el cálculo de la primera iteración para las variables auxiliares, se toman,

$$\psi_{x,c} = 0,$$

$$\psi_{z,c} = 0,$$

$$\zeta_{x,c} = 0,$$

$$\zeta_{z,c} = 0;$$

Las variables auxiliares, se implementan mediante las ecuaciones

$$\psi_{x,c} = b_x \psi_{x,c} + a_x \left(\frac{Present_c - Present_l}{dh} \right), \quad (2.36)$$

$$\psi_{z,c} = b_z \psi_{z,c} + a_z \left(\frac{Present_c - Present_t}{dh} \right), \quad (2.37)$$

$$\zeta_{x,c} = b_x \zeta_{x,c} + a_x \left[\frac{Present_r - 2Present_c + Present_l}{dh^2} + \frac{\psi_{x,r} - \psi_{x,c}}{dh} \right], \quad (2.38)$$

$$\zeta_{z,c} = b_z \zeta_{z,c} + a_z \left[\frac{Present_b - 2Present_c + Present_t}{dh^2} + \frac{\psi_{z,b} - \psi_{z,c}}{dh} \right]. \quad (2.39)$$

Agrupando las variables de la siguiente manera,

$$Present_{st} = Present_l + Present_r + Present_b + Present_t, \quad (2.40)$$

$$\psi_{st} = \psi_{x,r} - \psi_{x,c} + \psi_{z,b} - \psi_{z,c}, \quad (2.41)$$

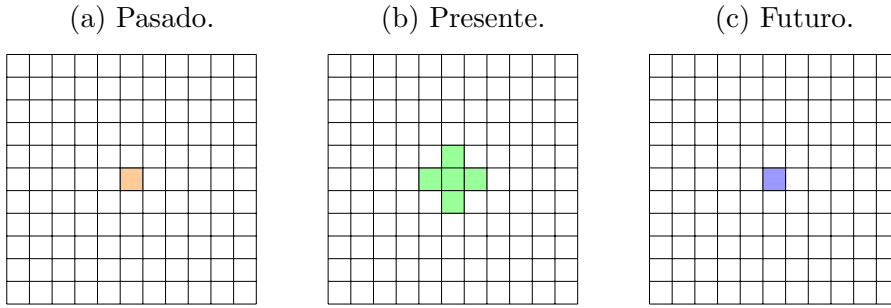
$$\zeta_{st} = \zeta_{x,c} + \zeta_{z,c}; \quad (2.42)$$

Obtenemos la expresión compacta que efectúa la propagación en dos dimensiones aplicando CPML en las fronteras,

$$Future_c = G^2 \cdot Present_{st} + (2 - 4G^2) \cdot Present_c + G^2 dh \cdot \psi_{st} + (G \cdot dh)^2 \zeta_{st} - Past_c. \quad (2.43)$$

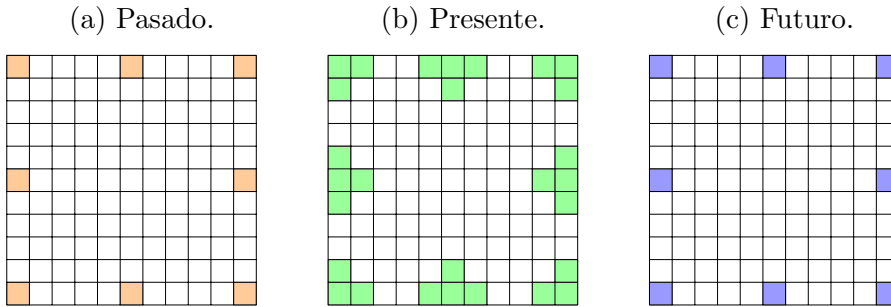
Una representación gráfica de esta ecuación en una región fuera de la zona límite, se muestra en la figura 2.4. Se puede observar que un solo punto del futuro requiere la información de 5 puntos del campo de onda presente y un punto en el campo de onda pasado, los cuales se suponen conocidos.

Figura 2.4: Stencil para ecuación de onda 2D fuera de la zona límite.



La representación gráfica del stencil para calcular un punto localizado en el area límite límite, es mostrada en la figura 2.5

Figura 2.5: Stencil para ecuación de onda 2D en la zona límite.



2.4. Implementación ecuación de onda acústica 2D

Realizando la implementación de las ecuaciones 2.36 a la 2.43, se detallan los resultados de incluir o no las condiciones frontera (CPML). Se usa un

modelo de velocidad constante ($4000\frac{m}{s}$), cuyas dimensiones son $Nx=4500m$ y $Nz=2500m$ donde el paso espacial es de $25m$. Se realiza simulación de 3 segundos de duración con un paso temporal de $4ms$. Los efectos de la implementación de CPML en la propagación de la onda de presión, se muestran en la figura 2.6.

El efecto de las condiciones frontera no es notorio en la propagación de la onda de presión, cuando el frente de onda no ha llegado a la zona límite (figuras 2.6a y 2.6b). En la figura 2.6c, la no presencia de la atenuación CPML se hace muy evidente en los $1.8[s]$ de simulación, tiempo en el cual, se ha producido la reflexión de la onda de presión en el límite inferior del modelo, mientras que en el mismo tiempo, la implementación con presencia de CPML (figura 2.6d) no presenta estas reflexion. La figura 2.6e presentan una reflexión total de todo el frente de onda, en cambio, la figura 2.6f muestra un campo de presión con reflexiones casi inexistentes, mostrando la utilidad y eficacia de la implementación de CPML en la ecuación de onda acústica.

