UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

PROGRAMA DE MAESTRÍA Y DOCTORADO EN CIENCIAS MATEMÁTICAS Y DE LA ESPECIALIZACIÓN EN ESTADÍSTICA APLICADA

TOMOGRAFÍA SÍSMICA E INVERSIÓN DE FORMAS DE ONDA USANDO MÉTODOS DE OPTIMIZACIÓN EMPLEADOS POR LAS REDES NEURONALES ARTIFICIALES

TESIS
QUE PARA OPTAR POR EL GRADO DE:
DOCTOR EN CIENCIAS

PRESENTA:
MARCOS BERNAL ROMERO

DIRECTOR DE LA TESIS:
Dra. URSULA XIOMARA ITURRARÁN VIVEROS
FACULTAD DE CIENCIAS, UNAM

MIEMBROS DEL COMITÉ TUTOR:

Dr. FRANCISCO JOSÉ SÁNCHEZ SESMA
INSTITUTO DE INGENIERÍA, UNAM

Dr. TOMÁS MORALES ACOLTZI
CENTRO DE CIENCIAS DE LA ATMÓSFERA, UNAM

CIUDAD DE MÉXICO, 18 DE JUNIO DE 2021.

Para Alejandra

"No nos atrevemos a muchas
cosas porque son difíciles,
pero son difíciles porque no
nos atrevemos a hacerlas."
-Séneca.

Agradecimientos

Primeramente agradezco a Dios por la vida y por hacerse siempre presente en mis dificultades; a mis padres, que sin darse cuenta me han mostrado su amor a través de su acciones y sacrificios; a mi esposa Alejandra, por el apoyo, amor y comprensión que me ha dado, haciendo que mis días de trabajo sean más livianos.

Agradezco a mi tutora, la Dra. Ursula Iturrarán Viveros, por el apoyo brindado a lo largo de este trabajo y por atenderme incluso en días no laborables. A los miembros del comité tutor que siempre estuvieron dispuestos a apoyarme. Me llena de orgullo concer y haber tomado clase con el Dr. Franciso José Sánchez Sesma, pues su trayectoria es un ejemplo a seguir; de igual manera agradezco al Dr. Tomás Morales Acoltzi, por su amistad, sus consejos y haberme dado la oportunidad de seguirlo a varios eventos.

Agradezco a los revisores: Dr. Josué Tago Pacheco, Dr. Alessio Franci y Dr. Lorenzo Héctor Juárez Valencia, por haberse tomado el tiempo de revisar este trabajo pues sus comentarios y sugerencias me permitieron mejorarlo.

Agradezco a todas las personas que de alguna forma pusieron su granito de arena para darme ideas de cómo enfrentarme a un problema que para mí era totalmente nuevo.

Es un orgullo haber formado parte de la mejor universidad pública de México, la UNAM. Finalmente, agradezco al CONACYT por la beca que me otorgó durante mis estudios de doctorado.

Índice general

"Solving an inverse problem means to describe
the infinite-dimensional space of data-fitting models"
-George Backus & Freeman Gilbert, 1968.

Introducción

Desde el invento del sismógrafo (en 1842 por el físico escocés James David Forbes), el hombre ha sido capaz de registrar los movimientos que se propagan en el interior de la tierra, los cuales se producen ya sea por fuentes de energía liberada de forma natural (como los sismos) o fuentes de energía generadas por el hombre en algún proceso de exploración como se muestra en la figura 1. Dicha energía se manifiesta mediante ondas que se propagan en el subsuelo y que al chocar con distintos materiales generan reflexiones que son registradas por una serie de receptores (geófonos). La colección del registro de energía en cada receptor es un conjunto de trazas o formas de onda conocido como sismograma (figura 2). Los sismogramas contienen una gran cantidad de información acerca de la estructura y propiedades físicas del subsuelo, sin embargo, reconstruir dichas propiedades a partir de los sismogramas (un problema inverso) hoy en día sigue siendo un problema retador (Virieux and Operto, 2009). Uno de los primeros intentos para extraer información sobre la estructura del susbuelo a partir de los sismogramas dio lugar a la migración en tiempo reverso o RTM (McMechan, 1983; Whitmore, 1983; Baysal et al., 1983); dicha técnica sólo permite construir modelos de reflectividades, es decir, imágenes que describen burdamente la estructura del subsuelo sin conocer sus propiedades fiísicas. Una generalización iterativa de esta técnica que permite obtener modelos de reflectividad con mayor resolución es la migración por mínimos cuadrados o LSM (Tarantola, 1984; Nemeth et al., 1999), sin embargo, a finales de la década de los 80's Albert Tarantola propone una técnica capaz de aproximar las propiedades físicas y su distribución en cada punto de la región del subsuelo en cuestión (Tarantola, 1986, 1987), lo que permite obtener un modelo de parámetros físicos que explica el comportamiento de los sismogramas. Con el avance de la tecnología se ha ido mejorando en los últimos años y hoy en día es una técnica de tomografía sísmica conocida como Inversión de Forma de Onda Completa o FWI (del inglés: Full Waveform Inversion).

La construcción de modelos de parámetros físicos del susbsuelo a partir de sismogramas, es un problema joven y seguirá siendo un reto durante la siguiente década (Operto et al., 2013). La importancia y aplicaciones que conlleva conocer la estructura y propiedades físicas del interior de la tierra, van desde la detección de fallas para la prevención de riesgos, hasta la exploración de hidrocarburos, entre otras (Treitel and Lines, 2001).

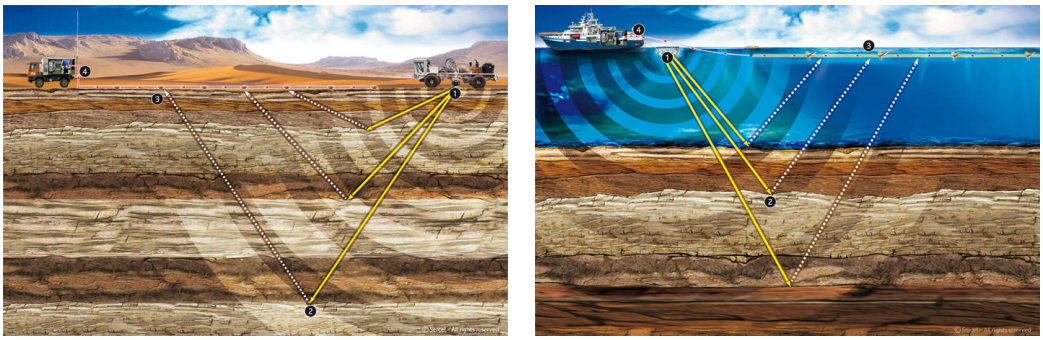

Figura 1: Ejemplos de campos de exploración para la extracción de datos [Figura de, https://ingenierosgeofisicos.blogspot.com/2016/06/en-que-consiste-la-reflexion-sismica.html].

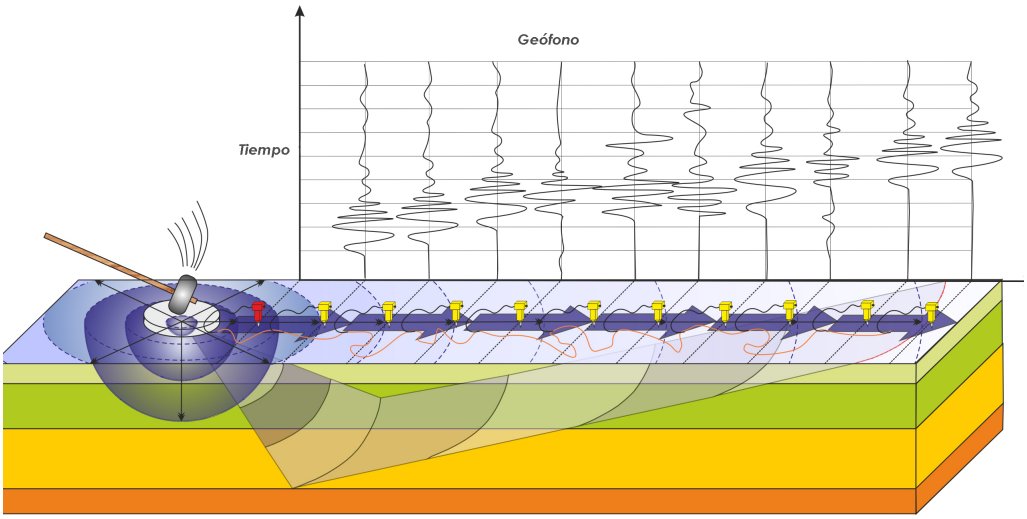

Figura 2: Ejemplo de una fuente externa generando un impulso que se propaga en el subsuelo y se registra en geófonos para obtener un sismograma [Figura de, https://terraestable.com/masw.php].

La FWI representa un problema inverso de gran escala y se caracteriza por ser un proceso altamente costoso computacionalmente, pues generalmente se requiere cómputo de alto rendimiento para procesar y modelar las grandes cantidades de datos que se obtienen en los casos reales (Schuster, 2017), por lo que nuestro objetivo principal se enfocará en implementar nuevas estrategias que reduzcan el costo computacional, combinándolas con recientes métodos de optimización que superan el popular método L-BFGS (un método quasi-Newton altamente usado en FWI, que mostraremos en la Sección 5.0.1) y ofrecen mejores resultados, es decir, modelos de parámetros físicos más precisos.

Como el objetivo de la FWI es aproximar modelos de parámetros físicos $\vec{m}$ que expliquen los datos reales (u observados) d_{obs}, debemos ser capaces de construir datos aproximados d_{aprx} comparables con d_{obs}; así que dedicaremos el Capítulo 1 para mostrar la construcción e implementación numérica del modelo directo L (ecuaciones que gobiernan el fenómeno de propagación de ondas), que nos ayudará a generar los datos aproximados mediante $L(\vec{m}) = d_{aprx}$. En el Capítulo 2 revisaremos la teoría de la FWI en el dominio del tiempo y mostraremos la técnica de fuentes simultáneas dinámicas que nos permite acelerar el proceso del cálculo del gradiente. Veremos que la FWI se traduce en un problema de minimización, por lo que su eficiencia computacional también depende del método de optimización que se aplique, así que dedicaremos el Capítulo 3 para estudiar nuevos y poderosos métodos de optimización (algunos desarrollados por Google Inc.) que han sido altamente usados como optimizadores de caja negra para el entrenamiento de redes neuronales artificiales (RNA) de gran escala y resultan ser más eficientes que los métodos quasi-Newton. Aquí cabe mencionar que hemos aplicado directamente los mejores métodos de optimización que se usan para entrenar las redes neuronales artificiales. La aplicación de estos métodos de optimización en FWI requieren una forma especial de calcular el step-length, así que dedicaremos el Capítulo 4 para mostrar una nueva fórmula para este proceso. Finalmente, en el Capítulo 5 mostramos los resultados de nuestros experimentos numéricos, restringidos al caso acústico de FWI basada en las normas L_1 y L_2, que validan la efectividad de las nuevas estrategias que proponemos.

Capítulo 1

Modelado directo de la Propagación de Ondas

El objetivo de la modelación matemática es describir y/o aproximar algún fenómeno de la vida real mediante herramientas matemáticas que permitan simular su comportamiento, por ejemplo, si tenemos un conjunto de sismogramas reales es porque provienen del registro de algún fenómeno físico de propagación de ondas y dado que nuestro objetivo es conocer las propiedades físicas del subsuelo que expliquen dichos datos reales, debemos ser capaces de simular la propagación de ondas para generar datos sintéticos que sean comparables con los datos reales. Para esto nos auxiliaremos de la teoría de la elastodinámica para entender, a grandes rasgos, la construcción de los modelos que gobiernan la propagación de ondas que usaremos en este trabajo. Para más detalles puede consultar, Aki and Richards (2009); Stein and Wysession (2003).

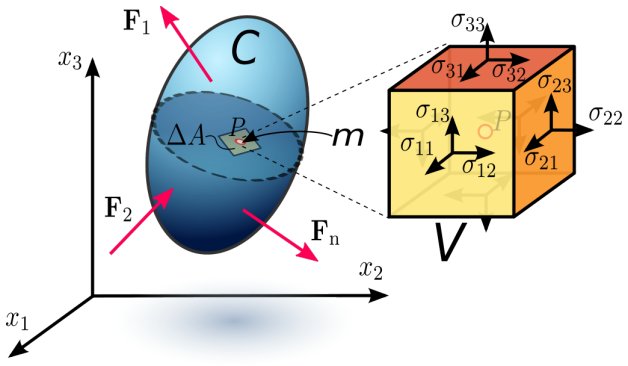

Figura 1.1: Medio continuo, un elemento del medio y su tensor de esfuerzos.

Consideremos un medio continuo C y tomemos un elemento de masa m y volumen V, como en la figura 1.1. El estado de equilibrio de dicho elemento está gobernado por la segunda ley de Newton para medios continuos, la cual establece que,

$$\overbrace{\int_{\partial V} \vec{t} dS}^{resultante\ de\ fuerzas\ superficiales} + \overbrace{\int_V \vec{f} dV}^{resultante\ de\ fuerzas\ de\ cuerpo} = \int_M \vec{a} dm, \quad (1.1)$$

donde $\vec{f}$ es una fuerza de cuerpo, $\vec{t}$ es el vector de tracciones (ecuación (1.3)) y $\vec{a} = \frac{\partial^2 \vec{u}}{\partial t^2}$ es la aceleración de las partículas del elemento cuyo campo de desplazamientos es,

$$\vec{u} = (u_x(x, y, z, t), u_y(x, y, z, t), u_z(x, y, z, t)).$$

Debido a que la densidad se definie como $\rho = m/V$ (masa contenida por unidad de volumen), entonces $dm = \rho dV$, por lo cual,

$$\int_{\partial V} \vec{t} dS + \int_V \vec{f} dV = \int_V \rho \vec{a} dV. \quad (1.2)$$

Usando la ley de Cauchy que relaciona el tensor de esfuerzos σ con el vector de tracciones $\vec{t}$ mediante,

$$\vec{t} = \sigma \vec{n} \quad \rightarrow \quad t_i = \sigma_{ij} n_j, \quad (1.3)$$

donde $\vec{n}$ es el vector normal unitario y usando el teorema de la divergencia,

$$\int_{\partial V} \sigma_{ij} n_j dS = \int_V \frac{\partial \sigma_{ij}}{\partial x_j} dV, \quad (1.4)$$

podemos escribir la ecuación (1.2) como,

$$\int_V \left(\frac{\partial \sigma_{ij}}{\partial x_j} + f_i \right) dV = \int_V \rho a_i dV, \quad (1.5)$$

y dado que el volumen V fue arbitrario, tenemos que,

$$\frac{\partial \sigma_{ij}}{\partial x_j} + f_i = \rho a_i, \quad (1.6)$$

es decir,

$$\rho \frac{\partial^2 \vec{u}}{\partial t^2} = \nabla \cdot \sigma + \vec{f}. \quad (1.7)$$

La ecuación (1.7) corresponde a la ecuación de movimiento para medios continuos.

1.1. La Ecuación de Navier

El medio continuo con el que estaremos trabajando debe poseer propiedades elásticas semejantes a las del susbsuelo. La relación de elasticidad entre los desplazamientos $\vec{u}$, de las partículas del medio y el tensor de esfuerzos σ, está dada por la ley de Hooke para medios elásticos isótropos,

$$\sigma_{ij} = \lambda \frac{\partial u_k}{\partial x_k}\delta_{ij} + \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right), \tag{1.8}$$

donde δ_{ij} es la delta de Kronecker; λ y μ son los parámetros de Lamé, que caracterizan por completo el comportamiento elástico lineal de un medio isótropo bajo pequeñas deformaciones.

Sustituyendo la Ley de Hooke (1.8) en la ecuación de movimiento (1.7), obtenemos la Ecuación de Navier,

$$\rho\frac{\partial^2 u_i}{\partial t^2} = \lambda\frac{\partial^2 u_k}{\partial x_k \partial x_j}\delta_{ij} + \mu\left(\frac{\partial^2 u_i}{\partial x_j \partial x_j} + \frac{\partial^2 u_j}{\partial x_i \partial x_j}\right) + f_i, \tag{1.9}$$

que en su forma cartesiana se puede escibir como,

$$\rho(\vec{x})\frac{\partial^2 \vec{u}(\vec{x},t)}{\partial t^2} = \mu(\vec{x})\nabla^2\vec{u}(\vec{x},t) + (\lambda(\vec{x}) + \mu(\vec{x}))\nabla(\nabla\cdot\vec{u}(\vec{x},t)) + \vec{f}(\vec{x},t), \tag{1.10}$$

que corresponde a la ecuación de movimiento para un medio isótropo elástico que gobierna el fenómeno de propagación de ondas sísmicas.

Usando la identidad,

$$\nabla^2\vec{u} = \nabla(\nabla\cdot\vec{u}) - \nabla\times(\nabla\times\vec{u}), \tag{1.11}$$

podemos escribir la ecuación de Navier (1.10) como,

$$\rho\frac{\partial^2 \vec{u}(\vec{x},t)}{\partial t^2} = (\lambda+2\mu)\nabla(\nabla\cdot\vec{u}(\vec{x},t)) - \mu\nabla\times(\nabla\times\vec{u}(\vec{x},t)). \tag{1.12}$$

La solución de la ecuación (1.12) se puede descomponer en términos de un potencial escalar $\phi(\vec{x},t)$ (ondas P) y un potencial vectorial $\vec{\gamma}(\vec{x},t)$ (ondas S), mediante

$$\vec{u}(\vec{x},t) = \nabla\phi(\vec{x},t) + \nabla\times\vec{\gamma}(\vec{x},t), \tag{1.13}$$

sustituyendo esta descomposición en la ecuación (1.12) y usando las identidades,

$$\nabla\times(\nabla\phi) = 0 \text{ y } \nabla\cdot(\nabla\times\vec{\gamma}) = 0,$$

tenemos que,

$$(\lambda + 2\mu)\nabla(\nabla^2\phi) - \mu\nabla \times \nabla \times (\nabla \times \vec{\gamma}) = \rho\frac{\partial^2}{\partial t^2}(\nabla\phi + \nabla \times \vec{\gamma}). \qquad (1.14)$$

Usando la identidad en la ecuación (1.11), el segundo término de la ecuación (1.14) se simplifica como,

$$\begin{aligned} \nabla \times \nabla \times (\nabla \times \vec{\gamma}) &= -\nabla^2(\nabla \times \vec{\gamma}) + \nabla(\nabla \cdot (\nabla \times \vec{\gamma})) \\ &= -\nabla^2(\nabla \times \vec{\gamma}), \end{aligned} \qquad (1.15)$$

sustituyendo la ecuación (1.15) en la ecuación (1.14), tenemos,

$$\nabla\left[(\lambda + 2\mu)\nabla^2\phi(\vec{x}, t) - \rho\frac{\partial^2\phi(\vec{x}, t)}{\partial t^2}\right] = -\nabla \times \left[\mu\nabla^2\vec{\gamma}(\vec{x}, t) - \rho\frac{\partial^2\vec{\gamma}(\vec{x}, t)}{\partial t^2}\right]. \qquad (1.16)$$

Se puede encontrar una solución de la ecuación de Navier, si ambos lados de la ecuación (1.16) son cero. En este caso tenemos una ecuación de onda para el potencial escalar $\phi(\vec{x}, t)$ y otra para el potencial vectorial $\vec{\gamma}(\vec{x}, t)$:

$$\nabla^2\phi(\vec{x}, t) = \frac{1}{\nu^2}\frac{\partial^2\phi(\vec{x}, t)}{\partial t^2}, \qquad (1.17)$$

$$\nabla^2\vec{\gamma}(\vec{x}, t) = \frac{1}{\beta^2}\frac{\partial^2\vec{\gamma}(\vec{x}, t)}{\partial t^2}. \qquad (1.18)$$

Las soluciones ϕ y $\vec{\gamma}$, corresponden a las ondas P (o compresionales) y a las ondas S (o cortantes), respectivamente y sus correspondientes velocidades de propagación están dadas por,

$$\nu = \sqrt{\frac{\lambda + 2\mu}{\rho}} \qquad \text{y} \qquad \beta = \sqrt{\frac{\mu}{\rho}}. \qquad (1.19)$$

1.2. Ecuación de Onda para medios acústicos

En un medio acústico el módulo de corte es nulo, es decir, $\mu = 0$ así que la ley de Hooke, ecuación (1.8), se reduce a,

$$\sigma_{ij} = -p(\vec{x}, t)\delta_{ij}, \tag{1.20}$$

donde $p(\vec{x}, t)$ es la presión escalar dada por,

$$p(\vec{x}, t) = -\lambda(\vec{x})\nabla \cdot \vec{u}(\vec{x}, t). \tag{1.21}$$

Sustituyendo la ecuación (1.20) en la ecuación (1.7) obtenemos,

$$\frac{\partial^2 \vec{u}(\vec{x}, t)}{\partial t^2} + \frac{1}{\rho(\vec{x})}\nabla p(\vec{x}, t) = \frac{1}{\rho(\vec{x})}\vec{f}(\vec{x}, t). \tag{1.22}$$

Usando la definición de presión escalar, ecuación (1.21) en la divergencia de la ecuación (1.22), obtenemos,

$$-\frac{1}{\lambda(\vec{x})}\frac{\partial^2 p(\vec{x}, t)}{\partial t^2} + \nabla \cdot \left(\frac{1}{\rho(\vec{x})}\nabla p(\vec{x}, t)\right) = -\nabla \cdot \left(\frac{1}{\rho(\vec{x})}\vec{f}(\vec{x}, t)\right) \tag{1.23}$$

y suponiendo que $\rho(\vec{x})$ varía mucho más lento que $p(\vec{x}, t)$ y $\vec{f}(\vec{x}, t)$ entonces,

$$\nabla \cdot (\rho^{-1}\nabla p) \approx \rho^{-1}\nabla^2 p \qquad \text{y} \qquad \nabla \cdot (\rho^{-1}\vec{f}) \approx \rho^{-1}\nabla \cdot \vec{f}, \tag{1.24}$$

considerando esto último en la ecuación (1.23) obtenemos la **ecuación de onda para medios acústicos**,

$$\frac{\partial^2 p(\vec{x}, t)}{\partial t^2} - \nu^2\nabla^2 p(\vec{x}, t) = \nu^2\nabla \cdot \vec{f}, \tag{1.25}$$

para el campo de presión escalar $p(\vec{x}, t)$, con velocidad de propagación,

$$\nu = \sqrt{\frac{\lambda}{\rho}}.$$

1.3. Implementación Numérica

Con el fin de utilizar un esquema de malla alternada que garantice estabilidad numérica (Virieux, 1986; Levander, 1988) y poder implementar fronteras absorbentes que simulen la propagación de ondas en un medio semi-infinito, usaremos una formulación velocidad-esfuerzos de la ecuación (1.25). Así que usando el cambio de variables,

$$\begin{aligned} v &= \frac{\partial p}{\partial t}, \\ \sigma_{xx} &= \frac{\partial p}{\partial x}, \\ \sigma_{zz} &= \frac{\partial p}{\partial z}, \end{aligned} \tag{1.26}$$

obtenemos la formulación velocidad-esfuerzos de la ecuación (1.25),

$$\begin{aligned} \frac{\partial v}{\partial t} &= \nu^2 \left(\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{zz}}{\partial z} \right) + f(X_s), \\ \frac{\partial \sigma_{xx}}{\partial t} &= \frac{\partial v}{\partial x}, \\ \frac{\partial \sigma_{zz}}{\partial t} &= \frac{\partial v}{\partial z}, \\ v(X, t=0) &= \vec{\sigma}(X, t=0) = 0; f(X_s, t) = w(t)\delta(X - X_s), \end{aligned} \tag{1.27}$$

siendo $f(X_s)$ una fuente puntual actuando en las coordenadas $X_s = (x_s, z_s)$; δ es la delta de Dirac y $w(t)$ es la onduleta de la señal de la fuente (en este trabajo siempre usaremos el pulso de Ricker [1] con las frecuencias que se especifiquen en cada experimento).

Para la discretización del sistema de ecuaciones (1.27) usaremos el esquema de malla alternada propuesto por Levander (1988), en donde se aproximan las derivadas espaciales con diferencias finitas de cuarto orden y las derivadas temporales se aproximan con diferencias finitas centradas (de segundo orden). Este esquema supone que la aproximación de las derivadas espaciales y temporales considera incrementos de $\Delta x/2$ y $\Delta t/2$, respectivamente, lo que ofrece mayor precisión en la aproximación de las derivadas (Virieux, 1986).

[1] El pulso de Ricker es una onduleta de la forma, $r(t) = (a^2 - \frac{1}{2})e^{-a^2}$, con $a = \frac{\pi(t-t_s)}{t_p}$, donde t_p es el ancho del pulso y t_s es el retraso del pulso.

Para denotar que las ecuaciones (1.27) se implementan con un esquema de malla alternada usaremos subíndices y superíndices fraccionarios, por ejemplo:

$$v_{i+1,j}^{m+1/2} := v(x_i + \Delta x, z_j, t_m + \Delta t/2),$$

representa el valor de v en la coordenada espacio-temporal,

$$(x_i + \Delta x, z_j, t_m + \Delta t/2).$$

Por otro lado, una malla cuyas coordenadas son de la forma (por ejemplo) $(x_i + \Delta x/2, z_j, t_m + \Delta t/2)$, la denotaremos como la malla $(i+1/2, j, m+1/2)$.

La primer ecuación del sistema (1.27) se construye en la malla (i, j, m), la segunda ecuación se construye en la malla $(i + 1/2, j, m + 1/2)$ y la tercer ecuación se construye en la malla $(i, j + 1/2, m + 1/2)$ (véase la figura 1.2). Por lo tanto, la discretización del sistema (1.27) usando mallas alternadas se muestra en las ecuaciones (1.28-1.29).

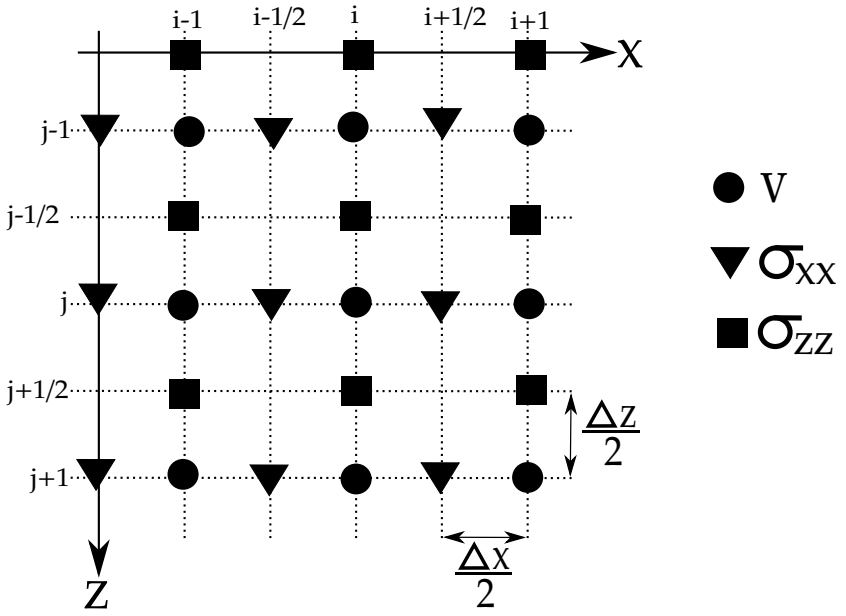

Figura 1.2: Esquema de discretización para el caso acústico.

Si la discretización de este problema en diferencias finitas se realizara en una malla convencional con incrementos espaciales Δx, podríamos aumentar la precisión de la solución al usar $\Delta x/2$ lo que implicaría un refinamiento de la malla que duplica el número de puntos y que a su vez eleva el costo computacional. Sin embargo, el esquema de malla alternada, al aproximar los campos de velocidades y esfuerzos en mallas independientes con incrementos $\Delta x/2$, permite aumentar la precisión sin duplicar el número de puntos de la malla espacial (Levander, 1988; Virieux, 1986). Esto es posible debido a que el sistema (1.27) es acoplado. Evidentemente la discretización en malla alternada es consistente pues si los incrementos espaciales y temporales tienden a cero, el problema numérico, ecuaciones (1.28 - 1.29), converge al problema continuo, ecuaciones (1.27).

$$\begin{aligned}
v_{i,j}^{m+1/2} &= v_{i,j}^{m-1/2} + \Delta t \nu_{i,j}^2 \left[\left(\frac{\partial \sigma_{xx}}{\partial x} \right)_{i,j}^{m} + \left(\frac{\partial \sigma_{zz}}{\partial z} \right)_{i,j}^{m} \right] + f, \\
{\sigma_{xx}}_{i+1/2,j}^{m+1} &= {\sigma_{xx}}_{i+1/2,j}^{m} + \Delta t \left(\frac{\partial v}{\partial x} \right)_{i+1/2,j}^{m+1/2}, \\
{\sigma_{zz}}_{i,j+1/2}^{m+1} &= {\sigma_{zz}}_{i,j+1/2}^{m} + \Delta t \left(\frac{\partial v}{\partial z} \right)_{i,j+1/2}^{m+1/2},
\end{aligned} \tag{1.28}$$

donde,

$$\begin{aligned}
\left(\frac{\partial \sigma_{xx}}{\partial x} \right)_{i,j}^{m} &= \frac{-1/24({\sigma_{xx}}_{i+3/2,j}^{m} - {\sigma_{xx}}_{i-3/2,j}^{m}) + 9/8({\sigma_{xx}}_{i+1/2,j}^{m} - {\sigma_{xx}}_{i-1/2,j}^{m})}{\Delta x}, \\
\left(\frac{\partial \sigma_{zz}}{\partial z} \right)_{i,j}^{m} &= \frac{-1/24({\sigma_{zz}}_{i,j+3/2}^{m} - {\sigma_{zz}}_{i,j-3/2}^{m}) + 9/8({\sigma z}_{i,j+1/2}^{m} - {\sigma_{zz}}_{i,j-1/2}^{m})}{\Delta z}, \\
\left(\frac{\partial v}{\partial x} \right)_{i+1/2,j}^{m+1/2} &= \frac{-1/24(v_{i+2,j}^{m+1/2} - v_{i-1,j}^{m+1/2}) + 9/8(v_{i+1,j}^{m+1/2} - v_{i,j}^{m+1/2})}{\Delta x}, \\
\left(\frac{\partial v}{\partial z} \right)_{i,j+1/2}^{m+1/2} &= \frac{-1/24(v_{i,j+2}^{m+1/2} - v_{i,j-1}^{m+1/2}) + 9/8(v_{i,j+1}^{m+1/2} - v_{i,j}^{m+1/2})}{\Delta z}.
\end{aligned} \tag{1.29}$$

Para asegurar estabilidad usamos la condición de Courant (Courant et al., 1928, 1967),

$$\Delta t \leq \frac{\Delta x}{h\sqrt{2}V_{max}}, \tag{1.30}$$

donde V_{max} es la velocidad máxima del medio y h es un factor que depende del orden del operador de diferencias finitas (FD). Moczo (1998) recomienda usar $h = 1$ para FD de segundo orden, $h = 7/6$ para FD de cuarto orden y $h = 149/120$ para FD de octavo orden.

Para reducir el error de dispersión numérica consideramos la condición,

$$\Delta x \leq \frac{\lambda_{min}}{n}, \tag{1.31}$$

donde n es el número de puntos en la mínima longitud de onda λ_{min}, la cual está dada en términos de la frecuencia máxima f_{max} y la velocidad mínima V_{min}, como:

$$\lambda_{min} = \frac{V_{min}}{f_{max}}. \tag{1.32}$$

Mediante un análisis de Von Neumann se obtiene que para diferencias finitas de segundo, cuarto y sexto orden obtenidas del truncamiento de la serie de Taylor, se debe usar $n = 12$, $n = 8$ y $n = 6$, respectivamente (Moczo, 1998).

En la figura 1.3 se muestran algunos snapshots del campo de velocidades obtenidos de la implementación computacional de las ecuaciones (1.28), donde la fuente f corresponde a un pulso de Ricker (de $40Hz$) que se ubica al centro de la malla, usando un modelo de velocidades homogéneo (de $5000m/s$). Podemos ver que a partir de cierto tiempo, en las fronteras del dominio se generan reflexiones que no son físicamente aceptables pues no corresponden a un medio semi-infinito. En la siguiente sección mostraremos una técnica muy poderosa para eliminar dichas reflexiones.

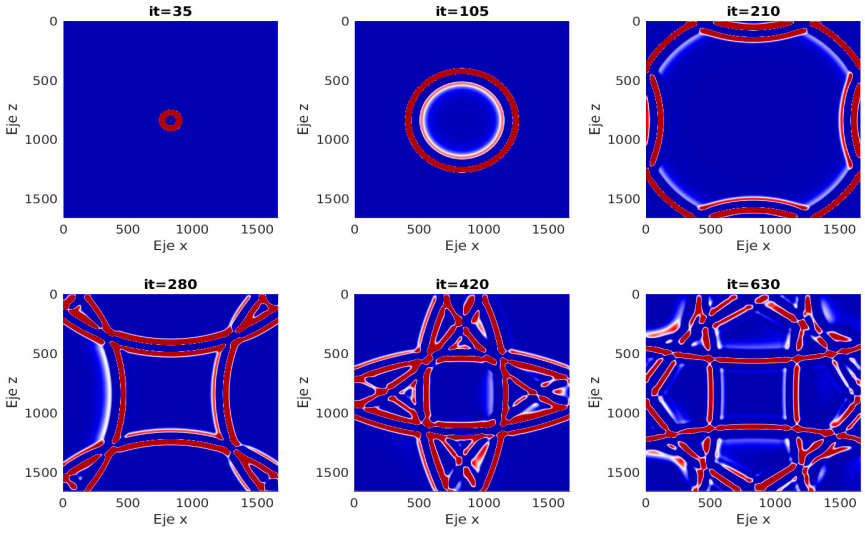

Figura 1.3: Snapshots del campo de velocidades para diferentes tiempos sin usar fronteras absorbentes.

1.3.1. Implementación de fronteras absorbentes C-PML

Para evitar las reflexiones que no son físicamente aceptables (como las que se muestran en la figura 1.3), extenderemos el dominio espacial agregando nodos en las fronteras de la malla, en donde se construyen parámetros que actúan como esponjas que absorben la energía de las ondas en dichas regiones, de tal forma que se pueda simular la propagación de ondas en un medio semi-infinito; para esto implementaremos las condiciones de fronteras absorbentes C-PML (del inglés, Convolutional Perfectly Matched Layers), propuestas por Komatitsch and Martin (2007).

Las condiciones de fronteras absorbentes C-PML son una variante de las fronteras absorbentes PML propuestas por Berenger (1994), quien inicialmente las desarrolló para usarlas en las ecuaciones de Maxwell. Desde entonces se han podido incorporar en distintos tipos de ecuaciones de onda como las elastodinámicas, ecuaciones de Helmholtz y de poroelasticidad. La teoría se desarrolla en el dominio de la frecuencia ω y mediante transformada Fourier inversa se implementa en el dominio del tiempo.

Suponiendo que la ecuación de onda está en el dominio de la frecuencia ω, la idea de esta técnica consiste en aplicar una transformación $(\hat{x}(x), \hat{z}(z))$ a cada coordenada espacial (x, z), de tal forma que disipe la energía que llega a las fronteras absorbentes y deje intacta la energía que viaja en el interior del dominio. Para la dirección horizontal (eje x) se aplicará la transformación,

$$\hat{x}(x) = x - \frac{i}{\omega}\int_0^x d_x(s)ds, \tag{1.33}$$

donde $d_x(x)$ es una función de amortiguamiento tal que $d_x(x) = 0$ en el interior del dominio y $d_x(x) \neq 0$ en las fronteras absorbentes, como se muestra en la figura 1.4. Dicha función de amortiguamiento controla la rapidez con que se disipa la energía que llega a la frontera absorbente. Berenger (1994) define la función de amortiguamiento como:

$$d_x(x) = \begin{cases} dx_0\left(\dfrac{x}{L_x}\right)^N & \text{si } x \in \{\text{región con C-PML}\}, \\ 0 & \text{en otro caso} \end{cases}$$

con,

$$dx_0 = -(N+1)V_p\frac{\log(R_c)}{2L_x}, \tag{1.34}$$

donde $L_x = q\Delta x$, con q el número de nodos del eje x en donde se definen las C-PML (véase la figura 1.5), V_p es el máximo de la velocidad de propagación; N es el orden de suavizamiento y R_c es un coeficiente de reflexión (con $0 < R_c < 1$).

Análogamente para la dirección vertical (eje z) se aplica una tranformación $\hat{z}(z)$ en términos de una función de amortiguamiento $d_z(z)$.

La transformación $\hat{x}(x)$ induce el operador diferencial $\frac{\partial(\cdot)}{\partial\hat{x}}$ (en el dominio de la frecuencia), dado por [2]:

$$\frac{\partial(\cdot)}{\partial\hat{x}} = \frac{1}{S_x(x,\omega)}\frac{\partial(\cdot)}{\partial x}, \tag{1.35}$$

donde,

$$S_x(x,\omega) = 1 + \frac{d_x(x)}{i\omega}. \tag{1.36}$$

[2]Si $(\cdot)$ es función de $\hat{x}(x)$, por la regla de la cadena, $\frac{\partial(\cdot)}{\partial x} = \frac{\partial(\cdot)}{\partial\hat{x}}\frac{\partial\hat{x}}{\partial x}$, de donde, $\frac{\partial(\cdot)}{\partial\hat{x}} = \frac{1}{\frac{\partial\hat{x}}{\partial x}}\frac{\partial(\cdot)}{\partial x}$. Como $\frac{\partial\hat{x}}{\partial x} = 1 - \frac{i}{\omega}d_x(x) = 1 + \frac{d_x(x)}{i\omega} = S_x(x,\omega)$, entonces $\frac{\partial(\cdot)}{\partial\hat{x}} = \frac{1}{S_x(x,w)}\frac{\partial(\cdot)}{\partial x}$.

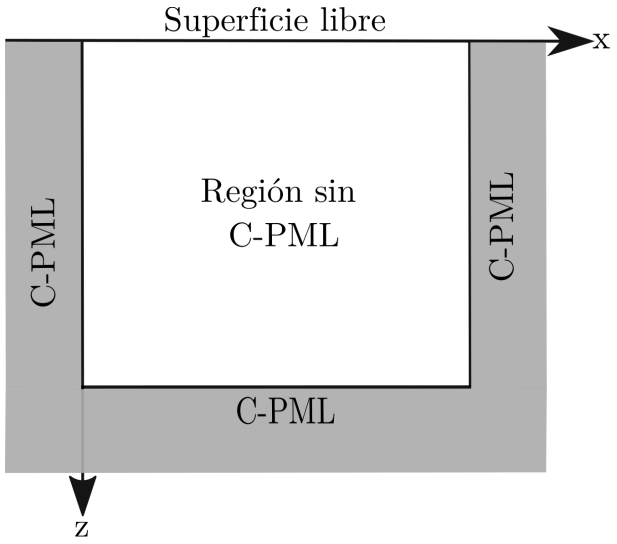

Figura 1.4: Extensión de la malla que incluye fronteras absorbentes.

La diferencia principal entre la formulación PML y C-PML es que esta última generaliza la ecuación (1.36) introduciendo los parámetros, $k_x(x) \geq 1$ y $\alpha_x(x) \geq 0$, de tal forma que:

$$S_x(x,\omega) = k_x(x) + \frac{d_x(x)}{\alpha_x(x) + i\omega}, \tag{1.37}$$

por lo que,

$$\frac{1}{S_x(x,\omega)} = \frac{1}{k_x(x)} - \frac{d_x(x)}{[k_x(x)]^2} \frac{1}{[\frac{d_x(x)}{k_x(x)} + \alpha_x(x)] + i\omega}. \tag{1.38}$$

Nos interesa obtener el operador $\frac{\partial(\cdot)}{\partial \hat{x}}$ en el dominio del tiempo, así que aplicamos transformada de Fourier inversa a la ecuación (1.35) y obtenemos [3]:

$$\frac{\partial(\cdot)}{\partial \hat{x}} = S_x^{'}(x,t) * \frac{\partial(\cdot)}{\partial x}, \tag{1.39}$$

donde,

$$S_x^{'}(x,t) = \mathcal{F}^{-1}\left\{\frac{1}{S_x(x,\omega)}\right\}(t). \tag{1.40}$$

[3]El símbolo $*$ representa el operador de convolución, de aquí el nombre C-PML (Convolutional Perfectly Matched Layers). La convolución entre dos funciones f y g, se define como la integral del producto de ambas funciones después de desplazar una de ellas una distancia t, es decir: $(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau$.