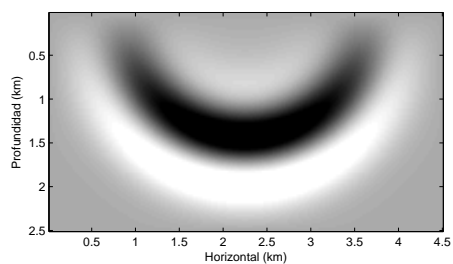
*Una traza sísmica representa la respuesta del campo de ondas elásticas a la velocidad y los contrastes de densidad a lo largo de las interfases de las capas de rocas o los sedimentos a medida que la energía viaja a través de las capas del subsuelo, desde una fuente hasta un receptor o un arreglo de receptores [14]. A su vez, el conjunto de trazas obtenidas para un mismo disparo se llama *Shot gather*. Con la correcta implementación de la ecuación de onda acústica 2D de densidad constante, se pueden extraer *Shot gather* para cualquier disparo partiendo de un modelo de velocidades, como se muestra en la figura 2.7, donde se muestra un modelo de 2 capas de velocidad, y el conjunto de trazas obtenidas. se aprecia la onda directa, generada por el disparo y la onda reflejada al momento de cambiar de capa de velocidad, siendo la primera en registrarse la onda directa, seguida por la onda reflejada.*

2.5. Implementando la FWI

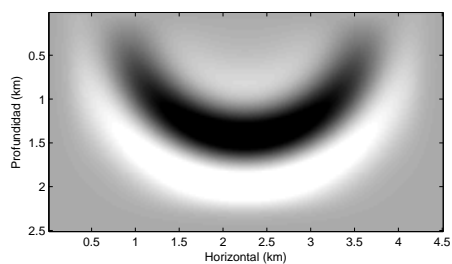
La inversión de onda completa, a nivel de implementación, se puede resumir en la correcta ejecución de una ecuación que modele la propagación de la onda de presión sobre un perfil de velocidades, usando como fuentes primeramente, disparos puntuales y luego, trazas de residuales; y realizar la condición de imagen a partir de la ecuación 2.44 determinada por [12], para calcular el nuevo modelo de velocidades.

Figura 2.6: Propagación,. Figuras de la izquierda sin CPML. Figuras de la derecha con CPML

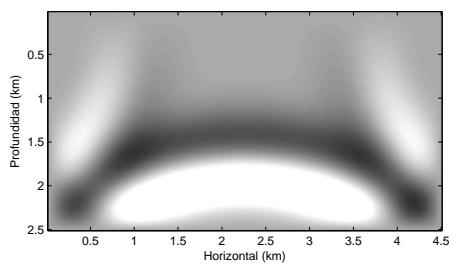
(a) 1400 ms



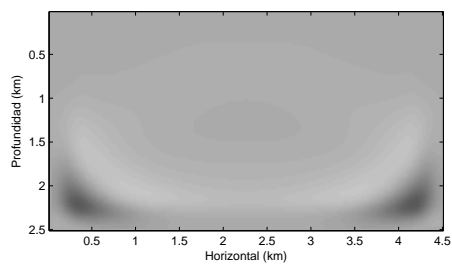
(b) 1400 ms



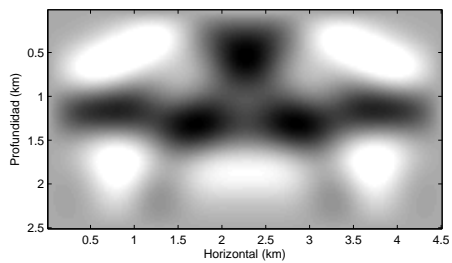
(c) 1800 ms



(d) 1800 ms



(e) 2600 ms



(f) 2600 ms

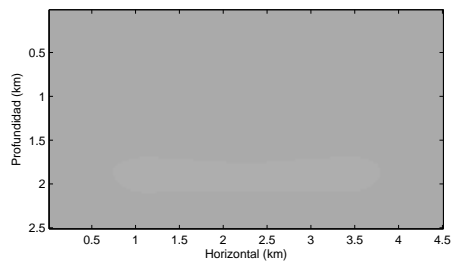
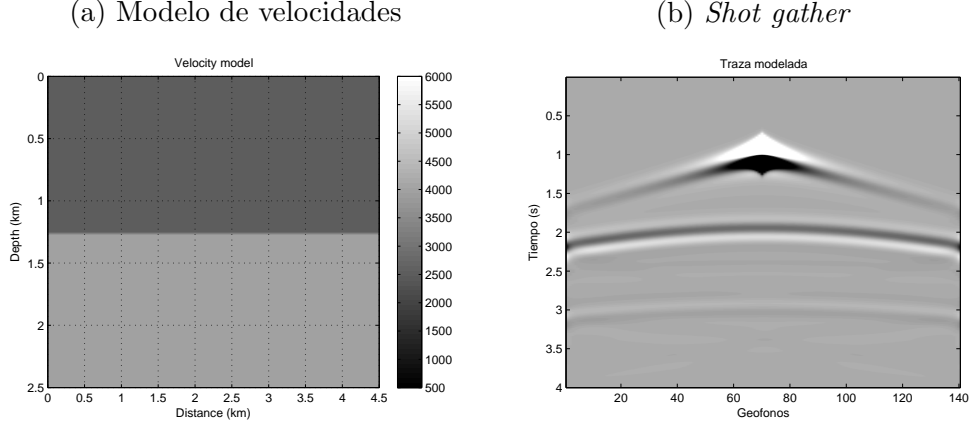


Figura 2.7: *Shot gather* y su modelo asociado



$$G(x, z) = - \sum_{N_s} \int_0^T q_s(x, z, T - t) \frac{\partial^2 p_s(x, z, t)}{\partial t^2} dt \quad (2.44)$$