Usando las siguientes propiedades de la transformada de Fourier inversa:

$$\mathcal{F}^{-1}\{1\}(t) = \delta(t),$$

$$\mathcal{F}^{-1}\left\{\frac{1}{i\omega + a}\right\}(t) = e^{-at}H(t),$$

donde $\delta(t)$ y $H(t)$ son la delta de Dirac y la función de Heaviside, respectivamente, obtenemos:

$$S_x^{'}(x,t) = \frac{\delta(t)}{k_x(x)} - \frac{d_x(x)}{[k_x(x)]^2}H(t)e^{-[\frac{d_x(x)}{k_x(x)}+\alpha_x(x)]t}. \tag{1.41}$$

Sustituyendo la ecuación (1.41) en la ecuación (1.39) y realizando la operación de convolución, obtenemos el operador diferencial de la transformación $\hat{x}(x)$ en el dominio del tiempo, dado por:

$$\frac{\partial(\cdot)}{\partial\hat{x}} = \frac{1}{k_x(x)}\frac{\partial(\cdot)}{\partial x} + \psi_x(x,t), \tag{1.42}$$

donde,

$$\psi_x(x,t) = -\left\{\frac{d_x(x)}{[k_x(x)]^2}H(t)e^{-[\frac{d_x(x)}{k_x(x)}+\alpha_x(x)]t}\right\} * \frac{\partial(\cdot)}{\partial x}. \tag{1.43}$$

Al derivar la ecuación (1.43) con respecto al tiempo, obtenemos:

$$\frac{\partial\psi_x(x,t)}{\partial t} = -\left(\frac{d_x(x)}{k_x(x)} + \alpha_x(x)\right)^t \psi_x(x,t) - \frac{d_x(x)}{[k_x(x)]^2}\left(\frac{\partial(\cdot)}{\partial x}\right)^t, \tag{1.44}$$

donde los super-índices t representan la evaluación del operador en el tiempo t. La ecuación (1.44) muestra la recursividad de la derivada temporal de $\psi_x(x,t)$. Por lo tanto, la discretización de la ecuación (1.42) en el m-ésimo tiempo está dada por:

$$\left(\frac{\partial(\cdot)}{\partial\hat{x}}\right)^m = \left(\frac{1}{k_x(x)}\frac{\partial(\cdot)}{\partial x}\right)^m + \psi_x^m(x), \tag{1.45}$$

donde,

$$\psi_x^m(x) = b_x(x)\psi_x^{m-1/2}(x) + a_x(x)\left(\frac{\partial(\cdot)}{\partial x}\right)^m, \tag{1.46}$$

con,

$$b_x(x) = \exp\left(-\left[\frac{d_x(x)}{k_x(x)} + \alpha_x(x)\right]\Delta t\right), \tag{1.47}$$

$$a_x(x) = \left(\frac{d_x(x)}{k_x(x)\left[d_x(x) + \alpha_x(x)k_x(x)\right]}\right)(b_x(x) - 1) \tag{1.48}$$

y $\alpha_x(x)$ es una función lineal que sólo toma valores en las regiones de las fronteras absorbentes, la cual está dada por,

$$\alpha_x(x) = \begin{cases} \left(\dfrac{\pi f_{max}}{x_{r+1} - x_r}\right)(x - x_r), & \text{si } x_r \leq x \leq x_{r+1}, \\ \left(\dfrac{\pi f_{max}}{x_l - x_{l+1}}\right)(x - x_{l+1}), & \text{si } x_l \leq x \leq x_{l+1}\ , \end{cases}$$

donde f_{max} es la frecuencia máxima de la fuente; x_r y x_{r+1} son las coordenadas de los extremos de la región derecha con nodos absorbentes; x_l y x_{l+1} son las coordenadas de los extremos de la región izquierda con nodos absorbentes, como se muestra en la figura 1.5.

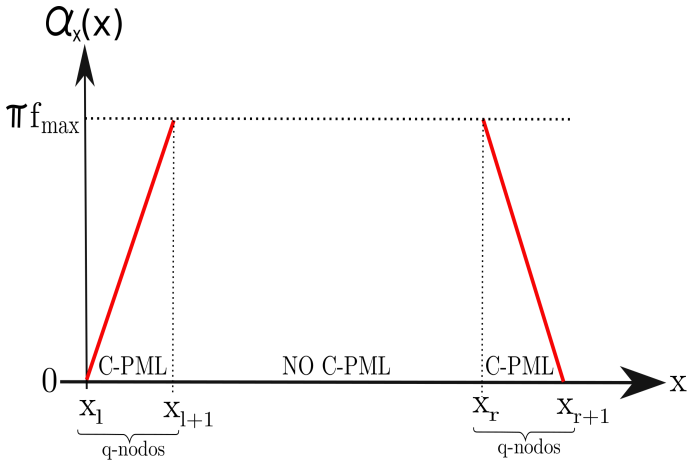

Figura 1.5: Función $\alpha_x(x)$ tomando valores en los nodos de las fronteras absorbentes.

Análogamente se construye el operador $\left(\frac{\partial(\cdot)}{\partial \hat{z}}\right)^m$ correspondiente a la transformación $\hat{z}(z)$ que se aplica en las coordenadas de la dirección vertical.

Aplicando los operadores $\left(\frac{\partial(\cdot)}{\partial \hat{x}}\right)^m$ y $\left(\frac{\partial(\cdot)}{\partial \hat{z}}\right)^m$, con $k_x(x) = k_z(z) = 1$, en las derivadas espaciales de las ecuaciones (1.28), obtenemos las ecuaciones que incluyen las condiciones de fronteras absorbentes C-PML,

$$
\begin{aligned}
v_{i,j}^{m+1/2} &= v_{i,j}^{m-1/2} + \Delta t \nu_{i,j}^2 \left[\left(\frac{\partial \sigma_{xx}}{\partial x} \right)_{i,j}^{m} + (\psi_x \sigma_{xx})_{i,j}^{m} + \left(\frac{\partial \sigma_{zz}}{\partial z} \right)_{i,j}^{m} + (\psi_z \sigma_{zz})_{i,j}^{m} \right] + f, \\
{\sigma_{xx}}_{i+1/2,j}^{m+1} &= {\sigma_{xx}}_{i+1/2,j}^{m} + \Delta t \left[\left(\frac{\partial v}{\partial x} \right)_{i+1/2,j}^{m+1/2} + (\psi_x v)_{i+1/2,j}^{m+1/2} \right], \\
{\sigma_{zz}}_{i,j+1/2}^{m+1} &= {\sigma_{zz}}_{i,j+1/2}^{m} + \Delta t \left[\left(\frac{\partial v}{\partial z} \right)_{i,j+1/2}^{m+1/2} + (\psi_z v)_{i,j+1/2}^{m+1/2} \right],
\end{aligned}
\tag{1.49}
$$

donde las derivadas espaciales de v, σ_{xx} y σ_{zz} corresponden a las ecuaciones (1.29) y

$$
\begin{aligned}
(\psi_x \sigma_{xx})_{i,j}^{m} &= b_x(x)(\psi_x \sigma_{xx})_{i,j}^{m-1/2} + a_x(x) \left(\frac{\partial \sigma_{xx}}{\partial x} \right)_{i,j}^{m}, \\
(\psi_z \sigma_{zz})_{i,j}^{m} &= b_z(z)(\psi_z \sigma_{zz})_{i,j}^{m-1/2} + a_z(z) \left(\frac{\partial \sigma_{zz}}{\partial z} \right)_{i,j}^{m}, \\
(\psi_x v)_{i+1/2,j}^{m+1/2} &= b_x(x)(\psi_x v)_{i+1/2,j}^{m} + a_x(x) \left(\frac{\partial v}{\partial x} \right)_{i+1/2,j}^{m+1/2}, \\
(\psi_z v)_{i,j+1/2}^{m+1/2} &= b_z(z)(\psi_z v)_{i,j+1/2}^{m} + a_z(z) \left(\frac{\partial v}{\partial z} \right)_{i,j+1/2}^{m+1/2}
\end{aligned}
\tag{1.50}
$$

En la figura 1.6 se muestran algunos snapshots del campo de velocidades obtenidos de la implementación computacional de las ecuaciones (1.49), donde la fuente f corresponde a un pulso de Ricker (de $40Hz$) que se ubica al centro de la malla, usando un modelo de velocidades homogéneo (de $5000m/s$) e implementando la condición de superficie libre mediante la construcción de una simetría par con respecto a los nodos de la superficie, donde $v(\text{superficie libre}) = 0$. Para los parámetros de la función de amortiguamiento, en la ecuación (1.34) usamos $N = 2$ y $R_c = 0{,}00005$; el número de nodos en cada frontera absorbente es $q = 25$.

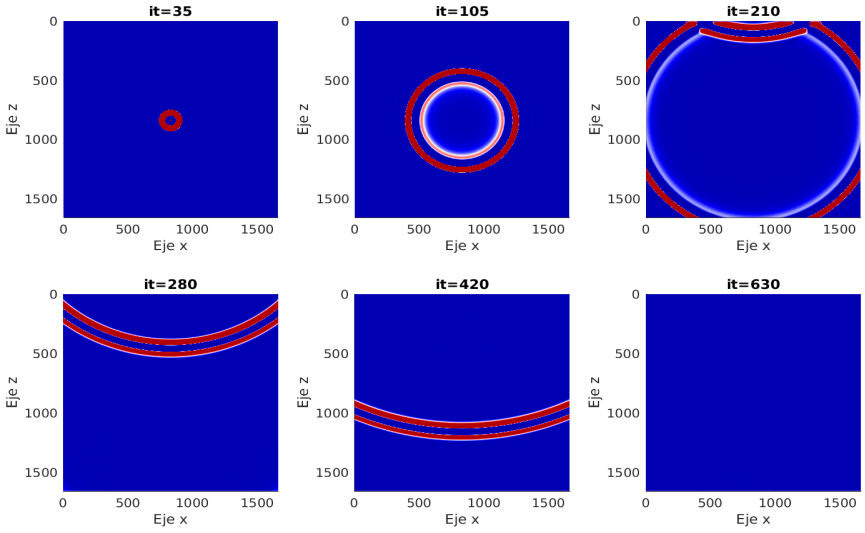

Figura 1.6: Snapshots del campo de velocidades para diferentes tiempos usando fronteras absorbentes con superficie libre.

1.3.2. Validación del modelo directo

Uno de los resultados más importantes del análisis numérico en las ecuaciones diferenciales parciales es el *Teorema de equivalencia de Lax*, el cual afirma que si un problema de valor inicial es un problema bien planteado (es decir, tiene solución única, la cual depende continuamente de las condiciones iniciales) y tiene una discretización en diferencias finitas, consistente y estable, entonces la solución numérica converge a la solución analítica.

Notemos que si $\Delta t \to 0$ y $\Delta x \to 0$, entonces el esquema en diferencias finitas, ecuaciones (1.28 - 1.29), converge al problema continuo, ecuaciones (1.27), es decir, el problema es consistente y por la condición de Courant que estamos usando, ecuación (1.30), el problema es estable. Por lo tanto la solución numérica que obtenemos con este esquema de diferencias finitas converge a la solución analítica. Con el objetivo de verificar esta última afirmación, calcularemos una solución analítica para mostrar que nuestra implementación numérica produce resultados correctos.

Debido a que las ecuaciones (1.27) tienen la misma estructura que la formulación velocidad-esfuerzos del caso SH (ecuaciones que modelan la propagación de ondas S, es decir, ondas cuyo movimiento de partículas es per-

pendicular a la dirección de propagación), para el cual se puede calcular una solución analítica con la función de Green (en el dominio de la frecuencia ω) que se muestra en Kausel (2006) (pág. 69):

$$g_{yy}(r_1, r_2, \omega) = -\frac{i}{4\nu^2}\left[H_0^{(2)}(\frac{\omega r_1}{\nu}) + H_0^{(2)}(\frac{\omega r_2}{\nu})\right]. \tag{1.51}$$

Dicha función de Green se obtiene con el método de las imágenes (para incluir el efecto de superficie libre), el cual incorpora una fuente simétrica (ficticia) con respecto a la superficie libre, donde r_1 es la distancia de la superficie libre a la fuente principal y r_2 es la distancia de la superficie libre a la fuente simétrica. $H_0^{(2)}$ corresponde a la función de Hankel de segunda especie, siendo ν la velocidad de propagación y ω la frecuencia angular.

Considerando la función de Green dada en la ecuación (1.51) y alguna onduleta $W(t)$ como señal de la fuente, el procedimiento para obtener la solución analítica es el siguiente:

1. Pasamos $W(t)$ al dominio de la frecuencia usando transformada de Fourier (fft) para aproximar el valor de, $W(\omega) = \mathcal{F}\{W(t)\}$.

2. Calculamos la convolución, $C(\omega, \vec{x}) = W(\omega)g_{yy}(r_1, r_2, \omega)$, con sus respectivas simetrías en el dominio de la frecuencia.

3. Obtenemos la solución en el dominio del tiempo aplicando la transformada de Fourier inversa (ifft) para aproximar el campo de desplazamientos, $\vec{u}(\vec{x}, t) = \mathcal{F}^{-1}\{C(\omega, \vec{x})\}$.

En la figura 1.7 se muestran las coordenadas de la fuente (pulso de Ricker) y receptores que se usaron con un modelo de velocidades homogéneo para comparar la solución analítica, obtenida con la función de Green, ecuación (1.51) y la solución numérica con fronteras absorbentes C-PML, ecuaciones (1.49-1.50). La comparación de sismogramas obtenidos con ambas soluciones se muestran en la figura 1.8.

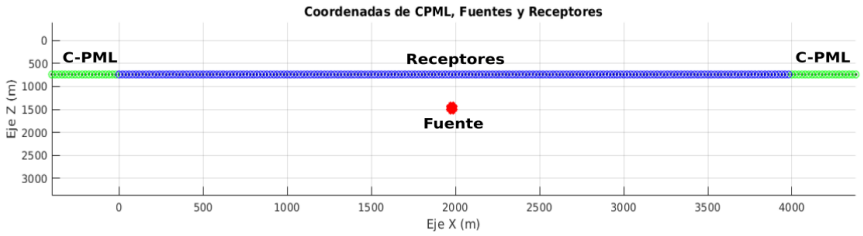

Figura 1.7: Coordenadas de fuente, receptores y nodos con C-PML.

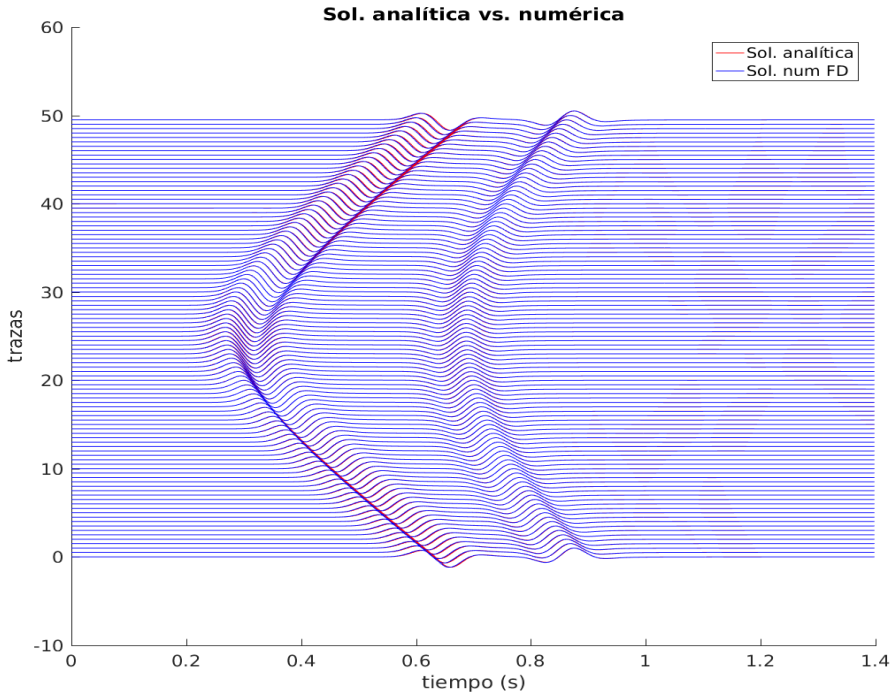

Figura 1.8: Sismogramas generados con la solución analítica (rojo) y con la solución numérica (azul).

En la figura 1.9 se puede observar que cuando $\Delta x \to 0$, la diferencia (usando la norma L_2) entre la solución numérica y la solución análitica se hace nula.

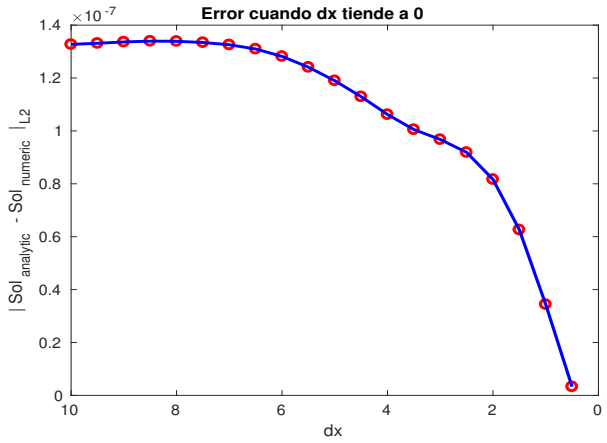

Figura 1.9: Evolución del error entre la solución analítica y la solución numérica cuando $\Delta x \to 0$.

Ya que hemos verificado que nuestra implementación computacional produce soluciones correctas, podemos resolver el modelo directo usando modelos de velocidades con geometrías más complejas como las que se pueden encontrar en los casos reales. Para este propósito usaremos dos modelos de velocidades altamente estudiados: Canadian overthrust BP (Gray and Marfurt, 1995) y Marmousi (Martin et al., 2006); ambos modelos de velocidades representan distintos campos de exploración y se muestran en la figura 1.10.

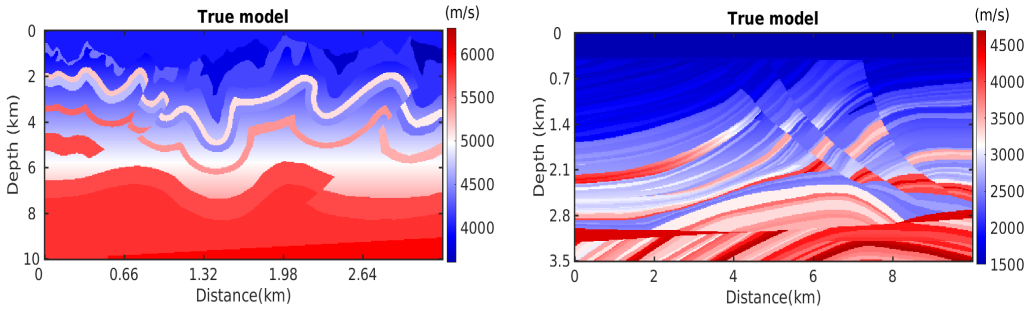

Figura 1.10: Modelos de velocidades: Canadian overthrust BP (izquierda) y Marmousi (derecha).

En la figura 1.11 se muestran algunos snapshots del campo de velocidades obtenidos de la implementación computacional de las ecuaciones (1.49-1.50), usando los modelos Canadiense y Marmousi, donde la fuente f se ubica a $10m$ de la superficie libre y corresponde a un pulso de Ricker de $40Hz$

para el modelo Canadiense y de $15Hz$ para el modelo de Marmousi; para conservar estabilidad y reducir error de dispersión, usamos $\Delta x = \Delta z = 10$ y $\Delta t = 0{,}001$. Los sismogramas que se generan con estas propagaciones de onda se muestran en la figura 1.12.