Donde q_s es el campo de onda retropropagado, y p_s el campo de onda propagado. Para realizar la condición de imagen, se necesita llevar a una forma discreta la ecuación 2.44. Usando la ecuación de onda acústica 2D con densidad constante y CPML 2.2, y las ecuaciones en diferencias finitas 2.25 a la 2.28, la integral de manera discreta estaría definida por,

$$dP_{p_s}(x, z, it) = m^2(x, z) \cdot \left[\left(\frac{Present_{st} - 4Present_c}{dh^2} \right) + \left(\frac{\psi_{st}}{dh} \right) + (\zeta_{st}) + S(x, it) \right] \quad (2.45)$$

$$P_{q_s}(x, z, it) = G^2 \cdot Present_{st} + (2 - 4G^2) \cdot Present_c + G^2 dh \cdot \psi_{st} + (G \cdot dh)^2 \zeta_{st} - Past_c + Residual(x, z, T - it) \quad (2.46)$$

$$G(x, z) = dt \cdot \sum_{it=0}^{Nt} \{P_{q_s}(x, z, T - it)\} \cdot \{dP_{p_s}(x, z, it)\} \quad (2.47)$$

Así, dicho de otra manera, la condición de imagen se realiza con la "película" de la retropropagación y la película de la derivada de la propagación. Para cada disparo se realiza un gradiente del modelo de velocidad y se suman entre ellos, obteniendo una sola condición de imagen para todos los disparos y se actualiza el modelo de velocidad usando la ecuación 2.48,

$$m^{k+1}(x, z) = m^k(x, z) + \alpha_k [\mathbf{I}]G(x, z) \quad (2.48)$$

Donde α_k es un valor del paso de acercamiento del gradiente hacia el nuevo modelo, teniendo en cuenta que un valor pequeño implicaría realizar demasiadas iteraciones para un resultado apreciable, pero un valor grande implicaría un problema de no convergencia del modelo.

El valor de ϕ , implica cuánto nos estamos acercando al modelo real, dado es la magnitud de los residuales entre las trazas reales y las trazas modeladas para cada iteración. El objetivo final de la FWI es obtener una función objetivo ϕ siempre decreciente, sin que presente oscilaciones a medida que avanza hacia el modelo final; determinar el valor del paso, y los diferentes parámetros de ejecución, son prioridad del método para lograr convergencia del problema y obtener buenos resultados.

2.6. Aplicando sobre un Modelo Geofísico complejo

El modelo de velocidad Marmousi, fue creado por el Institut Français du Pétrole (IFP) en 1988, se utiliza para probar el funcionamiento de la FWI. En particular, se utilizó una sección del modelo Marmousi de 5.25km x 1.7km tal como se muestra en la figura 2.8a. Cinco fuentes ubicadas a 625m, 1625m, 2625m, 3625m y 4625m con una profundidad de 75m, se utilizaron para generar los datos observados y el conjunto de trazas modeladas. Los geófonos están localizados cada 25 metros, desde la posición de 625m hasta la posición 4625m en la superficie. Cada geófono registra 2.5s en un intervalo de 4ms.

El modelo de partida para esta implementación, será el modelo Marmousi luego de un suavizado horizontal de 5 puntos de 20 iteraciones. El modelo resultante se muestra en la figura 2.8b.

El modelo de velocidad final se obtiene después de 300 iteraciones y se representa en la Figura 2.8c. En cada iteración, el modelo se actualiza utilizando la ecuación 2.48, con un valor de $\alpha=10$. La figura 2.9 muestra la

Algorithm 2 Modelo de partida

```
1:  $m_0(x, z) \leftarrow Marmousi$ 
2: for  $i \leftarrow 1, 20$  do
3:    $m_i(x, z) \leftarrow \frac{1}{5} \cdot (m_i(x, z - 2) + m_i(x, z - 1) + m_i(x, z) + m_i(x, z + 1) +$   

    $m_i(x, z + 2))$ 
4: end for
5:  $m_{ini}(x, z) \leftarrow m_{20}(x, z)$ 
```

función objetivo como una función del número de iteraciones. En la primera iteración $\phi(m)= 104.58$, y en la iteración 300 $\phi(m)= 6.0589$. El cambio en la función objetivo después de 300 iteraciones es insignificante.

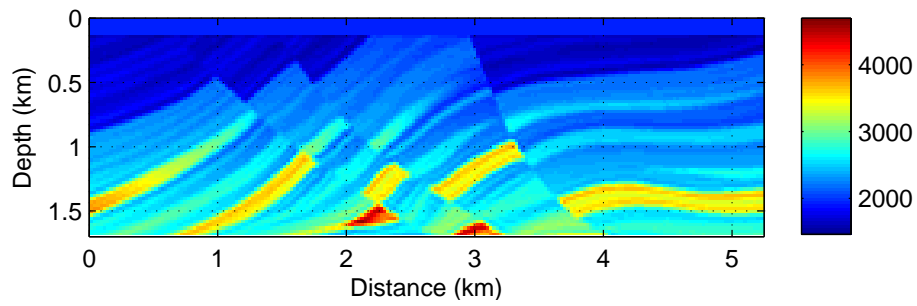
Cuadro 2.1: Arquitectura Intel Core i7-4700HQ

CPU arquitectura	Detalles
Procesador	Intel Core i7-4700HQ
Frecuencia	2.4 GHz
CPU cores	4
Cache size	6144 kB
Ram speed	1600 MHz
Ram size	12 GB

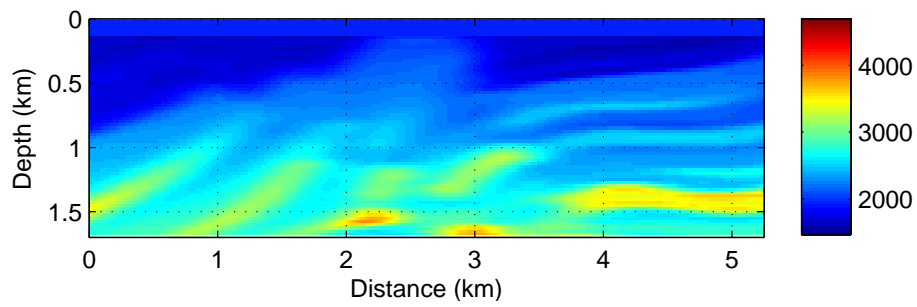
La implementación de FWI de manera serial requiere de 2792.447 segundos de ejecución, equivalente a 46 minutos y 32.447 segundos. Las especificaciones técnicas de la CPU sobre la cual se ejecutaron las pruebas se muestra en la tabla 2.1.

Figura 2.8: Actualización del modelo de velocidades usando FWI

(a) Modelo Original



(b) Modelo Inicial



(c) Modelo Final

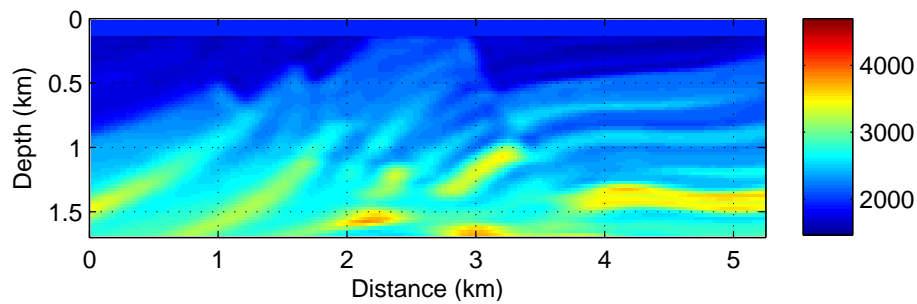
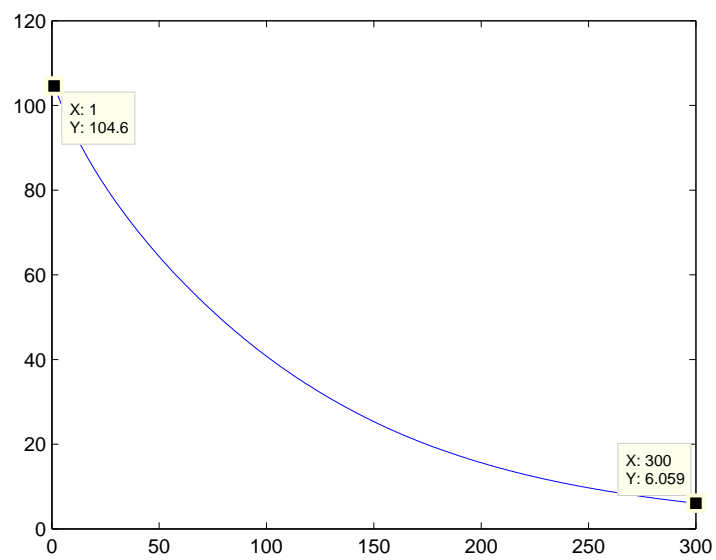


Figura 2.9: Evolución función objetivo $\phi(m)$ de una implementación serial



Capítulo 3

Inversión de Onda Completa de manera paralela

3.1. Arquitectura de Cómputo de Propósito General en unidades de procesamiento gráfico (GPGPU)

Antes de iniciar con la implementación de la Inversión de Onda Completa (FWI) de manera paralela sobre de una arquitectura GPU, se debe definir una serie de conceptos asociados a este tipo de procesadores.

Las unidades de procesamiento gráfico (GPU, por sus siglas en inglés) son procesadores de múltiples núcleos de alto rendimiento con capacidad de realizar procesos masivos de datos. En un inicio, diseñadas especialmente para el manejo de gráficos por ordenador y difíciles de programar, las GPU de hoy son procesadores paralelos de propósito general con soporte para interfaces de programación accesibles y lenguajes estándar de la industria, tales como C [3].

La FWI requiere de una gran cantidad de operaciones para realizar un solo paso de avance en la búsqueda de un mejor perfil de velocidades, la mayoría de estas operaciones son repetitivas y no recursivas en sus resultados, es decir, si pensamos en el nuevo modelo como un conjunto de puntos de búsqueda, no requiere que el cálculo de cada punto del modelo finalice para realizar de forma paralela el cálculo de otro punto de este modelo.

Una de las principal diferencias entre una CPU y un GPU, es la cantidad de

tareas que puede realizar de manera simultánea. Podemos realizar la comparación de la CPU con un carro de carreras, el cual puede ir a gran velocidad, pero llevando consigo un par de personas, mientras que la GPU puede ser un bus, el cual tardará mucho más en llegar a su destino, pero llevando consigo varias decenas de ellas. Dicho de otra manera, una CPU está formada por varios núcleos optimizados para el procesamiento en serie, mientras que una GPU consta de millares de núcleos más pequeños y eficientes diseñados para manejar múltiples tareas simultáneamente. De esta manera, una aplicación se puede acelerar trasladando la parte de la aplicación de mayor costo computacional a la GPU y dejar el resto del código ejecutándose de manera serial en CPU [9].

Usando este ejemplo como base, un *thread* (traducido comúnmente como hilo), es la unidad mínima de procesamiento en un programa, capaz de realizar tareas de manera independiente de forma secuencial. La cantidad de *threads* que podríamos usar en una CPU ronda el orden de una decena, en cambio en una GPU podríamos usar miles de ellos.