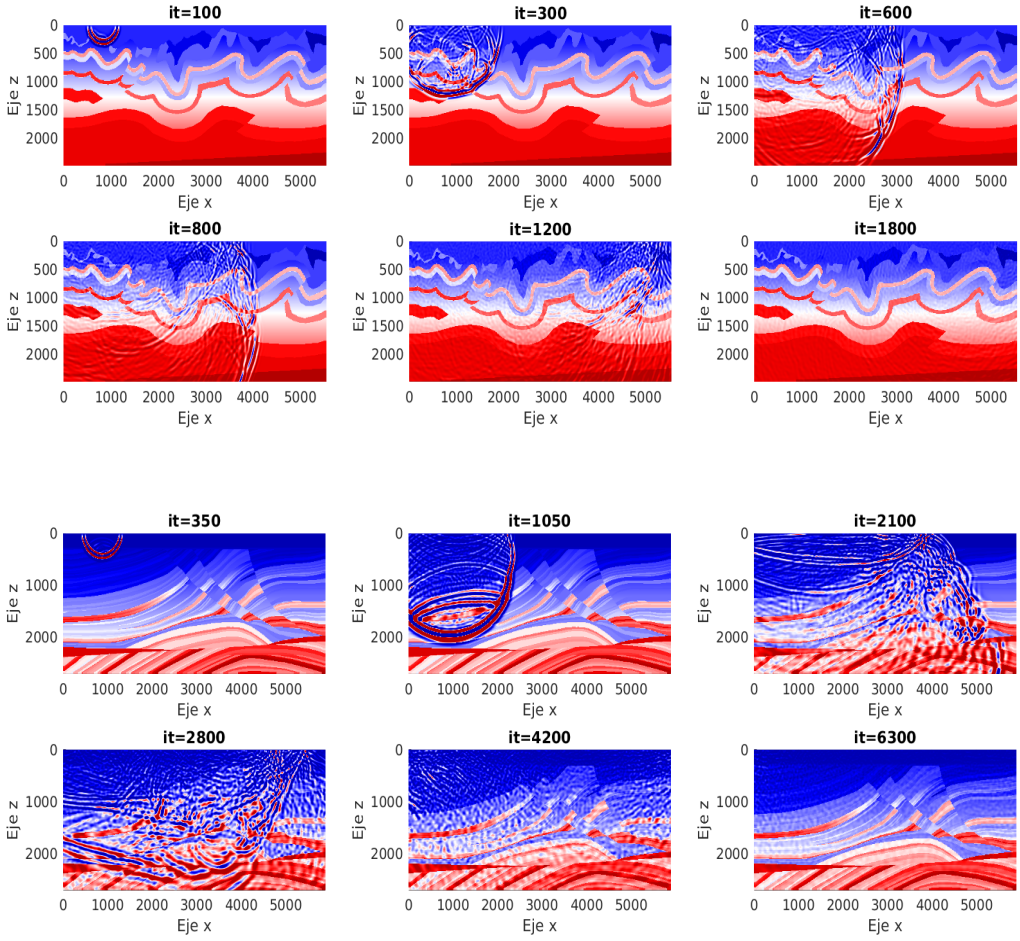

Figura 1.11: Snapshots del campo de velocidades en distintos tiempos, usando los modelos de velocidades: Canadian overthrust BP (renglones 1 y 2) y Marmousi (renglones 3 y 4).

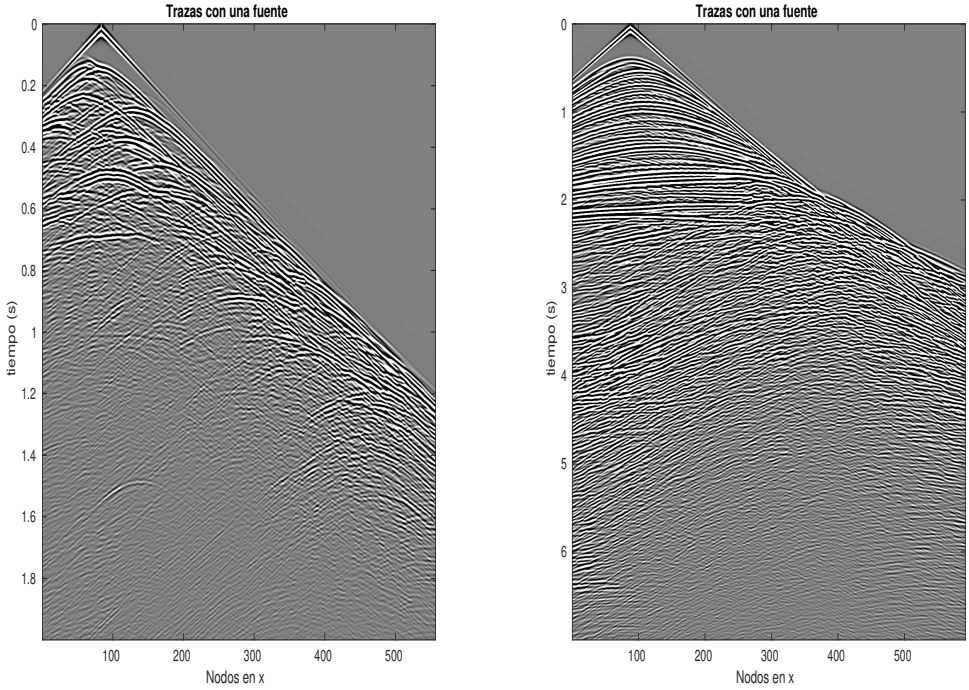

Figura 1.12: Sismogramas correspondientes a la propagación de ondas generados con una fuente dada, usando los modelos de velocidades: Canadian overthrust BP (izquierda) y Marmousi (derecha).

Capítulo 2

FWI en el dominio del tiempo

Dado un conjunto de sismogramas observados $V_{obs}(X_g, t|X_s)$, provenientes de un subsuelo cuyas propiedades físicas son desconocidas (figura 2.1), generados por una fuente en X_s y registrados por una serie de geófonos en X_g en cada tiempo t; el objetivo de la inversión de forma de onda completa o FWI (del inglés: Full-Waveform Inversion), es aproximar un modelo de parámetros físicos del subsuelo que explique los datos observados, por lo que dicho modelo también incluye información detallada sobre la estructura de las capas internas del medio en cuestión.

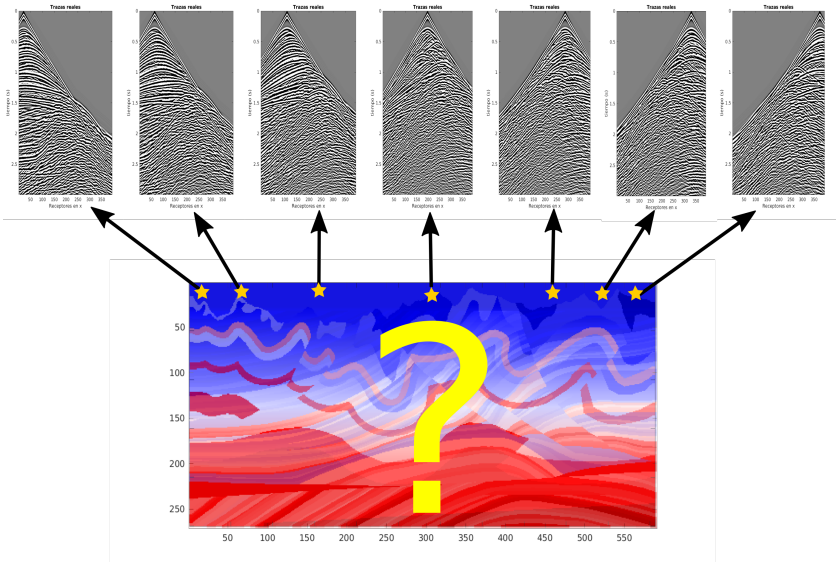

Figura 2.1: Conjunto de sismogramas observados provenientes de un subsuelo cuyas propiedades son desconocidas.

Para verificar que un modelo de parámetros físicos $\vec{m}(X)$ explica los datos $V_{obs}(X_g, t|X_s)$, debe ocurrir que,

$$L(\vec{m}) = V_{aprx}(X_g, t|X_s) \approx V_{obs}(X_g, t|X_s), \tag{2.1}$$

donde L es el operador de algún modelo matemático de propagación de ondas y $V_{aprx}(X_g, t|X_s)$ el conjunto de sismogramas aproximados (generados con $\vec{m}$). En otras palabras, nos interesa minimizar el residual $V_{res}(X_g, t|X_s)$ dado por:

$$V_{res}(X_g, t|X_s) = V_{aprx}(X_g, t|X_s) - V_{obs}(X_g, t|X_s), \tag{2.2}$$

en cada tiempo t y en cada una de las fuentes en X_s y geófonos en X_g, del experimento.

La FWI consiste en aproximar un modelo de parámetros físicos $\vec{m}(X)$ que minimice una función de costo $\epsilon(\vec{m})$, generalmente basada en la norma L_2 aplicada a la diferencia entre datos aproximados y observados (Tarantola, 1986; Virieux and Operto, 2009):

$$\epsilon = \frac{1}{2} \sum_{s=1}^{N_s} \sum_{g=1}^{N_g} \int_0^T \parallel V_{res}(X_g, t|X_s) \parallel^2 dt, \tag{2.3}$$

o en la función de costo no-cuadrática como la basada en la norma L_1 aplicada a la diferencia entre datos aproximados y observados (Tarantola, 1987; Crase et al., 1990; Brossier et al., 2010; Jeong et al., 2013):

$$\epsilon = \sum_{s=1}^{N_s} \sum_{g=1}^{N_g} \int_0^T |V_{res}(X_g, t|X_s)| dt, \tag{2.4}$$

donde N_s y N_g son el número de fuentes y geófonos, respectivamente, T es la duración de propagación de onda directa en cada fuente. Debemos notar que la función de costo ϵ, está definida en un espacio de modelos $\vec{m}(X)$, pues V_{obs} es fijo y V_{aprx} depende de $\vec{m}(X)$; para los casos multiparamétricos $\vec{m}(X) = [m_i(X)]$, donde las componentes $m_i(X)$ corresponden a cada parámetro físico (por ejemplo: velocidad, densidad, etc.).

Como podemos notar, la FWI es un problema inverso en el sentido de que se cuenta con datos reales y queremos encontrar un modelo de parámetros físicos que explique dichos datos reales. Al ser un problema inverso se traduce en un problema de minimización. El costo computacional y dificultad de la FWI radica en la realización los siguientes subprocesos en cada iteración del proceso de minimización:

1. Dado un modelo inicial $\vec{m}_i$, debemos resolver la ecuación de onda (ie. realizar una propagación directa) para generar los datos aproximados $L(\vec{m}_i) = V_{aprx}$ y calcular el residual,

$$V_{res} = V_{aprx} - V_{obs}.$$

2. Calcular el gradiente de la función de costo $\vec{G}(\vec{m}_i) = \frac{\partial \epsilon}{\partial \vec{m}_i}$, para obtener una dirección óptima en el espacio de modelos. Este proceso implica resolver el adjunto del modelo directo, el cual utiliza como fuente el residual V_{res}. Posteriormente el gradiente se calcula mediante la *zero-lag correlation*[1] de alguna ponderación entre los campos directo y adjunto, como ya veremos en la siguiente sección.

3. Calcular un *step-length* α_i, que controle el tamaño de paso en la dirección óptima. Este proceso puede ser extremadamente costoso, dependiendo del método de optimización que se vaya a aplicar. En el caso de aplicar métodos cuasi-Newton, el step-length debe satisfacer las condiciones de Wolfe (Wolfe, 1969, 1971; Nocedal and Nash, 1991) para garantizar convergencia, lo cual generalmente requiere la realización de al menos dos propagaciones extra (Ma et al., 2019).

4. Habiendo calculado el gradiente $\vec{G}(\vec{m}_i)$ y el step-length α_i, obtenemos el nuevo modelo mediante:

$$\vec{m}_{i+1} = \mathfrak{A}(\vec{m}_i, \vec{G}(\vec{m}_i), \alpha_i),$$

   para algún algoritmo de optimización $\mathfrak{A}$.

5. Actualizar el modelo anterior $\vec{m}_i \leftarrow \vec{m}_{i+1}$ y volver a repetir cada paso hasta satisfacer algún criterio de paro.

Convencionalmente los cálculos anteriores se deben realizar fuente por fuente, sin embargo, para acelerar el proceso podemos utilizar técnicas de fuentes simultáneas como veremos en la sección 2.3.

[1] La *zero-lag correlation* entre dos campos $D(X, t_i|X_s)$ y $U(X, t_i|X_s)$ nos devuelve un arreglo (de la misma dimensión de los campos) que corresponde a la suma del producto (punto a punto) de ambos campos para cada tiempo $t_i \in [0, T]$.

2.1. Cálculo de gradiente usando el método de estado adjunto

Existen varias formas de poder calcular el gradiente de la función de costo en FWI, por ejemplo, mediante teoría de perturbaciones como se muestra en Tarantola (2005) o mediante el método de estado adjunto (Plessix, 2006; Schuster, 2017). En particular nosotros usaremos este último pues nos resulta más práctico visualizar los operadores del modelo directo y adjunto.

A continuación describiremos el método de estado adjunto de forma general, en la siguiente sección lo aplicaremos a las ecuaciones (1.27) con las que generamos los datos sintéticos.

Supongamos que la linealización de las ecuaciones del modelo de propagación de ondas (modelo directo) correspondientes al modelo perturbado $\vec{m} + \delta\vec{m}$ [2] se pueden representar como:

$$A(\vec{m})W(\vec{m}) = \vec{f}, \tag{2.5}$$

donde $\vec{m} = [m_1, ..., m_n]^T$, siendo cada componente m_i algún parámetro físico del subsuelo (ejemplo: velocidad, densidad, etc.); $A(\vec{m})$ representa el operador del modelo matemático y $\vec{f}$ la fuente puntual que se aplica en algún punto del espacio. De esta forma, los campos de onda directa [3] $W(\vec{m})$ se obtienen al resolver la ecuación (2.5), es decir:

$$W(\vec{m}) = A(\vec{m})^{-1}\vec{f}. \tag{2.6}$$

En términos de esta notación y por simplicidad podemos escribr la función de costo (basada en la norma L_2) como sigue:

$$\epsilon(\vec{m}) = \frac{1}{2}\int_0^T \| \underbrace{W(\vec{m}) - d_{obs}}_{\Delta d} \|^2 \, dt, \tag{2.7}$$

$$= \frac{1}{2}\int_0^T (W(\vec{m}) - d_{obs}, W(\vec{m}) - d_{obs})dt, \tag{2.8}$$

donde $(\cdot, \cdot)$ representa el producto escalar (o producto interno) y d_{obs} es el conjunto de datos observados.

[2] Si $\delta\vec{m}$ es una perturbación del modelo $\vec{m}$, el modelo perturbado es: $\vec{m} + \delta\vec{m}$.

[3] El campo $W(\vec{m})$ correspondiente a la linealización del modelo perturbado (2.5) también se conoce como aproximación de Born.

La i-ésima componente del gradiente de $\epsilon(\vec{m})$ con respecto a algún parámetro físico $m_i(X)$ está dada por:

$$\frac{\partial \epsilon(\vec{m})}{\partial m_i} = \int_0^T \left(\frac{\partial W(\vec{m})}{\partial m_i}, \underbrace{W(\vec{m}) - d_{obs}}_{\Delta d} \right) dt, \tag{2.9}$$

donde $\frac{\partial W(\vec{m})}{\partial m_i}$ es la derivada de Fréchet, la cual determina el cambio en los datos $W(\vec{m})$ con respecto a perturbaciones en el parámetros físico m_i. Para calcular la derivada de Fréchet aplicaremos el método de estado adjunto, (Plessix, 2006).

De acuerdo a Dutta and Schuster (2014), podemos calcular la derivada de Fréchet derivando la ecuación (2.5) con respecto a m_i para obtener:

$$A(\vec{m}) \frac{\partial W(\vec{m})}{\partial m_i} + \frac{\partial A(\vec{m})}{\partial m_i} W(\vec{m}) = 0, \tag{2.10}$$

de donde se obtiene la derivada de Fréchet:

$$\frac{\partial W(\vec{m})}{\partial m_i} = -A(\vec{m})^{-1} \frac{\partial A(\vec{m})}{\partial m_i} W(\vec{m}). \tag{2.11}$$

Sustituyendo la ecuación (2.11) en la ecuación (2.9), obtenemos:

$$\begin{aligned} \frac{\partial \epsilon(\vec{m})}{\partial m_i} &= -\int_0^T \left(A(\vec{m})^{-1} \frac{\partial A(\vec{m})}{\partial m_i} W(\vec{m}), \underbrace{W(\vec{m}) - d_{obs}}_{\Delta d} \right) dt, \\ &= -\int_0^T \left(\frac{\partial A(\vec{m})}{\partial m_i} W(\vec{m}), \underbrace{\{A(\vec{m})^{-1}\}^\dagger \Delta d}_{W(\vec{m})^\dagger} \right) dt, \end{aligned} \tag{2.12}$$

donde $\{A(\vec{m})^{-1}\}^\dagger$ representa el adjunto del operador $A(\vec{m})^{-1}$ (véase el Apéndice B).

El término $\frac{\partial A(\vec{m})}{\partial m_i} W(\vec{m})$ de la ecuación (2.12), es una ponderación de los campos de onda directa y el término $W(\vec{m})^\dagger = \{A(\vec{m})^{-1}\}^\dagger \Delta d$ se conoce como campo adjunto (o estado adjunto) pues corresponde a la solución de las ecuaciones del modelo adjunto:

$$A(\vec{m})^\dagger W(\vec{m})^\dagger = \Delta d, \tag{2.13}$$

donde la fuente está determinada por el residual de datos Δd.

Gradiente de la función de costo basada en la norma L_1

Sea

$$\Delta d_{L_1} = |\underbrace{W(\vec{m}) - d_{obs}}_{\Delta d}|,$$

la diferencia (basada en la norma L_1) entre los datos aproximados $W(\vec{m})$ y los datos observados d_{obs}, entonces [4]:

$$\frac{\partial \Delta d_{L_1}}{\partial m_i} = \left(\frac{\Delta d}{|\Delta d|}, \frac{\partial W(\vec{m})}{\partial m_i} \right),$$

donde $(\cdot, \cdot)$ denota el producto interno, así que la ecuación (2.9) en términos de la norma L_1 queda:

$$\frac{\partial \epsilon(\vec{m})}{\partial m_i} = \int_0^T \left(\frac{\partial W(\vec{m})}{\partial m_i}, \frac{\Delta d}{|\Delta d|} \right) dt. \tag{2.14}$$

Aplicando el método de estado adjunto a la ecuación (2.14), obtenemos:

$$\frac{\partial \epsilon(\vec{m})}{\partial m_i} = -\int_0^T \left(\frac{\partial A(\vec{m})}{\partial m_i} W(\vec{m}), \underbrace{\{A(\vec{m})^{-1}\}^\dagger \frac{\Delta d}{|\Delta d|}}_{W(\vec{m})^\dagger} \right) dt. \tag{2.15}$$

En este caso podemos ver que el modelo adjunto es:

$$A(\vec{m})^\dagger W(\vec{m})^\dagger = \frac{\Delta d}{|\Delta d|}. \tag{2.16}$$

[4]Recordemos que si $f(x) = |g(x)|$, donde f y g son funciones diferenciables, entonces $f'(x)$ está definida en los puntos donde $g(x) \neq 0$ y $f'(x) = \frac{g(x)}{|g(x)|} g'(x)$.

2.1.1. Gradiente para el caso acústico

La linealización de las ecuaciones (1.27) con respecto al modelo perturbado $\nu + \delta\nu$, se pueden escribir en forma matricial como[5]:

$$\underbrace{\begin{bmatrix} \frac{\partial}{\partial t} & -\nu^2\frac{\partial}{\partial x} & -\nu^2\frac{\partial}{\partial z} \\ -\frac{\partial}{\partial x} & \frac{\partial}{\partial t} & 0 \\ -\frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial t} \end{bmatrix}}_{A(\nu)} \underbrace{\begin{bmatrix} \delta v \\ \delta\sigma_{xx} \\ \delta\sigma_{zz} \end{bmatrix}}_{W(\nu)} = \underbrace{\begin{bmatrix} 2\nu\delta\nu\nabla\cdot\vec{\sigma} \\ 0 \\ 0 \end{bmatrix}}_{\vec{f}}. \tag{2.17}$$

En este caso $W(\nu) = [\delta v, \delta\sigma_{xx}, \delta\sigma_{zz}]^T$ corresponde a campos de onda aproximados con las ecuaciones (1.27). Notemos que en este caso $\vec{m} = [\nu(X)]$, siendo $\nu(X)$ un modelo de velocidades del subsuelo con $X = (x, z)$.