En un GPU, el conjunto de *threads* conforman un bloque, y a su vez un conjunto de bloques, generan una *Grid*. Cada thread posee su propia memoria local, al cual solo él tiene acceso. Cada bloque su propia memoria compartida por todos los threads del bloque y con el mismo tiempo de vida que los threads que lo componen, de esta manera los hilos de un bloque pueden cooperar entre sí mediante el uso de memoria compartida dentro del bloque y sincronizando su ejecución para coordinar los accesos a memoria. Todos los hilos en un GPU tienen acceso a la memoria global.

Un Kernel es una función la cual al ejecutarse lo hará en N distintos hilos de manera paralela, es decir, es el segmento de código que designa la "tarea" de cada thread. Los kernel se ejecutan en el *device* (GPU), y necesitan previamente conocer la distribución de los bloques y de los thread de cada bloque.

Las tareas designadas por kernel, se realizan por bloques, hasta que no finalice el primer bloque todas las instrucciones del kernel, no se podrá lanzar el siguiente bloque, y así sucesivamente hasta finalizar la totalidad de los bloques definidos para esta función. Las GPU's pueden lanzar varios bloques de manera simultánea, la cantidad de bloques que pueden operar en paralelo van ligados a la cantidad de *Streaming multiprocessors* y tipo de GPU.

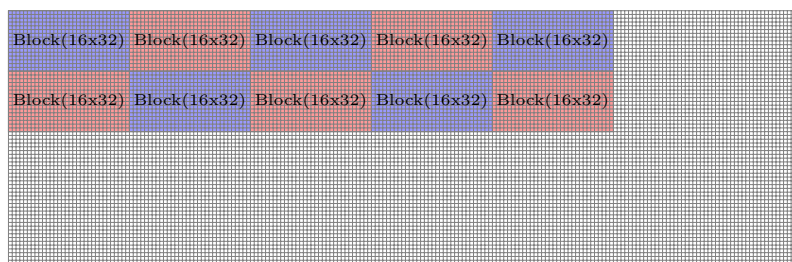
3.2. Segmentando el código

En el algoritmo 1, las secciones de propagación, retropropagación, condición de imagen (gradiente) y cálculo de nuevo modelo, son las secciones de mayor grado de granularidad, que se pueden paralelar haciendo uso del concepto de GPU.

El manejo de la información dentro de un GPU es por medio de punteros, es decir, una dirección de cabecera y una reserva de memoria, cuyo acceso es similar a un vector de posiciones. A pesar de que físicamente la información del modelo, trazas, gradiente y variables auxiliares están distribuidas de forma vectorial, podemos pensar que virtualmente tenemos una matriz, a la que a cada punto, le asignaremos un thread, que realice el cálculo correspondiente a esa ubicación.

Aplicando las ecuaciones 2.36 a la 2.43, se pensará el implementar la propagación de la onda de presión en una GPU. La matriz del modelo, se dividirá en varias secciones de bloques (figura 3.1), compuestos de un número determinado de threads (figura 3.2), cada uno de los cuales tendrá asociada su ecuación en diferencias finitas con los parámetros correspondientes a su ubicación espacial (figura 3.3).

Figura 3.1: Modelo de propagación (68x210 puntos) usando bloques de 16x32 threads.



A diferencia de un código serial en el que se debía esperar a que realizara el cálculo para cada punto, para poder avanzar al siguiente, en un algoritmo paralelo, estos cálculos se realizan de manera simultánea en grupos de threads (bloques). Pero se debe realizar una sincronización de operaciones, es decir, para implementar las ecuaciones en diferencias finitas, deben de calcularse totalmente cada parámetro antes de iniciar el siguiente cálculo; en el caso de la propagación, no se puede calcular la variable ζ , si previamente no se ha calculado la variable ψ .

Figura 3.2: Distribucion de un bloque (16x32 threads).

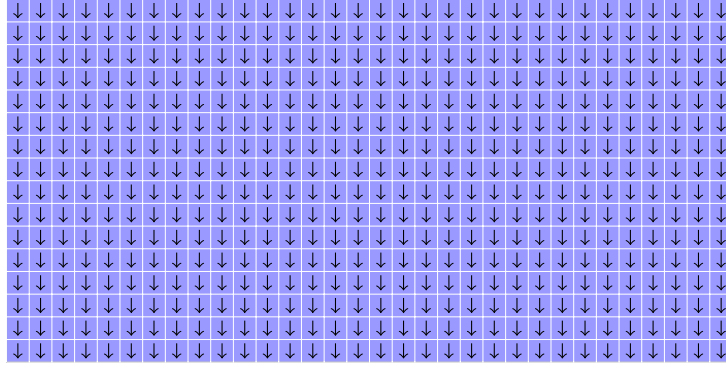
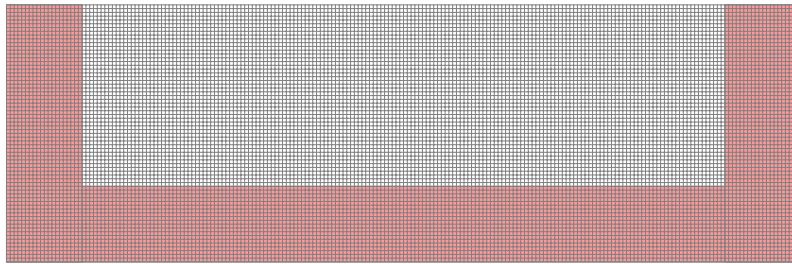


Figura 3.3: Modelo de propagación (68x210 puntos) usando un área de CPML de 20 puntos para la frontera no natural.



En el cálculo de la condición de imagen, cada thread, se encargará de aplicar la ecuación 2.47 de manera discreta. Una vez calculado el gradiente, se aplica el mismo concepto para realizar la actualización del modelo de velocidades.