Si los datos observados (o recolectados) son d_{obs}, entonces el residual de datos es: $\Delta d = W(\nu) - d_{obs}$.[6]

De acuerdo a la ecuación (2.12), el gradiente de la función de costo con respecto al modelo de velocidades $\nu(X)$ está dado por,

$$\frac{\partial\epsilon(\nu)}{\partial\nu} = -\int_0^T \left(\frac{\partial A(\nu)}{\partial\nu} W(\nu), \underbrace{\{A(\nu)^{-1}\}^\dagger \Delta d}_{W(\nu)^\dagger} \right) dt, \tag{2.18}$$

donde,

$$\begin{aligned} \frac{\partial A(\nu)}{\partial\nu} W(\nu) &= \begin{bmatrix} 0 & -2\nu\frac{\partial}{\partial x} & -2\nu\frac{\partial}{\partial z} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v \\ \sigma_{xx} \\ \sigma_{zz} \end{bmatrix}, \\ &= \begin{bmatrix} -2\nu\left(\frac{\partial\sigma_{xx}}{\partial x} + \frac{\partial\sigma_{zz}}{\partial z}\right) \\ 0 \\ 0 \end{bmatrix}. \end{aligned} \tag{2.19}$$

Por otro lado $W(\nu)^\dagger = [\check{v}, \check{\sigma}_{xx}, \check{\sigma}_{zz}]^T$ es el estado adjunto que corresponde a la solución de las ecuaciones del modelo adjunto,

$$A(\nu)^\dagger W(\nu)^\dagger = \Delta d. \tag{2.20}$$

[5]La perturbación de la ecuación $\frac{\partial v}{\partial t} - \nu^2\nabla\cdot\vec{\sigma} = f$, con respecto a $\delta\nu$, está dada por: $\frac{\partial\delta v}{\partial t} - (\nu^2\nabla\cdot\vec{\delta\sigma} + 2\nu\delta\nu\nabla\cdot\vec{\sigma}) = 0$, es decir, $\frac{\partial\delta v}{\partial t} - \nu^2\nabla\cdot\vec{\delta\sigma} = 2\nu\delta\nu\nabla\cdot\vec{\sigma}$.

[6]Si d_{obs} está dado por sismogramas de velocidades observadas, $W(\nu)$ será el correspondiente sismograma de velocidades aproximadas.

Considerando que el adjunto del operador $\frac{\partial}{\partial x}$ es $-\frac{\partial}{\partial x}$ (véase el Apéndice B) y $A(\nu)^\dagger$ es la transpuesta de los elementos adjuntos de $A(\nu)$, podemos escribir explícitamente las ecuaciones del modelo adjunto como:

$$\underbrace{\begin{bmatrix} -\frac{\partial}{\partial t} & \frac{\partial}{\partial x} & \frac{\partial}{\partial z} \\ \nu^2\frac{\partial}{\partial x} & -\frac{\partial}{\partial t} & 0 \\ \nu^2\frac{\partial}{\partial z} & 0 & -\frac{\partial}{\partial t} \end{bmatrix}}_{A(\nu)^\dagger} \underbrace{\begin{bmatrix} \check{v} \\ \check{\sigma}_{xx} \\ \check{\sigma}_{zz} \end{bmatrix}}_{W(\nu)^\dagger} = \underbrace{\begin{bmatrix} \Delta v \\ 0 \\ 0 \end{bmatrix}}_{\Delta d}, \tag{2.21}$$

donde $\Delta v = v - v_{obs}$ es el residual de velocidades.

Notemos que las ecuaciones del modelo adjunto, ecuaciones (2.21), implican resolver un nuevo problema, sin embargo, debido a que el modelo directo se basa en la ecuación de onda cuyo operador lineal es de segundo orden y éste es autoadjunto (véase el Apéndice B), entonces resolver las ecuaciones adjuntas equivale a resolver el problema directo en tiempo reverso usando como fuente el residual de datos $\Delta d = [\Delta v, 0, 0]^T$. Por lo tanto el campo adjunto $W(\nu)^\dagger = [\check{v}, \check{\sigma}_{xx}, \check{\sigma}_{zz}]^T$ se obtiene al resolver:

$$\begin{aligned} \frac{\partial \check{v}}{\partial t} &= \nu^2\left(\frac{\partial \check{\sigma}_{xx}}{\partial x} + \frac{\partial \check{\sigma}_{zz}}{\partial z}\right) + \sum_{g=1}^{N_g} \underbrace{\Delta v(X_g, t)}_{\Psi}\, \delta(X - X_g), \\ \frac{\partial \check{\sigma}_{xx}}{\partial t} &= \frac{\partial \check{v}}{\partial x}, \\ \frac{\partial \check{\sigma}_{zz}}{\partial t} &= \frac{\partial \check{v}}{\partial z}, \\ \check{v}(X, t = T) &= \check{\vec{\sigma}}(X, t = T) = 0, \end{aligned} \tag{2.22}$$

donde $\Delta v(X_g, t)$ se inyecta como fuente en sus respectivas coordenadas de los receptores $X_g = (x_g, z_g)$, partiendo del tiempo $t = T$ al tiempo $t = 0$, véase Schuster (2017).

Por lo tanto, sustituyendo la ecuación (2.19) y $W(\nu)^\dagger = [\check{v}, \check{\sigma}_{xx}, \check{\sigma}_{zz}]^T$ en la ecuación (2.18), obtenemos el gradiente para el caso acústico:

$$\begin{aligned} \frac{\partial \epsilon(\nu)}{\partial \nu} &= -\int_0^T \left(\begin{bmatrix} -2\nu\left(\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{zz}}{\partial z}\right) \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \check{v} \\ \check{\sigma}_{xx} \\ \check{\sigma}_{zz} \end{bmatrix} \right) dt \\ &= 2\nu \int_0^T (\nabla \cdot \vec{\sigma})\, \check{v} dt \end{aligned} \tag{2.23}$$

Para eliminar las derivadas espaciales en $\nabla \cdot \vec{\sigma}$ usamos el hecho de que:

$$\nabla \cdot \vec{\sigma} = \frac{1}{\nu^2} \frac{\partial v}{\partial t}, \tag{2.24}$$

entonces el gradiente de la función de costo (basada en la norma L_2) con respecto a las velocidades $\nu(X)$ está dado por:

$$\frac{\partial \epsilon(\nu)}{\partial \nu} = \frac{2}{\nu(X)} \sum_{s=1}^{N_s} \int_0^T \underbrace{\partial v(X,t)/\partial t}_{D(X,t|X_s)} \underbrace{\check{v}(X,t)}_{U(X,t|X_s)} \, dt, \tag{2.25}$$

donde $v(X,t)$ satisface el modelo directo dado por las ecuaciones (1.27) y $\check{v}(X,t)$ satisface el modelo adjunto (o backward model) dado por las ecuaciones (2.22).

La ecuación (2.25) también sirve para obtener el gradiente de la función de costo basada en la norma L_1 a diferencia de que $\check{v}$ debe satisfacer las ecuaciones (2.22) con $\Psi = \frac{\Delta v}{|\Delta v|}$, como lo indica la ecuación (2.16).

Nota 2.1.1. *Para dar una interpretación física a la ecuación (2.25) se suele definir el cambio de variables $\tilde{U}(X,t|X_s) = U(X,T-t|X_s)$, es decir:*

$$\tilde{\check{v}}(X,t) = \check{v}(X,T-t), \quad \tilde{\check{\sigma}}_{xx}(X,t) = \check{\sigma}_{xx}(X,T-t), \quad \tilde{\check{\sigma}}_{zz}(X,t) = \check{\sigma}_{zz}(X,T-t),$$

por lo que las ecuaciones (2.22) se pueden escribir como:

$$\begin{aligned}
\frac{\partial \tilde{\check{v}}}{\partial t} &= \nu^2 \left(\frac{\partial \tilde{\check{\sigma}}_{xx}}{\partial x} + \frac{\partial \tilde{\check{\sigma}}_{zz}}{\partial z} \right) + \sum_{g=1}^{N_g} \underbrace{\Delta v(X_g, T-t)}_{\Psi} \delta(X - X_g), \\
\frac{\partial \tilde{\check{\sigma}}_{xx}}{\partial t} &= \frac{\partial \tilde{\check{v}}}{\partial x}, \\
\frac{\partial \tilde{\check{\sigma}}_{zz}}{\partial t} &= \frac{\partial \tilde{\check{v}}}{\partial z}, \\
\tilde{\check{v}}(X, t=0) &= \tilde{\check{\vec{\sigma}}}(X, t=0) = 0,
\end{aligned} \tag{2.26}$$

de esta forma la ecuación (2.25) se puede escribir como[7]*:*

$$\frac{\partial \epsilon(\nu)}{\partial \nu} = \frac{2}{\nu(X)} \sum_{s=1}^{N_s} \int_0^T \underbrace{\partial v(X,t)/\partial t}_{D(X,t|X_s)} \underbrace{\tilde{\check{v}}(X,T-t)}_{\tilde{U}(X,T-t|X_s)} \, dt, \tag{2.27}$$

[7]En la ecuación (2.27) se puede ver claramente que cuando se calcula el gradiente en tiempo reverso (de $t = T$ a $t = 0$), el campo D viaja de $D(X,T|X_s)$ a $D(X,0|X_s)$ mientras que el campo adjunto $\tilde{U}$ viaja de $\tilde{U}(X,0|X_s)$ a $\tilde{U}(X,T|X_s)$.

donde $v(X,t)$ satisface el modelo directo dado por las ecuaciones (1.27) y $\tilde{\check{v}}(X,t)$ satisface el modelo adjunto dado por las ecuaciones (2.26).

En la Figura 2.3 se muestran algunos snapshots de los campos de velocidades, directo y adjunto (v y $\check{v}$, respectivamente) junto con la acumulación en cada instante del tiempo, de la zero-lag correlation indicada por la ecuación (2.25) usando una fuente y un solo receptor, por lo que el residual de datos es una simple traza: la diferencia entre la traza observada y la traza aproximada en dicho receptor. En este caso se utilizó un modelo de velocidades homogéneo y el modelo Canadiense para construir las trazas aproximada y observada, respectivamente. El resultado de esta operación corresponde al gradiente de velocidades (eq.(2.25)) que se muestra en la Figura 2.2.

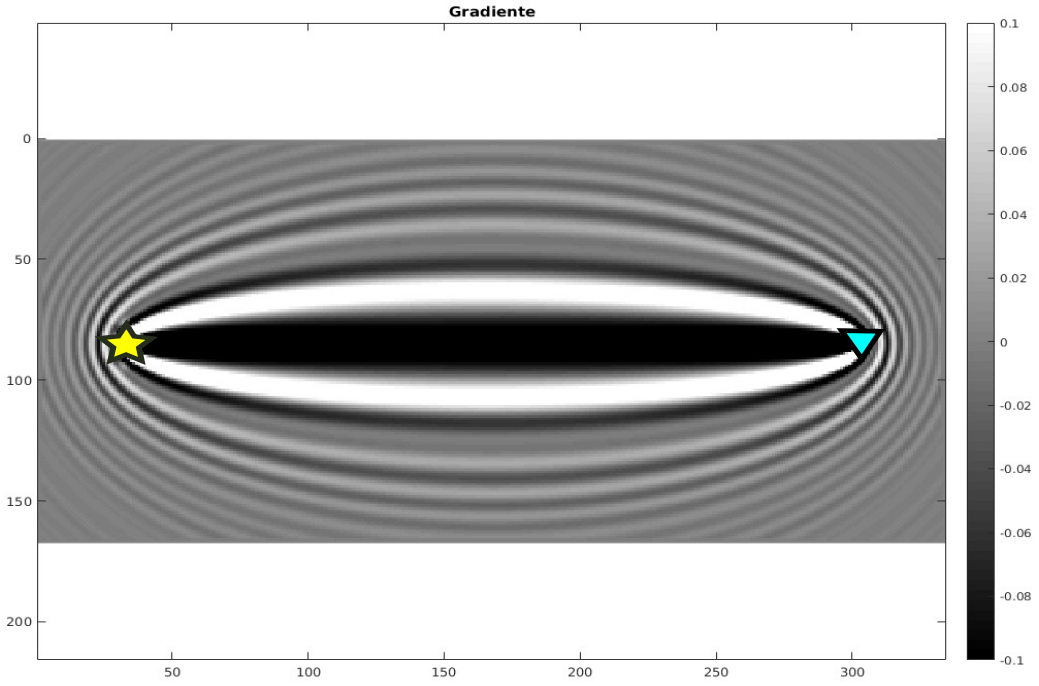

Figura 2.2: Gradiente de velocidades correspondiente a una fuente (estrella amarilla) y un solo receptor (triángulo verde).

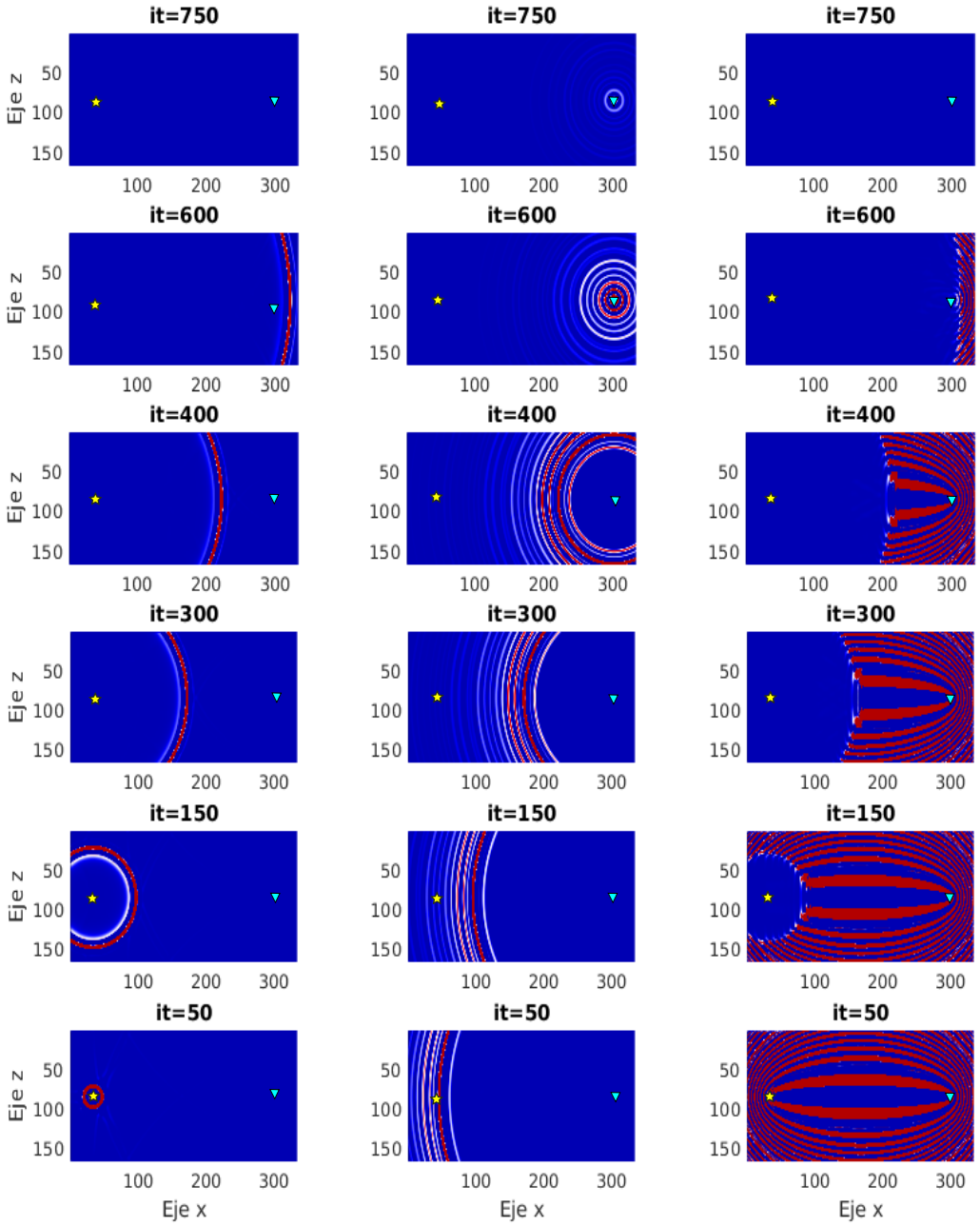

Figura 2.3: Propagación en tiempo reverso de los campos de velocidades directo (columna 1) y adjunto (columna 2), junto con la acumulación de valores del gradiente (columna 3); usando una fuente (estrella amarilla) y un solo receptor (triángulo verde).

2.2. Multiscaling en FWI

Una técnica popular en FWI que permite enfrentar la no linealidad de la función de costo, es el proceso de multiscaling (Bunks et al., 1995; Boonyasiriwat et al., 2009; dos Santos and Pestana, 2015). Dicho proceso consiste en iniciar la FWI usando datos aproximados con bajas frecuencias, lo cual representa una suavización de la función de costo, reduciendo la cantidad de mínimos locales y facilitando la búsqueda del mínimo global. Cuando se realiza cierto número de iteraciones de FWI o se alcanza convergencia en la frecuencia actual, se incrementa la frecuencia de los datos aproximados y se repite el proceso hasta alcanzar la máxima frecuencia soportada por el esquema numérico que se esté usando (véase la figura 2.4). Por lo tanto, para alcanzar el mínimo global, un modelo inicial puede estar mucho más alejado del modelo real para bajas frecuencias, mientras que se requiere un modelo inicial mucho más cercano para datos con altas frecuencias (Bunks et al., 1995). Debido a que los datos aproximados deben ser comparables con los datos observados, estos últimos deberán ser filtrados al mismo rango de frecuencias de los datos aproximados. Iniciar la FWI con bajas frecuencias permite recobrar las estructuras de gran escala en el subsuelo y conforme se van incrementando las frecuencias, se van recobrando las estructuras más finas.

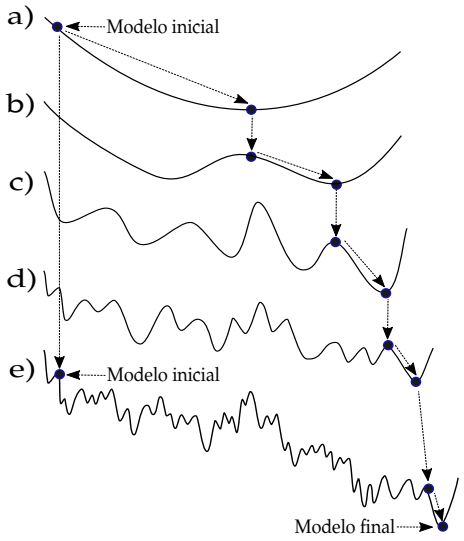

Figura 2.4: Evolución de la función de costo durante el proceso de multiscaling. Las frecuencias de los datos aumentan de a) a e) hasta alcanzar una frecuencia máxima.

2.3. Fuentes simultáneas dinámicas

Como se muestra en la ecuación (2.25), cada componente del gradiente $G_i = \frac{\partial \epsilon(\vec{m})}{\partial m_i}$, se obtiene mediante la *zero-lag correlation* entre el campo de onda directo (o propagado hacia adelante) $D(X,t|X_s)$ y el campo de onda adjunto (o propagado hacia atrás) $U(X,t|X_s)$, véase Tarantola (1986, 1987); Boonyasiriwat et al. (2009), mediante:

$$G_i = \frac{\partial \epsilon(\vec{m}(X))}{\partial m_i(X)} = \sum_{s=1}^{N_s} \int_0^T D(X,t|X_s)U(X,t|X_s)dt, \tag{2.28}$$

donde la forma de los campos D y U depende de la función de costo ϵ y de las ecuaciones del modelo directo como se vio en la sección 2.1.1.

Si suponemos que se debe realizar la operación indicada en la ecuación (2.28) para una sola fuente, se requiere casi el triple del tiempo de cómputo del que se necesita para resolver el modelo directo correspondiente a dicha fuente, pues como vimos en la sección 2.1.1, el campo $U(X,t|X_s)$ se genera con el residual (fuente adjunta) $V_{res} = V_{aprx} - V_{obs}$, donde V_{aprx} se obtiene al realizar una propagación directa con un modelo de velocidades dado. Considerando que la FWI convencional tiene que realizar este proceso para cada una de las N_s fuentes en el experimento, podemos notar que calcular el gradiente representa uno de los procesos más costosos computacionalmente.

Para reducir el costo computacional que requiere la construcción del gradiente, inicialmente se aplicaron técnicas de codificación de fuentes aplicadas en *prestack migration* (Romero et al., 2000; Jing et al., 2000) y posteriormente se aplicaron en FWI (Krebs et al., 2009). Algunas variantes de estas técnicas para FWI se basan en inversión de fase, desplazamiento de fase y desplazamiento de tiempos, véase Herrmann et al. (2009); Ben-Hadj-Ali et al. (2011). Estas técnicas aprovechan la linealidad del modelo directo con respecto a las fuentes, para activar varias fuentes simultáneamente en una sola propagación directa, generando lo que se conoce como *super-shot*.

Debemos ser cuidadosos al utilizar técnicas de codificación de fuentes, pues estas técnicas tienen el defecto de introducir una cantidad de ruido conocido como *crosstalk noise*, el cual si no se reduce puede degradar la precisión de los parámetros físicos obtenidos de la FWI, véase Romero et al. (2000). Para reducir el crosstalk noise, en este trabajo usamos una técnica de fuentes simultáneas dinámicas que consiste en la combinación tres estrategias populares de codificación de fuentes: *random-in-subgroup shot sub-sampling*

(Díaz and Guitton, 2011; Ha and Shin, 2013; Shi and He, 2018); *random time shifting* (Zhan et al., 2013; Schuster et al., 2011) y *random polarities* (Boonyasiriwat and Schuster, 2010); con la propiedad de que las configuraciones de esas técnicas van cambiando en cada iteración de FWI, de aquí que hemos acuñado el término "fuentes simultáneas dinámicas"para describir este método (Krebs et al., 2009). Hemos implementado la combinación de estas estrategias en el siguiente orden:

1. Consideremos $X_{ss} = \{X_s : s = 1, 2, ..., N_s\}$ el conjunto de fuentes a usar en el experimento. En cada iteración de FWI, seleccionaremos aleatoriamente un subconjunto de fuentes a activar, $\gamma(X_{ss}) \subset X_{ss}$ (circulos negros de la figura 2.5) mediante la técnica de *random-in-subgroup shot sub-sampling* (Díaz and Guitton, 2011), por lo que el número de fuentes inactivas (círculos blancos de la figura 2.5) entre dos fuentes activas dependerá de la frecuencia actual del proceso de multiscaling. El número de fuentes en el super-shot $\gamma(X_{ss})$, está dado por $N_{ss} = round(\frac{f}{\lambda})$, donde f es la frecuencia actual del proceso de multiscaling y $\lambda = \frac{f_{max}}{n_s}$, es una proporción entre el número de fuentes activadas n_s cuando el proceso de multiscaling alcance la frecuencia máxima f_{max}, véase Shi and He (2018). Por lo tanto, N_{ss} incrementa conforme incrementan las frecuencias en el proceso de multiscaling.

2. De forma aleatoria se asigna a cada fuente activa $X_s \in \gamma(X_{ss})$, una polaridad $(-1)^{p(s)}$ y un desplazamiento temporal τ_s (Schuster et al., 2011); de tal forma que a cada fuente activa X_s le corresponda una onduleta de la forma: $(-1)^{p(s)}w(t - \tau_s)$, véase la figura 2.6. Ambas cantidades son aleatorias, $p(s)$ es un número entero positivo y τ_s viene de la N_{ss}-partición equiespaciada del intervalo $[0, \frac{T}{5}]$, donde T es el tiempo que dura la propagación directa.

Considerando esta codificación de fuentes, si

$$\gamma(X_{ss}) = \{X_s : s = 1, 2, ..., N_{ss} < N_s\},$$

es el conjunto de fuentes seleccionadas para un super-shot actual, el campo propagado hacia adelante y el campo residual propagado hacia atrás (campo adjunto), adquieren la forma:

$$\tilde{D}(X, t|\gamma(X_{ss})) = \sum_{s=1}^{N_{ss}} (-1)^{p(s)} D(X, t - \tau_s|X_s) \tag{2.29}$$

y

$$\tilde{U}(X, t|\gamma(X_{ss})) = \sum_{k=1}^{N_{ss}} (-1)^{p(k)} U(X, t - \tau_k|X_k). \tag{2.30}$$

Así cada componente del gradiente con esta técnica de fuentes simultáneas dinámicas, está dada por:

$$\begin{aligned}
\tilde{G} &= \int_0^T \tilde{D}(X,t|\gamma(X_{ss}))\tilde{U}(X,t|\gamma(X_{ss}))dt \\
&= \underbrace{\int_0^T \sum_{s=k}^{N_{ss}} D(X,t-\tau_s|X_s)U(X,t-\tau_k|X_k)dt}_{G} + \\
&\quad \underbrace{\int_0^T \sum_{s\neq k}^{N_{ss}} (-1)^{p(s)} D(X,t-\tau_s|X_s)(-1)^{p(k)} U(X,t-\tau_k|X_k)dt}_{\delta G} \\
&= G + \delta G, \qquad (2.31)
\end{aligned}$$

donde G corresponde a la componente de gradiente obtenida de forma convencional y δG es el ruido *crosstalk noise* que genera la correlación entre campos D y U con polaridades y desplazamientos temporales distintos.

Justificación de la reducción del crosstalk noise

Notemos que los índices de las fuentes en la integral del gradiente G de la ecuación (2.31), son iguales ($s = k$), así que la polaridad de los campos D y U que determinan el gradiente G, siempre es positiva y además ambos campos preservan el mismo desfazamiento ($\tau_s = \tau_k$), lo que asegura un apilamiento coherente de los campos al momento de hacer la zero-lag correlation; esto permite reconstruir las componentes del gradiente como en la forma convencional. Sin embargo, en el término de crosstalk noise δG en la ecuación (2.31), las polaridades y desfazameintos de los campos D y U son direfentes, así que la zero-lag correlation se calcula con campos apilados incoherentemente, reduciendo el efecto del corsstalk noise. Por otro lado, la técnica dinámica de "random-in-subgroup shot subsampling"permite que todas las fuentes adyacentes en $\gamma(X_{ss})$ estén separadas equidistantemente, iniciando con un número pequeño de fuentes e incrementando el número de fuentes conforme incrementan las frecuencias con el multiscaling, esto reduce el crosstalk noise para los casos en que dos fuentes adyacentes tienen desplazamientos temporales cercanos.

En la figura 2.7 se muestran algunos snapshots del campo de velocidades generados con un super-shot usando nuestra técnica de fuentes simultáneas dinámicas (cada iteración corresponde a un tiempo de paso en la propagación

directa); en la figura 2.8 se muestra el sismograma registrado sobre una línea de receptores a 10m de la superficie libre.

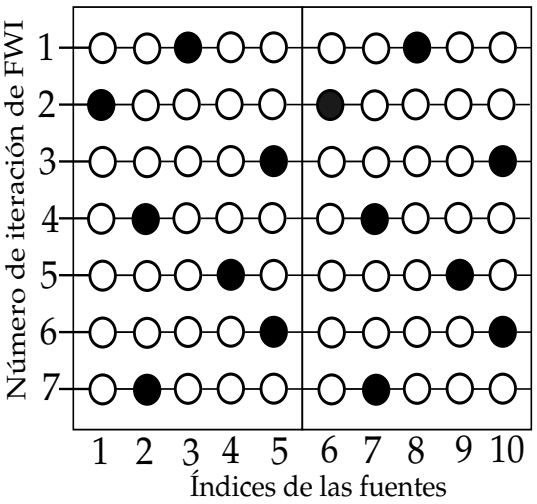

Figura 2.5: Selección de fuentes a activar (circulos negros) en cada super-shot mediante la técnica de *random-in-subgroup shot sub-sampling*, para una frecuencia fija del multiscaling, cuya configuración cambia en cada iteración de FWI.

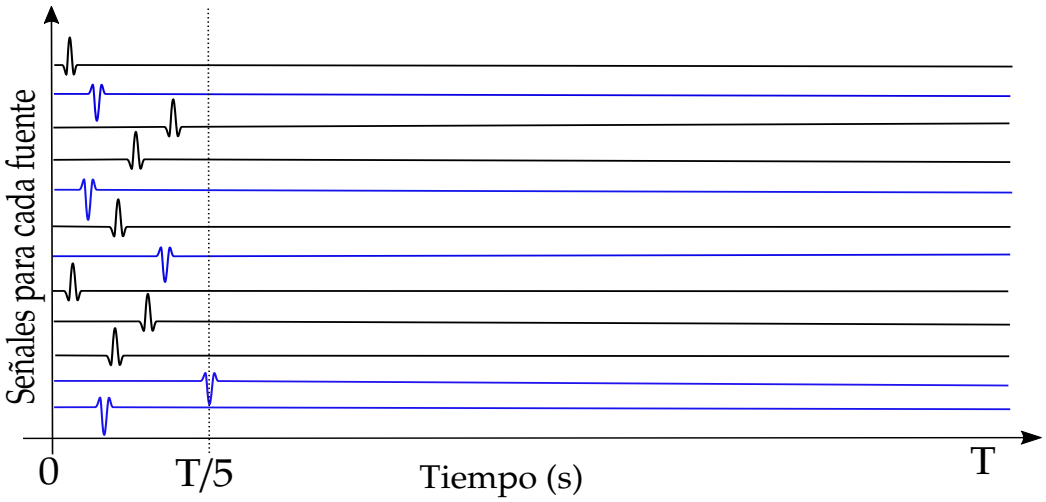

Figura 2.6: Onduletas con polaridades y desplazamientos temporales aleatorios asignadas a las fuentes de un super-shot, cuya configuración cambia en cada iteración de FWI.

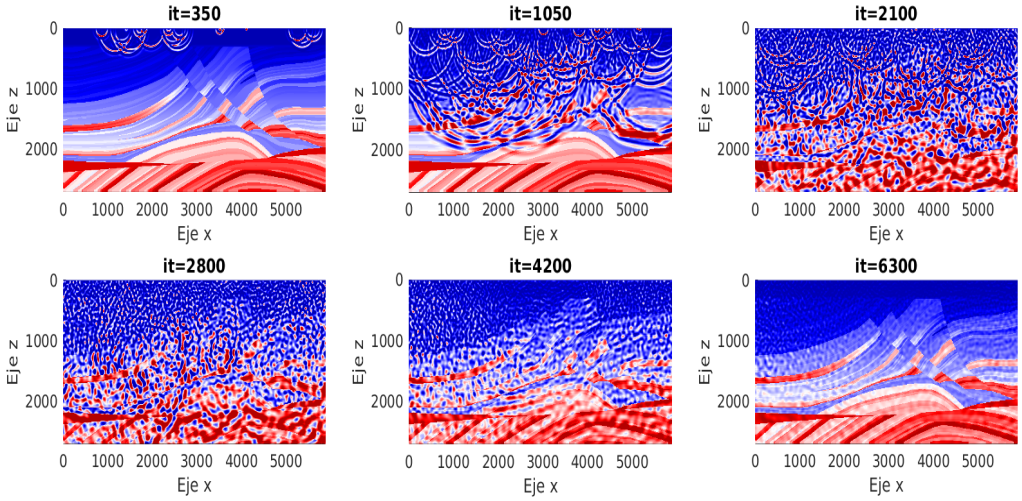

Figura 2.7: Snapshots del campo de velocidades en tiempos diferentes, usando nuestra estrategia de fuentes simultáneas dinámicas.

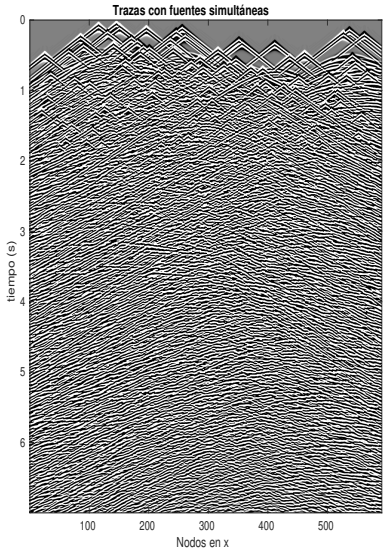

Figura 2.8: Sismograma de velocidades registrado sobre una línea de receptores a $10m$ de la superficie libre, usando nuestra estrategia de fuentes simultáneas dinámicas.

2.4. Métodos de optimización clásicos en FWI

Para minimizar la función de costo en la ecuación (2.3) o ecuación (2.4) y aproximar un modelo de parámetros físicos $\vec{m}(X)$, que expique los datos observados, la forma más simple es aplicar un método de descenso de gradiente, que consiste en empezar con un modelo inicial $\vec{m}_0(X)$ y realizar un proceso de búsqueda iterativo como sigue:

$$\vec{m}_{k+1} = \vec{m}_k - \alpha_k \vec{G}(\vec{m}_k), \tag{2.32}$$

donde α_k es el k-ésimo step-length y $\vec{G}(\vec{m}_k)$ es el k-ésimo gradiente de la función de costo. La forma específica de la ecuación (2.32) dependerá del método de optimización que se aplique.

Aunque el método de descenso de gradiente resulta simple de implementar, es un método de convergencia lenta, en este caso es mejor utilizar el método de gradiente conjugado, que resulta como sigue:

$$\vec{m}_{k+1} = \vec{m}_k - \alpha_k \vec{d}(\vec{m}_k), \tag{2.33}$$

donde $\vec{d}(\vec{m}_k)$ es la k-ésima dirección conjugada, dada por:

$$\vec{d}(\vec{m}_k) = \vec{G}(\vec{m}_k) + \beta_k \vec{d}(\vec{m}_{k-1}), \tag{2.34}$$

donde,

$$\beta_k = \frac{\| \vec{G}(\vec{m}_k) \|^2}{\| \vec{G}(\vec{m}_{k-1}) \|^2}. \tag{2.35}$$

En teoría el método de Newton (ver Apéndice A) es uno de los métodos de optimización más rápidos, pues éste usa la información de la inversa del Hessiano para precondicionar el gradiente y acelerar la convergencia. Sin embargo en problemas de gran escala, como lo es la FWI, se considera prohibido (Schuster, 2017; Virieux and Operto, 2009) ya que el cálculo de la inversa del Hessiano resulta extremadamente costoso computacionalmente, en este caso resulta mejor utilizar métodos Quasi-Newton, los cuales utilizan una aproximación de la inversa del Hessiano H_k^{-1} para precondicionar el gradiente y obtener:

$$\vec{m}_{k+1} = \vec{m}_k - \alpha_k \underbrace{H_k^{-1} \vec{G}(\vec{m}_k)}_{d_k}. \tag{2.36}$$

Uno de los métodos cuasi-Newton mayormente usado en FWI es el método L-BFGS (Nocedal and Writght, 1999), el cual revisaremos en la sección 5.0.1. En Ma et al. (2019) se pueden encontrar diferentes métodos para aproximar el step-length α_k en FWI.

Capítulo 3

Métodos de Optimización de Gradiente Adaptable

Es bien sabido que los procesos de entrenamiento en redes neuronales artificiales se traducen en un problema de minimización, por lo que en los últimos 10 años se han ido desarrollando paquetes científicos de machine learning como: TensorFlow, Keras, Caffe, etc., (Erickson et al., 2017) los cuales contienen poderosos métodos de optimización que generalmente se usan como *optimizadores de caja negra* y probablemente sólo han sido conocidos por la comunidad del *Deep Learning*, ya que es difícil encontrar explicaciones prácticas de sus fortalezas y debilidades (Ruder, 2016). Estos métodos a los que nos referimos, son mejor conocidos como Métodos de Optimización de Gradiente Adaptable o métodos AGO (por sus siglas en Inglés), los cuales surgen a partir de los Métodos de Optimización de Momentum, diseñados para acelerar los métodos de optimización estocástica (Ruder, 2016).

A diferencia de los métodos de gradiente simple cuya analogía corresponde al de un caminante que da un paso (de tamaño: step-length) hacia una dirección local que indica el gradiente, los métodos AGO consideran todos los gradientes de las iteraciones anteriores para ponderarlos (generalmente usando el promedio móvil exponencial) y construir una dirección más exacta, con un step-length capaz de irse adaptando bajo distintas condiciones conforme incrementa el número de iteraciones, dando lugar a diferentes métodos AGO. La analogía de algunos de estos métodos corresponde a una pelota inteligente que acumula impulso a medida que rueda cuesta abajo, volviéndose cada vez más rápida en el camino teniendo una idea hacia dónde ir para poder reducir la velocidad antes de que el camino vuelva a subir (Ruder, 2016), véase la figura 3.1.

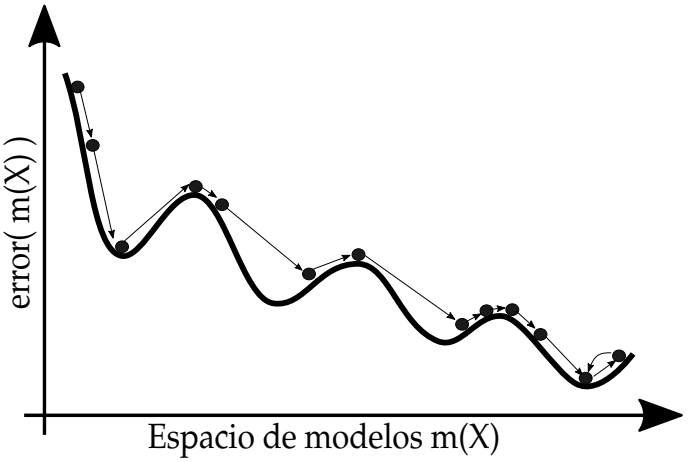

Figura 3.1: Pelota que rueda cuesta abajo y acumula impulso conforme va avanzando.

Los métodos AGO también son conocidos como *Adaptive Learning Rate (ALR) methods*, ya que en la literatura del Machine Learning, el parámetro step-length también se conoce como *learning-rate*. La justificación teórica sobre la estructura de los métodos AGO está diseñada para garantizar convergencia al mínimo global en funciones de costo convexas. Sin embargo, empíricamente se ha demostrado que algunos métodos AGO (por ejemplo: el método Adam) a menudo superan problemas con funciones de costo no convexas, véase Kingma and Ba (2015). En Adolphs et al. (2019) se muestra que algunos métodos AGO (en particular los métodos: AdaGrad, RMSProp y Adam) se pueden interpretar como métodos de región de confianza de primer orden con restricciones elipsoidales, los cuales superan a su contraparte esférica. Debido a que las funciones de costo en FWI son altamente no lineales, resulta un problema difícil asegurar la convergencia al mínimo global. Para asegurar, al menos, convergencia a un mínimo local los métodos AGO requieren una calibración adecuada del step-length; una mala selección/calibración de este parámetro puede generar altas fluctuaciones de la función de costo alrededor de un mínimo local o en el peor de los casos, el método puede diverger (Ruder, 2016), véase la figura 3.2.

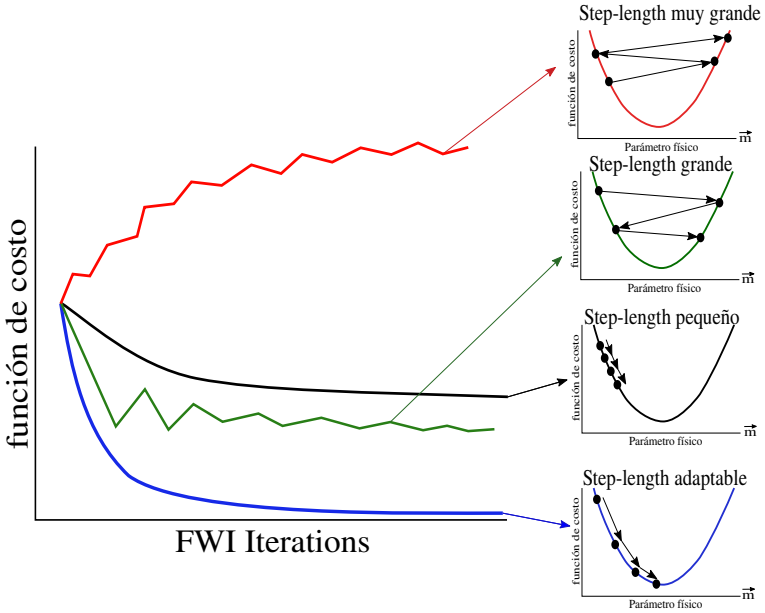

Figura 3.2: Efecto del step-length en FWI.

En general los métodos AGO utilizan actualizaciones de gradientes ponderados mediante raíces cuadradas del *promedio móvil exponencial (exponential moving average)* del cuadrado de los gradientes de las iteraciones anteriores, lo que permite automáticamente adaptar el step-length de acuerdo a la evolución del proceso de minimización, asignando mayor peso a los gradientes más recientes y asignando menos peso a los primero gradientes. El hecho de que el step-length se puede adaptar de distintas formas, da lugar a una familia de distintos métodos AGO, véase Reddi et al. (2018).

A continuación daremos una reseña de los métodos AGO que usaremos en este trabajo: AdaGrad, RMSprop, Adadelta, Adam, Nadam, AMSGrad y RAdam; no ahondaremos en sus detalles teóricos ya que se pueden consultar en sus respectivas referencias.

Consideremos $G(m_k) = \frac{\partial \epsilon(\vec{m})}{\partial m_k}$, una componente del gradiente calculado con el método de estado adjunto, para el modelo de parámetros físicos m_k en la k-ésima iteración de FWI y α un step-length dado. Los parámetros: G, S, D y V (que aparecen en los siguientes métodos de optimización), tienen la misma dimensión que el modelo de parámetros m_k.