Dada la restricción de acceso a memoria por los threads, se almacenará toda la información necesaria por la GPU en la memoria global, garantizando el acceso por cada uno de los hilos.

3.3. Algoritmo de programación

El algoritmo 3 muestra la estructura general de una FWI haciendo uso de la programación paralela en una GPU.

El algoritmo 3 estructuralmente es el mismo algoritmo que el presentado en la sección 2.1, con la diferencia principal que la ejecución de las instrucciones de las líneas 4, 10, 17, 20 y 21, se implementaron en GPU, aspecto

Algorithm 3 Inversión de onda Completa $FWI(m, obs, s, inf, \alpha)$

```
1: for  $i \leftarrow 0, iG$  do                                     ▷  $iG$ , numero de iteraciones de FWI
2:   for  $j \leftarrow 0, Ns$  do                               ▷  $Ns$ , numero de fuentes en superficie
3:     for  $t \leftarrow 0, Nt$  do                             ▷  $Nt$ , numero de pasos de tiempo
4:        $\frac{1}{m(x,z)^2} \frac{\partial^2 p_s}{\partial t^2} = \frac{\partial^2 p_s}{\partial x^2} + \frac{\partial^2 p_s}{\partial z^2} + S(sx, it)$ 
5:        $Mod_j(x, t) \leftarrow p_s(x, 0, t)$ 
6:     end for
7:   end for
8:    $acum \leftarrow 0$ 
9:   for  $j \leftarrow 0, Ns$  do
10:     $Residual_j(x, t) \leftarrow Mod_j(x, t) - Obs_j(x, t)$ 
11:     $norm \leftarrow \|Residual(x, t)_j\|_2$                                      ▷ Norma L2
12:     $acum \leftarrow \frac{1}{2}norm + acum$ 
13:  end for
14:   $\phi(i) \leftarrow acum$ 
15:  for  $j \leftarrow 0, Ns$  do
16:    for  $t \leftarrow 0, Nt$  do
17:       $\frac{1}{m(x,z)^2} \frac{\partial^2 q_s}{\partial t^2} = \frac{\partial^2 q_s}{\partial x^2} + \frac{\partial^2 q_s}{\partial z^2} + Residual_j(x, Nt - it)$ 
18:    end for
19:  end for
20:   $g(x, z) = -\sum_{Ns} \int_0^T q_s(x, z, T - t) \frac{\partial^2 p_s(x, z, t)}{\partial t^2} dt$                                      ▷ Gradiente
21:   $m(x, z) \leftarrow m(x, z) - \alpha_k [\mathbf{I}]g(x, z)$                                      ▷ Actualización del modelo
22: end for
23:  $m_{end} \leftarrow m(x, z)$ 
24: return  $\phi, m_{end}$                                      ▷ Resultados FWI
```

que cambia el uso del doble ciclo for para recorrer el modelo, por un ciclo relacionado con el numero de bloques reservados necesarios para cubrir el modelo y *Streaming multiprocessors* disponibles por la GPU.

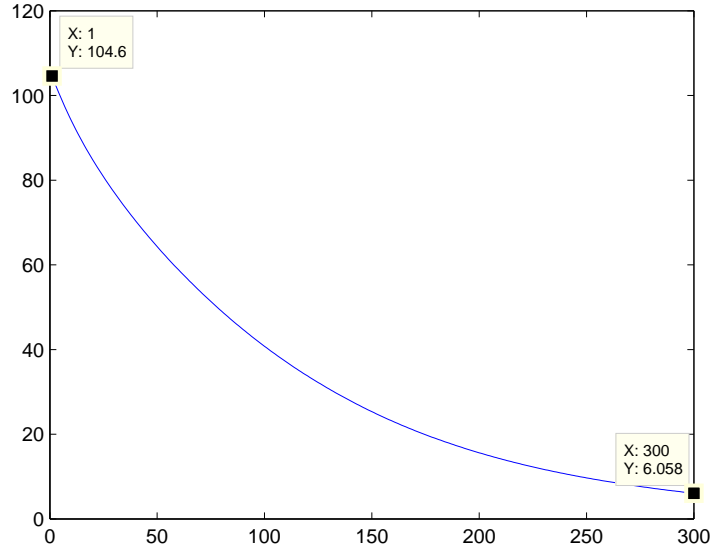
3.4. Aplicación de la FWI en GPU sobre un modelo geofísico

Al igual que en el capítulo anterior, se utilizó una sección del modelo Marmousi de 5.25km x 1.7km tal como se muestra en la figura 3.5a como modelo real. Cinco fuentes ubicadas a 625m, 1625m, 2625m, 3625m y 4625m

con un profundidad de 75m, se utilizaron para generar los datos observados y el conjunto de trazas modeladas. Los geófonos están localizados cada 25 metros, desde la posición de 625m hasta la posición 4625m en la superficie. Cada geófono registra 2.5s en un intervalo de 4ms.

El modelo de partida para esta implementación, es el modelo Marmousi luego de un suavizado horizontal de 5 puntos de 20 iteraciones. El modelo resultante se muestra en la figura 3.5b. El modelo de velocidad final se obtiene después de 300 iteraciones y se representa en la Figura 3.5c. En cada iteración, el modelo se actualizó usando un valor de $\alpha=10$. La figura 3.4 muestra la función costo como una función del número de iteraciones. En la primera iteración $\phi(m)= 104.5847$, y en la iteración 300 $\phi(m)= 6.0585$. El cambio en la función costo después de la iteración 300 es insignificante.