3.0.1. Método AdaGrad

Adaptive gradient o método AdaGrad (Duchi et al., 2011) es un algoritmo de optimización basado en gradiente, cuyo objetivo es ir adaptando el step-length en en cada iteración de tal forma que se apliquen pequeños cambios (i.e. step-length pequeño) en parámetros cuyas características cambian con mucha frecuencia y se realicen actualizaciones más grandes (i.e. step-length grande) en parámetros cuyas características cambian con poca frecuencia. Esto lo hace adecuado para tratar con datos dispersos por lo que es uno de los optimizadores más utilizados por Google Inc. para entrenar redes neuronales de gran escala (Ruder, 2016). El step-length es dividido por la raíz cuadrada de S, que corresponde a la acumulación de la suma de los cuadrados de los gradientes, la fórmula está dada por:

$$m_{k+1} = m_k - \frac{\alpha}{\sqrt{S_k} + \eta} \cdot G(m_k), \tag{3.1}$$

con

$$S_k = S_{k-1} + [G(m_k)]^2, \tag{3.2}$$

donde S se inicializa con 0 y η es un parámetro de suavizamiento para evitar divisiones entre cero (nostros usamos $\eta = 1 \times 10^{-7}$).

3.0.2. Método RMSprop

Root Mean Square prop o método RMSprop (Tieleman and Hinton, 2012) es un método de gradiente adaptable. A diferencia del método AdaGrad, éste considera el promedio móvil exponencial de todos los gradientes (como en el método de Momentum (Polyak, 1964)), en vez de acumular la suma de los cuadrados de los gradientes. La fórmula está dada como sigue:

$$m_{k+1} = m_k - \frac{\alpha}{\sqrt{S_k} + \eta} \cdot G(m_k), \tag{3.3}$$

con

$$S_k = \beta S_{k-1} + (1 - \beta)[G(m_k)]^2 \tag{3.4}$$

donde S se inicializa con 0; η es un término de suavizamiento para evitar divisiones entre cero (nosotros usamos $\eta = 1 \times 10^{-6}$) y $\beta = 0{,}9$ es un valor recomendado por los autores del método.

3.0.3. Método Adadelta

El método Adadelta (Zeiler, 2012) es una extensión del método AdaGrad que en vez de acumular el cuadrado de los gradientes desde el principio (lo cual puede reducir el step-length de forma prematura), reduce gradualmente la contribución de los gradientes anteriores, lo cual puede ayudar a escapar de mínimos locales. La fórmula está dada por:

$$m_{k+1} = m_k - \alpha \frac{\sqrt{D_{k-1} + \eta}}{\sqrt{S_k} + \eta} \cdot [G(m_k)], \tag{3.5}$$

con

$$D_k = \beta D_{k-1} + (1-\beta)[m_k - m_{k-1}]^2, \tag{3.6}$$
$$S_k = \beta S_{k-1} + (1-\beta)[G(m_k)]^2, \tag{3.7}$$

donde S y D se inicializan con 0; $\beta = 0{,}95$ y $\eta = 1 \times 10^{-6}$ son valores usados por la librería de machine-learning, Keras (Chollet, 2017).

Originalmente este método fue diseñado con un step-length fijo unitario, lo cual se vuelve incompatible con el proceso de multiscaling en FWI, ya que se requiere un step-length que cambie en función de las frecuencias.

3.0.4. Método Adam

Adaptive moment estimation o método Adam (Kingma and Ba, 2015), es una combinación del método de momentum (Polyak, 1964) y el método RMSprop, ya que el gradiente principal se calcula mediante un promedio móvil exponencial de los gradientes (como en el método de momentum) y el tamaño del step-length considera la raíz cuadrada del promedio móvil exponencial del cuadrado de los gradientes (como en el método RMSprop). La combinación de dichas estrategias hacen que el proceso de minimización con el método Adam se asemeje al de una pelota que rueda con fricción sobre la superficie de la función de costo en busca de alguna región plana. La fórmula está dada por:

$$m_{k+1} = m_k - \frac{\alpha}{\sqrt{\hat{S}_k} + \eta} \cdot \hat{V}_k, \tag{3.8}$$

donde

$$\hat{V}_k = \frac{V_k}{1-\beta_1^k} \tag{3.9}$$
$$\hat{S}_k = \frac{S_k}{1-\beta_2^k}, \tag{3.10}$$

con

$$V_k = \beta_1 V_{k-1} + (1-\beta_1) G(m_k) \tag{3.11}$$
$$S_k = \beta_2 S_{k-1} + (1-\beta_2)[G(m_k)]^2, \tag{3.12}$$

el súper índice k, en β_1 y β_2, representa una potencia correspondiente al número de la iteración; V y S se inicializan con 0. $\beta_1 = 0{,}9$, $\beta_2 = 0{,}999$ y $\eta = 1 \times 10^{-8}$ son valores recomendados por los autores del método.

3.0.5. Método Nadam

Nesterov-accelerated Adaptive Moment Estimation o método Nadam, véase Dozat (2016), es una combinación del método de gradiente acelerado de Nesterov (método NAG (Nesterov, 1983)) y el método Adam, lo que resulta en una versión acelerada del método Adam.
Para describir esta combinación, notemos que el método Adam se puede escribir como sigue:

$$m_{k+1} = m_k - \frac{\alpha}{\sqrt{\hat{S}_k} + \eta} \left[\beta_1 \hat{V}_{k-1} + \frac{1-\beta_1}{1-\beta_1^k} \cdot G(m_k) \right], \tag{3.13}$$

entonces se aplica el método de Nesterov para actualizar el gradiente en el siguiente paso, sustituyendo $\hat{V}_{k-1}$ por el actual $\hat{V}_k$, obteniendo:

$$m_{k+1} = m_k - \frac{\alpha}{\sqrt{\hat{S}_k} + \eta} \left[\beta_1 \hat{V}_k + \frac{1-\beta_1}{1-\beta_1^k} \cdot G(m_k) \right], \tag{3.14}$$

donde

$$\hat{V}_k = \frac{V_k}{1-\beta_1^k} \tag{3.15}$$
$$\hat{S}_k = \frac{S_k}{1-\beta_2^k}, \tag{3.16}$$

con

$$V_k = \beta_1 V_{k-1} + (1-\beta_1) G(m_k) \tag{3.17}$$
$$S_k = \beta_2 S_{k-1} + (1-\beta_2)[G(m_k)]^2, \tag{3.18}$$

el súper índice k, en β_1 y β_2, representa una potencia correspondiente al número de la iteración; V y S se inicializan con 0. $\beta_1 = 0{,}9$, $\beta_2 = 0{,}999$ y $\eta = 1 \times 10^{-7}$ son valores usados por la librería de machine-learning, Keras (Chollet, 2017).

3.0.6. Método AMSGrad

El método AMSGrad (Reddi et al., 2018) es uno de los métodos AGO más reciente desarrollados por Google Inc. Es una variante del método Adam que modifica el parámetro $\hat{S}$ de tal forma que se asegure que S siempre sea más grande que su correspondiente valor correspondiente en iteración anterior. Esto permite que el step-length se estabilice conforme incrementa el número de iteraciones, lo que resulta en una versión más estable del método Adam. La fórmula está dada por:

$$m_{k+1} = m_k - \frac{\alpha}{\sqrt{\hat{S}_k} + \eta} \cdot \hat{V}_k, \tag{3.19}$$

donde

$$\hat{S}_k = max(\hat{S}_{k-1}, S_k), \tag{3.20}$$

con

$$V_k = \beta_1 V_{k-1} + (1 - \beta_1) G(m_k), \tag{3.21}$$
$$S_k = \beta_2 S_{k-1} + (1 - \beta_2)[G(m_k)]^2, \tag{3.22}$$

donde V y S se inicializan con 0. $\beta_1 = 0{,}9$, $\beta_2 = 0{,}999$ y $\eta = 1 \times 10^{-7}$ son valores usados en la librería Keras (Chollet, 2017). Note que $\hat{S}_{k-1}$ y S_k en la ecuación (3.20), son arreglos del mismo tamaño que el gradiente, por lo que escogemos el valor máximo entre ambos arreglos, usando la norma matricial L_2.

3.0.7. Método RAdam

Rectified Adam o método RAdam (Liyuan et al., 2019) es una variante del método Adam. El método RAdam introduce un término rectificador que permite estabilizar la variabilidad del step-length durante las primeras iteraciones del método Adam. Esto último puede ser un arma de dos filos pues se corre el riesgo de estabilizar el step-length en un valor muy pequeño antes de llegar a un mínimo local o en el mejor de los casos en muy pocas iteraciones se puede llegar muy cerca del mínimo global. La fórmula está dada por:

$$m_{k+1} = m_k - \frac{\alpha}{\sqrt{\hat{V}_k} + \eta} \cdot r_k \hat{S}_k, \tag{3.23}$$

donde

$$\hat{S}_k = \frac{S_k}{1 + \beta_1^k}, \tag{3.24}$$

$$\hat{V}_k = \begin{cases} \frac{V_k}{1-\beta_2^k}, & \text{if } p_k > 4 \\ 1, & \text{en otro caso} \end{cases} \tag{3.25}$$

$$r_k = \begin{cases} \sqrt{\frac{(p_k-4)(p_k-2)p_\infty}{(p_\infty-4)(p_\infty-2)p_k}}, & \text{if } p_k > 4 \\ 1, & \text{en otro caso} \end{cases} \tag{3.26}$$

$$\begin{aligned} S_k &= \beta_1 S_{k-1} + (1-\beta_1) G(m_k), && (3.27) \\ V_k &= \beta_2 V_{k-1} + (1-\beta_2)[G(m_k)]^2, && (3.28) \\ p_k &= p_\infty - \frac{\beta_2^k}{1-\beta_2^k} \cdot 2k, && (3.29) \\ p_\infty &= \frac{2}{1-\beta_2} - 1, && (3.30) \end{aligned}$$

el súper índice k, en β_1 y β_2, representa una potencia que corresponde al número de la iteración; V y S se inicializan con 0. $\beta_1 = 0{,}9$, $\beta_2 = 0{,}999$ y $\eta = 1 \times 10^{-8}$ son los mismos valores usados en el método Adam.

3.1. Poniendo a prueba los métodos AGO

Para medir el alcance de los métodos AGO que acabamos de revisar y poder visualizar su comportamiento, nos restringimos a un simple caso $1D$, por lo que trataremos de minimizar la función:

$$f(x) = 10\sin(5(x - \frac{\pi}{2})) + 3x^2. \tag{3.31}$$

Como se puede ver en la figura 3.3 a) esta función tiene una gran cantidad de mínimos locales, así que resulta imposible alcanzar su mínimo global (en $x = 0$) con métodos clásicos de descenso y al mismo tiempo pone a prueba los métodos AGO.

Considerando $x_0 = -5{,}6$ como punto de partida y realizando 100 iteraciones con cada método, en la figura 3.3 a) podemos ver que mientras los métodos AdaGrad y RMSProp se quedan atrapados en un mismo mínimo local, el método Adadelta alcanza un mínimo local mejor que el obtenido por los dos métodos anteriores. Sin embargo, los métodos más recientes: Adam, Nadam, AMSGrad y Radam alcanzan con facilidad el mínimo global.

En la figura 3.3 b), se puede mostrar que los métodos AdaGrad y RMSProp tienen un comportamiento similar, a diferencia de que RMSProp muestra fluctuaciones en la curva de error originadas por el intento de escapar del mínimo local. Los demás métodos logran estabilizarse después de cierto número de iteraciones.

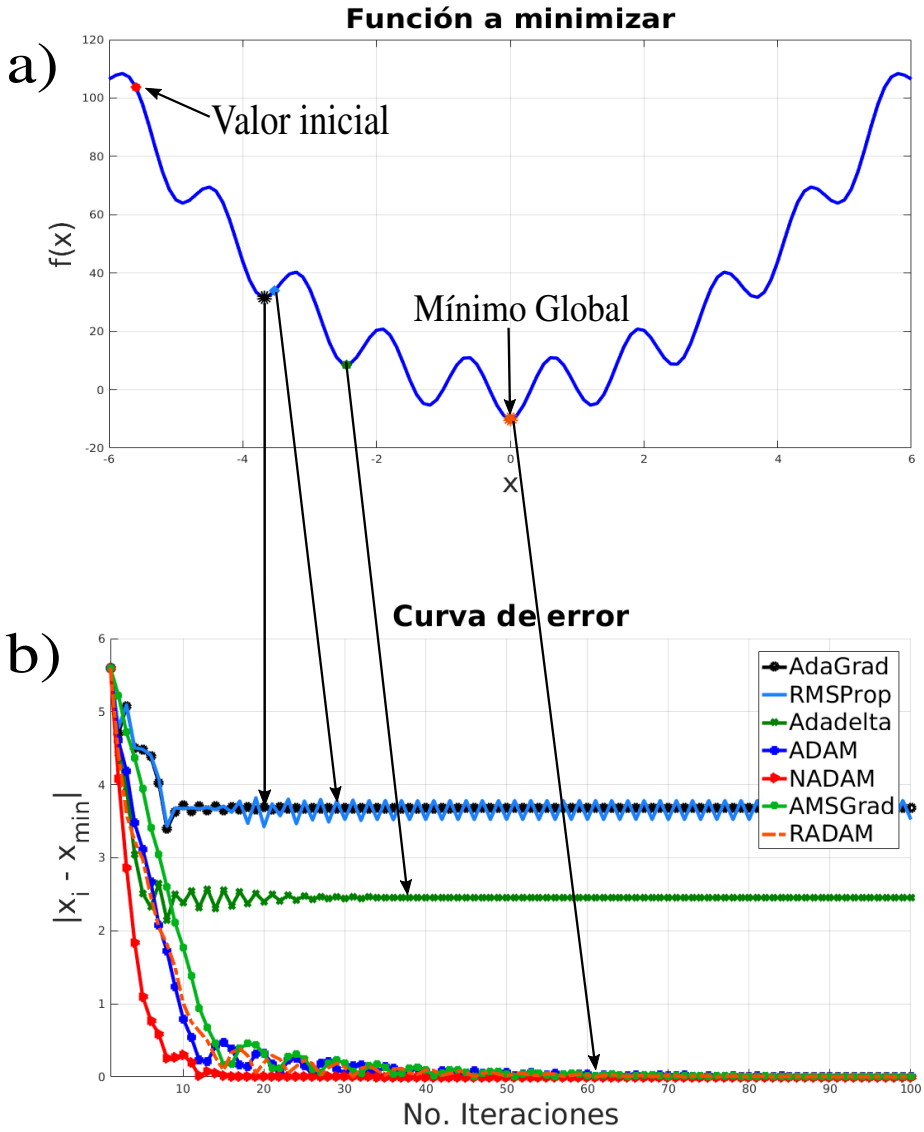

Figura 3.3: a): Función de a minimizar. b): Evolución del error al aplicar distintos métodos AGO.

Capítulo 4

Una nueva fórmula para asignar el step-length en los métodos AGO para FWI

Los escalares α, β_1 y β_2 (conocidos como hiper-parámetros en la literatura del Machine Learning) usados en los métodos AGO, juegan un papel importante en el proceso de minimización. Estos parámetros son extremadamente sensibles y calibrarlos resulta difícil ya que dependen del problema al que se estén aplicando (Sun et al., 2020). Una selección de valores ligeramente diferente puede alterar drásticamente la velocidad de convergencia o en el peor de los casos, conducir a la divergencia.

La mayoría de paquetes científicos que implementan métodos AGO, como TensorFlow, Keras, Caffe, entre otros; tienen valores predeterminados del step-length (o learning rate). La mayoría de reglas deterministas de step-length para métodos AGO aplican un factor de decaimiento exponencial de la forma $\alpha(k) \approx \varphi/\sqrt{k}$, donde φ es una constante y k corresponde al número de la $k-$ésima iteración, véase Reddi et al. (2018).

Otras fórmulas alternativas para la estimación de step-length en FWI son los métodos Barzilai-Borwein (BB) (dos Santos and Pestana, 2015). Sin embargo, los métodos BB estan diseñados para optimización basada en gradiente simple y no preservan un decaimiento exponencial como el que requieren los métodos AGO para FWI con multiscaling. También existen métodos analíticos como el método ASL (Pica et al., 1990), que podrían ser la mejor opción para la construcción del step-length, sin embargo, tampoco preserva el decaimiento exponencial que se requiere en los métodos AGO (Ma et al., 2019); además el método ASL requiere realizar una propagación directa extra por

cada iteración de FWI.

En consecuencia, hemos diseñado una fórmula para asignar un step-length que depende de las frecuencias (usadas en el multiscaling) y satisface los requerimientos de los métodos AGO para adaptarse en FWI con multiscaling. La construcción de esta fórmula se basa en los siguientes pasos:

- **Primer paso**. Suponiendo que $\{f_{min} = f_1, f_2, ..., f_n = f_{max}\}$ son las frecuencias a usar en el proceso de multiscaling, asignamos un step-length a cada frecuencia preservando un decaimiento exponencial conforme incrementan las frecuencias mediante la siguiente relación:

$$\hat{\alpha}(f) = Q\left(\frac{f_{max}}{f}\right)^p, \tag{4.1}$$

  donde f_{max} es la frecuencia máxima en el proceso de multiscaling. Las constantes $p > 0$ y $Q > 0$ dependen de cada método AGO que se aplique en FWI. Notemos que podemos establecer un valor para Q al evaluar f_{max} en la ecuación (4.1), obteniendo $\hat{\alpha}(f_{max}) = Q$. Una vez que tenemos un valor para Q, podemos evaluar f_{min} en la misma ecuación y poder establecer un valor para p. En la siguiente sección mostraremos el criterio que usamos para asignar valores a p y Q.

- **Segundo paso**. Dado que se debe realizar cierto número de iteraciones de FWI con cada frecuencia f_i, construimos una reparametrización que agregue un decaimiento lineal al cambiar de $\hat{\alpha}(f_i)$ a $\hat{\alpha}(f_{i+1})$. Esto reducirá las fluctuaciones en la función de costo y elimina la alta variabilidad durante la actualización de los modelos. Para esto, el step-length decrece conforme incrementan las frecuencias e iteraciones de FWI. Dicha reparametrización está dada por:

$$\alpha(k, f_i) = \begin{cases} \left(\frac{\hat{\alpha}(f_{i+1}) - \hat{\alpha}(f_i)}{n_{f_i} - 1}\right)(k-1) + \hat{\alpha}(f_i), & \text{if } f_i < f_{max}, \\ \hat{\alpha}(f_{max}), & \text{if } f_i = f_{max}, \end{cases} \tag{4.2}$$

  k indica la k-ésima iteración de FWI correspondiente a la frecuencia f_i con $1 \leq k \leq n_{f_i}$. n_{f_i} es el máximo número de iteraciones de FWI por cada frecuencia.

Por lo tanto en la k-ésima iteración de FWI que correspondiente a la frecuencia f_i, tendremos un step-length específico $\alpha(k, f_i)$. Notemos que la ecuación (4.2) viene de una correspondencia lineal entre los valores de la ecuación (4.1) y el número de iteraciones de FWI. Los valores de la ecuación (4.1) están representados por las estrellas que se muestran en la figura 4.1 y los valores de la ecuación (4.2) se representan con los círculos que se muestran en la misma figura.

Debido a que los métodos clásicos de asignación de step-length dependen del gradiente, resulta imposible conocer los valores de los step-lengths a usar en cada iteración de FWI antes de que iniciar el proceso, sin embargo, con nuestra fórmula de asiganción de step-length es posible conocer el valor exacto de los step-lengths a usar en cada iteración antes de iniciar la FWI, lo cual se debe a que nuestra regla de step-length no depende del gradiente. Como resultado de este proceso, evitamos el costoso proceso de line-search en el flujo de trabajo convencional de FWI. Por lo tanto en cada iteración de FWI sólo se requiere una propagación directa para actualizar el modelo en una dirección correcta. El flujo de trabajo que finalmente hemos implementado se muestra en la figura 4.2.