Figura 3.4: Evolución función objetivo $\phi(m)$ de una implementación paralela



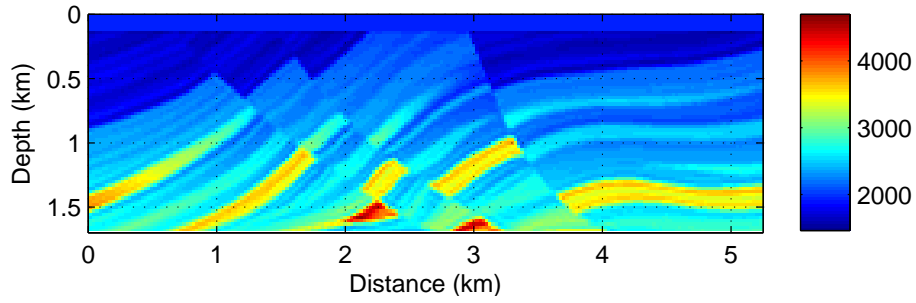
La memoria RAM necesaria para calcular el FWI acústica 2D utilizando la aplicación propuesta está dada por

$$Ram\ size = \frac{\beta}{1024^2 * 8} * (N_s * N_t * N_{traza} + N_t + 11 * N_x * N_z + N_x * N_t + 2 * N_x * N_z * N_t + 4 * N_x + 4 * N_z + N_t * N_{traza}) \quad (3.1)$$

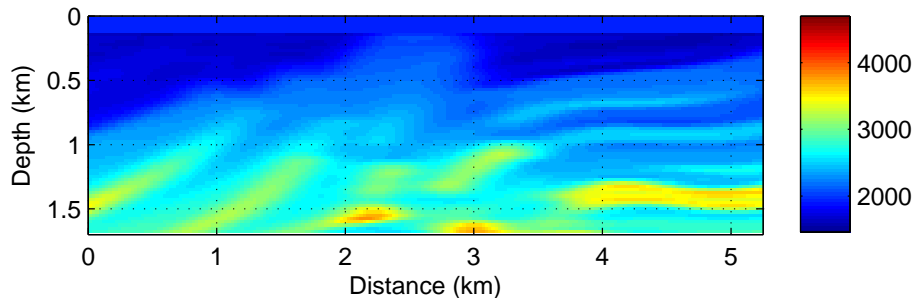
donde N_s es el número de disparos, N_t es el número de pasos de tiempo, N_{traza} es el número máximo de trazas, N_x , y N_z son las dimensiones

Figura 3.5: Actualización del modelo de velocidades usando FWI

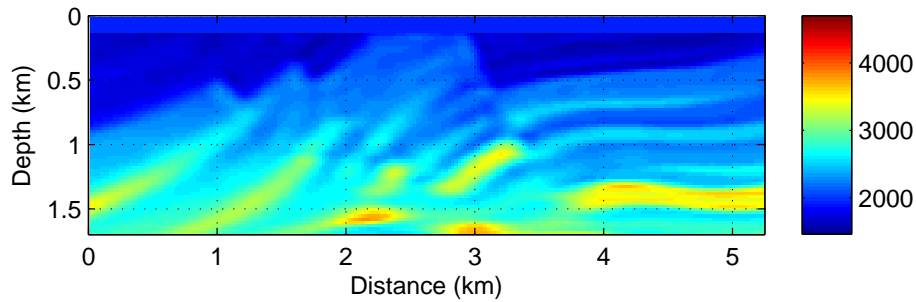
(a) Modelo Original



(b) Modelo Inicial



(c) Modelo Final



espaciales, y β es el número de bits. Para el experimento geofísico estudiado en este trabajo, los valores numéricos de los parámetros son $N_s = 5$, $N_t = 625$, $N_{traza} = 170$, $N_x = 210$, $N_z = 68$, y $\beta = 32$. El tamaño total de memoria requerida es 71,631 MB. Un experimento geofísico grande requerirá tanto arquitecturas de computadora más sofisticados o estrategias de

aplicación que tiene en cuenta la transferencia de datos desde la RAM al disco o de disco a la RAM.

La implementación de FWI de manera paralela requiere de 92.4008 segundos de ejecución, equivalente a 1 minutos y 32.4008 segundos. Se tomó medidas del tiempo en GPU, donde el 71.4 % del tiempo total de ejecución corresponde a lanzamientos de kernel e instrucciones en GPU (envío de datos entre *host* y *device*), distribuidos como lo muestra la tabla 3.2. Las especificaciones técnicas de la GPU sobre la cual se ejecutaron las pruebas se muestra en la tabla 3.1.

Cuadro 3.1: Arquitectura GPU GTX 660

Item	GTX 660
Stream Multiproc.(SMs)	5
Blocks per SM	2
Threads per Block (Max.)	1024
Threads per Block (Min.)	32
32b registers per Block (Max.)	64K
Global memory	2GB
L2 cache size	256kB
Shared memory per block	48kB
Clk frequency	540MHz

Cuadro 3.2: Tiempo ejecución de kernel en GPU

Kernel	Tiempo
Propagación	63.6867 s
Gradiente	2.2505 s
CPMLx	12.6031 ms
CPMLy	10.1855 ms
Residual	35.5631 ms
Tiempo total en GPU	65.9956 s
Tiempo total FWI	92.4008 s
Porcentaje	71.423 %

Capítulo 4

Conclusiones

Se utilizó diferencias finitas en el dominio del tiempo discreto para encontrar la solución numérica de la ecuación de onda acústica 2-D, que se utiliza para calcular el FWI. La aproximación utilizada en este trabajo es de segundo orden en el tiempo y segundo orden en el espacio. Se encontró que una aproximación de segundo orden no afecta en gran medida la calidad de la FWI, y ofrece un bajo costo computacional, lo que hace el cálculo del FWI factible para un problema geofísico. Un análisis más detallado se puede realizar en la evaluación de la mejora en la resolución, y el costo computacional de los modelos de velocidad cuando se utilizan aproximaciones de orden superior.

El uso de las GPU acelera el cálculo de la FWI ya que este algoritmo es altamente paralelizable. Sin embargo, la preocupación principal se plantea en términos de los requisitos de memoria RAM. Para el experimento geofísico estudiado en este trabajo, los requisitos de memoria RAM eran suficientes. Sin embargo, estudios geofísicos reales pueden requerir recursos de memoria RAM de más de 4 GB. Cuando el experimento geofísico es en 3D, los requisitos de memoria RAM aumenta aún más, y la implementación FWI deberían tener en cuenta la transferencia de datos entre la memoria RAM y disco.

Se encontró que para el experimento geofísico estudiado en este trabajo, con los recursos de *Hardware* disponibles, la implementación de la FWI es 30 veces más rápida a su contraparte serial. Se pueden realizar mejoras de eficiencia dentro del código obteniendo tiempos de ejecución mucho menores, usando de manera eficiente los recursos de la tarjeta como la memoria compartida o la distribución de *threads* y bloques para cada experimento geofísico en particular, obteniendo el máximo rendimiento de la GPU para cada implementación, teniendo una FWI eficiente dinámica al tipo de expe-

rimento y GPU a usar.

Los resultados finales por cada una de las implementaciones son similares, pero no son iguales, teniendo diferencias en el orden 0.02% al final de las ejecuciones. Esto se debe a que la tarjeta gráfica usada en este experimento, es un dispositivo especializado en la ejecución de gráficos y juegos, y no es una tarjeta de computo científico. En experimentos geofísicos más grandes, se debe de tener en cuenta la implementación sobre una GPU para computación de alto rendimiento (*High performance Computing* o HPC en inglés) como la Tesla K40 [18].

Por último, pueden producirse problemas de sincronización cuando el FWI se calcula utilizando arquitecturas paralelas como las GPUs. Es fundamental para implementar el algoritmo tener en cuenta la sincronización de todos los procesos que se ejecutan dentro de la GPU. Para la aplicación GPU propuesta, es posible calcular cada punto de la capa campo de presión futuro en paralelo. Sin embargo, es crítico garantizar que las capas de campo de presión presentes y pasados y se calculan antes de calcular la capa futuro, de lo contrario se propaga un error a través de todas las capas, generando errores de tipo aleatorio en cada ejecución que se realice.

Bibliografía

- [1] BOONYASIRIWAT, C., V. P. R. P. C. W. S. G. T., AND MACY, B. An efficient multiscale method for time-domain waveform tomography. *Geophysics* 74 (2009).
- [2] COLLINO, F., AND TSOGKA, C. Application of the perfectly matched absorbing layer model to the linear elastodynamic problem in anisotropic heterogeneous media. *Geophysics* 66, 1 (2001), 294–307.
- [3] GENERAL-PURPOSE COMPUTATION ON GRAPHICS HARDWARE. www.gpgpu.org/about.
- [4] HICKS, G. J., AND PRATT, R. G. Reflection waveform inversion using local descent methods: Estimating attenuation and velocity over a gas-sand deposit. *Geophysics* 66, 2 (2001), 598–612.
- [5] JONES, I. Tutorial: Incorporating near-surface velocity anomalies in pre-stack depth migration models. first break. 47–58.
- [6] KREBS, J. R. Fast full-wavefield seismic inversion using encoded sources. *Geophysics* 74 (2009).
- [7] LAILLY, P. The seismic inverse problem as a sequence of beforestack inversions. conference on inverse scattering: Theory and applications. 206–220.
- [8] MENG, Z., AND SCALES, J. A. 2d tomography in multi-resolution analysis model space. *SEG Technical Program Expanded Abstracts 15* (1996), 1126–1129.
- [9] NVIDIA, ¿QUÉ ES EL GPU COMPUTING? www.nvidia.es/object/gpu-computing-es.html.
- [10] OPERTO, S., R. C. I. L. V. J., AND HERRERO, A. Quantitative imaging of complex structures from dense wide-aperture seismic data by

- multiscale traveltime and waveform inversions: a case study. *Geophysics* 52, 6 (2004), 625–651.
- [11] PASALIC, D., MCGARRY, R., ET AL. Convolutional perfectly matched layer for isotropic and anisotropic acoustic wave equations. In *2010 SEG Annual Meeting* (2010), Society of Exploration Geophysicists.
- [12] PLESSIX, R.-E. A review of the adjoint-state method for computing the gradient of a functional with geophysical applications. *Geophysical Journal International* 167, 2 (2006), 495–503.
- [13] SAVA, P., AND BIONDI, B. Wave-equation migration velocity analysis. i. theory. geophysical prospecting. *Geophysics* 52, 6 (2004), 593–606.
- [14] SCHLUMBERGER. OILFIELD GLOSSARY. www.glossary.oilfield.slb.com.
- [15] SIRGHE, L., AND PRATT, R. Efficient waveform inversion and imaging: A strategy for selecting temporal frequencies. *Geophysics* 69 (2004), 231–248.
- [16] SOLANO, C. A. P. *Two-dimensional near-surface seismic imaging with surface waves: alternative methodology for waveform inversion*. PhD thesis, Ecole Nationale Supérieure des Mines de Paris, 2013.
- [17] TARANTOLA, A. Inversion of seismic-reflection data in the acoustic approximation. *Geophysics* (1984), 1259–1266.
- [18] TESLA K40 GPU ACCELERATOR - NVIDIA. www.nvidia.com/content/PDF/kepler/nvidia-tesla-k40.pdf.
- [19] VIRIEUX, J., AND OPERTO, S. An overview of full-waveform inversion in exploration geophysics. *Geophysics* 74, 6 (2009), WCC1–WCC26.