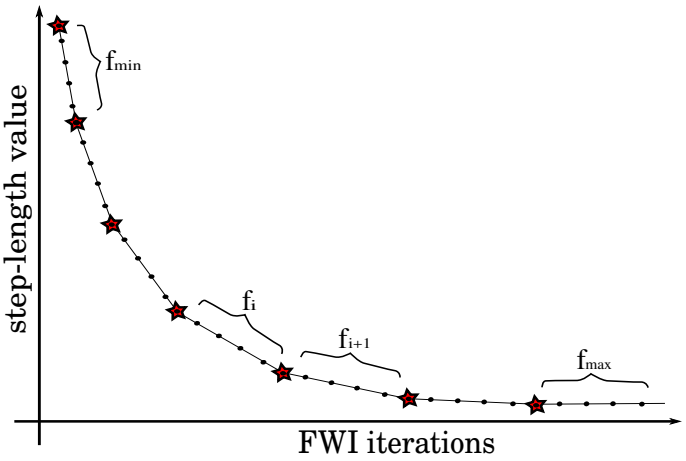

Figura 4.1: Decaimiento de los valores de step-length (para los métodos AGO) en cada lote de iteraciones de FWI correspondientes a las frecuencias del multiscaling.

4.1. Criterio empírico para calibrar el step-length en los métodos AGO

La mayor dificultad que tienen los métodos AGO es la calibración de hiper-parámetros. En consecuencia, asignar valores a p y Q en la ecuación (4.1) puede ser complicado. Durante experimentos a prueba y error encontramos un criterio empírico que nos permite escoger valores adecuados para p y Q, que ofrecen resultados finales comparables a los que ofrece el popular método L-BFGS. El criterio consiste en los siguientes dos puntos:

- Utilizar la frecuencia máxima f_{max} (sabiendo que $\alpha(1, f_{max}) = Q$) para encontrar un valor Q tal que el primer modelo generado con dicha frecuencia máxima, denotado por $m_1(f_{max})$ satisfaga la siguiente condición:

$$A_{min} \leq \frac{\parallel m_1(f_{max}) - m_0 \parallel}{max\{\parallel m_1(f_{max}) \parallel, \parallel m_0 \parallel\}} \times 100\,\% \leq A_{max}, \tag{4.3}$$

  donde m_0 es el modelo inicial con el que se inicia la FWI y $\parallel \cdot \parallel$ es la norma L_2.

- Habiendo fijado un valor Q, usamos la frecuencia mínima f_{min} para asignar un valor a p tal que el primer modelo generado con dicha frecuencia mínima, denotado por $m_1(f_{min})$ satisfaga la siguiente condición:

$$B_{min} \leq \frac{\parallel m_1(f_{min}) - m_0 \parallel}{max\{\parallel m_1(f_{min}) \parallel, \parallel m_0 \parallel\}} \times 100\,\% \leq B_{max}. \tag{4.4}$$

Los valores que usamos para A_{min}, A_{max}, B_{min} y B_{max} se muestran en la sección de experimentos numéricos.

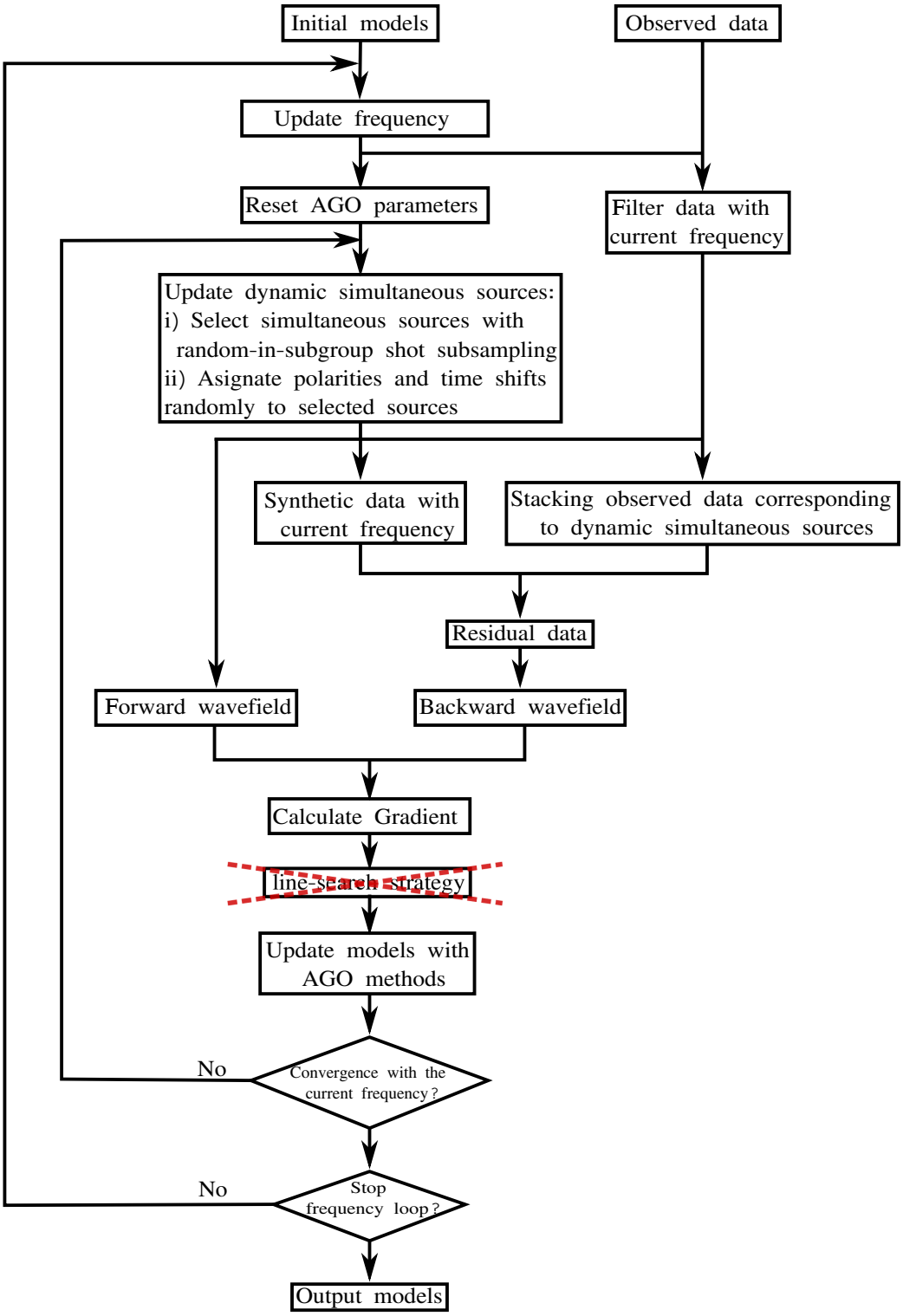

Figura 4.2: Flujo de trabajo para FWI usando las técnicas de fuentes simultáneas dinámicas y multiscaling. Debido a que los métodos AGO evitan la estrategia de line-search, hemos tachado ese paso.

Capítulo 5

Experimentos numéricos

A continuación mostraremos los resultados que se obtienen al realizar la FWI para recobrar los modelos de velocidades: Canadian overthrust BP y Marmousi; usando la técnica de fuentes simultáneas dinámicas con multiscaling que revisamos anteriormente, basada en el flujo de trabajo que se muestra en la figura 4.2. Hemos implementado desde ceros cada proceso de este flujo de trabajo, incluyendo los métodos de optimización descritos anteriormente. En cada iteración de FWI, el gradiente se calcula con la ecuación (2.25), donde los campos directo y adjunto se obtienen al resolver las ecuaciones (1.27 y 2.22) usando diferencias finitas espaciales de cuarto orden (con malla alternada) y de segundo orden en tiempo (Levander, 1988), considerando fronteras absobentes C-PML (Komatitsch and Martin, 2007) con 25 nodos de espesor.

Mostraremos los resultados de la FWI basada en las normas L_1 y L_2. Para la norma L_2 aplicaremos los siete métodos AGO revisados en el Capítulo 3 y para la norma L_1 sólo mostraremos resultados con los métodos más recientes: Adam, Nadam, AMSGrad y RAdam; debido a que con los otros métodos el proceso se vuelve inestable.

Las fuentes a activar en cada super-shot irán cambiando en cada iteración de la FWI de acuerdo a lo mostrado en la Sección 2.3; dichas fuentes estarán ubicadas a 10m de la superficiel libre y el máximo número de fuentes a activar en cada super-shot durante la frecuencia máxima es de $n_s = (1/5) * nx$, siendo nx el número de puntos en la dirección horizontal de la malla.

La onduleta de la fuente corresponde a un pulso de Ricker con frecuencias especificadas en cada experimento; para el proceso de multiscaling usamos 7 frecuencias diferentes con un máximo de 100 iteraciones de FWI por

frecuencia. Para garantizar estabilidad y baja dispersión numérica, usaremos incrementos espaciales de $dx = dz = 10$m e incrementos temporales de $dt = 0{,}001$s.

Para medir la evolución del error en cada iteración de FWI, usamos la función:

$$f(m_k) = \frac{\parallel d_{synt}(m_k) - d_{obs} \parallel}{\parallel d_{obs} \parallel}, \tag{5.1}$$

siendo $d_{syn}(m_k)$ los sismogramas sintéticos generados con el k-ésimo modelo de velocidades m_k y d_{obs} son los sismogramas reales observados. Dichos sismogramas son generados con un super-shot formado por un conjunto de fuentes fijas cuya separación entre ellas es de 100m. Los datos observados se construirán usando el modelo Canadiense o el modelo de Marmousi.

Para cada uno de los métodos AGO aquí considerados, fijamos los valores de los parámetros β, β_1 y β_2 con los valores recomendados que se muestran en el Capítulo 3; cada vez que el proceso de multiscaling realiza un cambio en las frecuencias, actualizamos los parámetros S, D y V (que aparecen en los métodos AGO) con ceros.

Para incrementar la resolución del modelo de velocidades en zonas profundas, reescalaremos el gradiente usando el factor de iluminación (Kaelin and Guitton, 2006) dado por:

$$Illum(X) = \int_0^T v^2(X,t)dt, \tag{5.2}$$

donde $v(X,t)$ es el campo de velocidades generado con el super-shot y modelo de velocidades correspondientes a la iteración actual, por lo que el gradiente se actualiza como:

$$G(X) \leftarrow \frac{G(X)}{[Illum(X) + \epsilon]}. \tag{5.3}$$

$G(X)$ es el gradiente correspondiente a la ecuación (2.25) y $\epsilon = 1 \times 10^{-20}$ para evitar divisiones entre cero.

El step-length α se calcula de acuerdo a la ecuación (4.2), por lo que su valor en cada iteración es conocido antes de que inicie el proceso de FWI. El valor de la constante p en la ecuación (4.1) ha mostrado ser independiente de los datos reales del problema (los cuales dependen del modelo Canadiense o del modelo de Marmousi) y es diferente para cada método AGO. Sin embargo, el valor de la costante Q en la ecuación (4.1) puede cambiar dependiendo

de los datos reales del problema.

Los valores de p y Q que usamos en este trabajo se muestran en los Cuadros 5.1 y 5.2. Dichos valores satisfacen los criterios de estabilidad dados en las ecuaciones (4.3)-(4.4). Para el modelo Canadiense usamos: $A_{min} = 0{,}05\,\%$, $A_{max} = 0{,}1\,\%$, $B_{min} = 0{,}1\,\%$ y $B_{max} = 0{,}3\,\%$. Para el modelo de Marmousi usamos: $A_{min} = 0{,}06\,\%$, $A_{max} = 0{,}2\,\%$, $B_{min} = 0{,}2\,\%$ y $B_{max} = 0{,}4\,\%$.

Cuadro 5.1: Valores de los parámetros p y Q usados en la ecuación (4.1) para cada método AGO aplicados en FWI basada en la norma L_2 con diferentes modelos de velocidades.

AGO method	p	Q	Q
		Canadian model	Marmousi model
AdaGrad	2	3	2,5
RMSprop	3	0,5	0,4
Adadelta	3	0,001	0,0009
Adam	0,05	4	6
Nadam	0,05	4	8
AMSGrad	0,05	1	2
RAdam	1,1	10	20

Cuadro 5.2: Valores de los parámetros p y Q usados en la ecuación (4.1) para cada método AGO aplicados en FWI basada en la norma L_1 con diferentes modelos de velocidades.

AGO method	p	Q	Q
		Canadian model	Marmousi model
Adam	0,05	4	6
Nadam	0,05	4	7
AMSGrad	0,05	1	1
RAdam	1,1	4	6,7

5.0.1. Método L-BFGS

Para medir el alcance de los métodos AGO en FWI, compararemos su desempeño contra el que ofrece el método L-BFGS (Nocedal and Writght, 1999). El método L-BFGS es un método Quasi-Newton que aproxima la dirección $d_k = -H_k^{-1}G_k$ en la ecuación (2.36) sin calcular la matriz completa H_k^{-1}, es decir, es una versión con memoria limitada del método BFGS (llamado así por los nombres de sus desarrolladores: Broyden, Fletcher, Goldfarb y Shanno). Para esto se requiere almacenar los cambios en los gradientes: $\gamma_k = G_{k+1} - G_k$ y los cambios en los modelos: $s_k = m_{k+1} - m_k$, con $r_k = 1/\gamma_k^T s_k$, de las últimas N iteraciones. Por lo tanto, con esta información y utilizando el ya conocido procedimiento de *two-loop recursion* (que se muestra a continuación), este método puede calcular una aproximación de $d_k = -H_k^{-1}G_k$ como sigue:

Algorithm 1: two-loop recursion L-BFGS algorithm

$q \leftarrow G_k$;
for $i = k-1, ..., k-N$ **do**
  $\alpha_i \leftarrow r_i s_i^T q$;
  $q \leftarrow q - \alpha_i \gamma_i$;
end
$z \leftarrow \frac{s_{k-1}^T \gamma_{k-1}}{\gamma_{k-1}^T \gamma_{k-1}} q$;
for $i = k-N, ..., k-1$ **do**
  $\beta \leftarrow r_i \gamma_i^T z$;
  $z \leftarrow z + s_i(\alpha_i - \beta)$;
end
$d_k = -z$.

En nuestros experimentos numéricos con L-BFGS usaremos un tamaño de memoria de $N = 10$. Para satisfacer las condiciones de Wolfe [1] (las cuales aseguran decremento óptimo en la función de costo) usaremos el método ASL (Analytical step-length), tal como se muestra en Pica et al. (1990) ya que resulta la mejor opción al momento de aplicar L-BFGS en FWI (Ma et al., 2019).

[1] Si se minimiza una función $f(\vec{x})$ mediante, $\vec{x}_{k+1} = \vec{x}_k - \alpha_k \vec{d}_k$, con una dirección de descenso $\vec{d}_k$, decimos que α_k satisface las condiciones de Wolfe si:
i) $f(\vec{x}_{k+1}) \leq f(\vec{x}_k) + c_1 \alpha_k \vec{d}_k^T \nabla f(\vec{x}_k)$,
ii) $-\vec{d}_k^T \nabla f(\vec{x}_{k+1}) \leq -c_2 \vec{d}_k^T \nabla f(\vec{x}_k)$,
con $0 < c_1 < c_2 < 1$.

El método ASL sólo requiere una propagación directa extra en cada iteración ya que su regla de actualización de step-length está dada por:

$$\alpha = \alpha_t \frac{\sum_s \sum_g [d_{synt}(m_k + \alpha_t d_k) - d_{synt}(m_k)]^T [d_{synt}(m_k + \alpha_t d_k) - d_{obs}]}{\sum_s \sum_g [d_{synt}(m_k + \alpha_t d_k) - d_{synt}(m_k)]^T [d_{synt}(m_k + \alpha_t d_k) - d_{synt}(m_k)]}, \tag{5.4}$$

donde α_t es un step-length de prueba. Aquí usamos, $\alpha_t = \zeta(\| m_k \| / \| d_k \|)$ (siendo ζ un coeficiente de proporcionalidad y $\| \cdot \|$ la norma L_2). α_t necesita satisfacer la siguiente condición:

$$max(|\alpha_t d_k|) \leq \frac{1}{100} max(m_k). \tag{5.5}$$

$d_{synt}(m_k)$ son los datos aproximados con el modelo m_k y d_{obs} son los datos observados. Los parámetros s y g representan las fuentes y receptores que participan en cada iteración de FWI.

5.0.2. Inversión del modelo de velocidades: Canadian overthrust BP

En nuestro primer experimento tomamos una muestra del modelo de velocidades: Canadian overthrust BP (Gray and Marfurt, 1995), en una malla de tamaño $nz \times nx = 250 \times 556$ (sin incluir las fronteras C-PML). Este modelo de velocidades se muestra en la figura 5.1 (izquierda). El modelo inicial de velocidades corresponde a un modelo de capas planas como se muestra en la figura 5.1 (derecha), el cual se obtuvo mediante un promedio de los valores (a lo largo de cada línea horizontal) del modelo real.

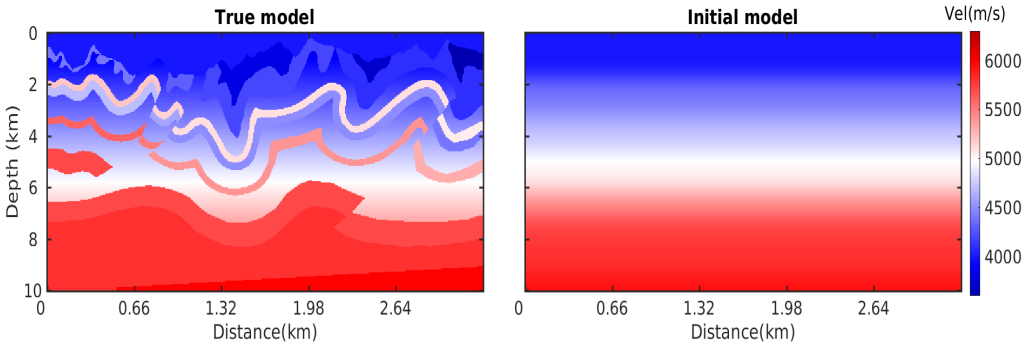

Figura 5.1: Modelo de velocidades: Canadian overthrust BP (izquierda); Modelo de velocidades inicial (derecha).

El campo de onda acústico es registrado por 556 receptores localizados a 10m bajo la superficie libre durante un tiempo de $T = 3{,}5$s lo cual requiere 3500 pasos de tiempo. Aunque usamos un incremento de $dx = 10$, las escalas que se muestran en las figuras 5.3, 5.4, 5.5 y 5.6, corresponden a las escalas que se muestran en Gray and Marfurt (1995), por lo que el modelo de velocidades tiene una longitud de 3,3km y una profundidad de 10km. Durante el proceso de multiscaling usamos frecuencias de: 4Hz, 8Hz, 14Hz, 20Hz, 24Hz, 32Hz y 40Hz, partiendo del modelo inicial y aplicando el flujo de trabajo de FWI que se muestra en la figura 4.2.

Primero invertimos ambos modelos de velocidades (Canadiense y Marmousi) aplicando el método L-BFGS y los siete métodos AGO: AdaGrad, RMSprop, Adadelta, Adam, Nadam, AMSGrad y RAdam con FWI basada en la norma L_2. Como resultado, obtenemos los modelos finales que se muestran en la figura 5.3. Posteriormente, aplicamos FWI basada en la norma L_1 usando los métodos AGO más recientes (Adam, Nadam, AMSGrad and RAdam), obteniendo los modelos finales que se muestran en la figura 5.4. Las

diferencias entre el modelo real de velocidades y los modelos aproximados, obtenidos de la FWI basada en la normas L_2 y L_1, se muestran en las figuras 5.5 y 5.6, respectivamente. La curva de error en cada iteración de FWI (usando escala logarítmica) obtenida con la ecuación (5.1), se muestra en la figura 5.2. Los perfiles de profundidad a $x = 820$m y a $x = 2$km se muestran en las figuras 5.7 y 5.8, respectivamente. En la figura 5.9 se muestra la comparación del residual de datos obtenidos (diferencia entre datos observados y aproximados) de la FWI basada en la norma L_2 (con L-BFGS, Adadelta, AMSGrad and RAdam) y de la FWI basada en la norma L_1 (con Adam, Nadam, AMSGrad and RAdam); correspondientes a una fuente ubicada en el punto medio de la línea de receptores. Sólo mostramos dichos resultados (para la FWI basada en la norma L_2), pues los resultados obtenidos con el método Adadelta son muy similares a los que se obtienen con los métodos AdaGrad y RMSProp; los resultados obtenidos con el método AMSGrad son muy similares a los obtenidos con Adam y Nadam. Para visualizar la precisión en las formas de onda, comparamos las trazas reales con las trazas aproximadas (correspondientes al receptor No. 140) generadas con los modelos finales de la FWI basada en la norma L_2 (figura 5.10) y basada en la norma L_1 (figura 5.11).

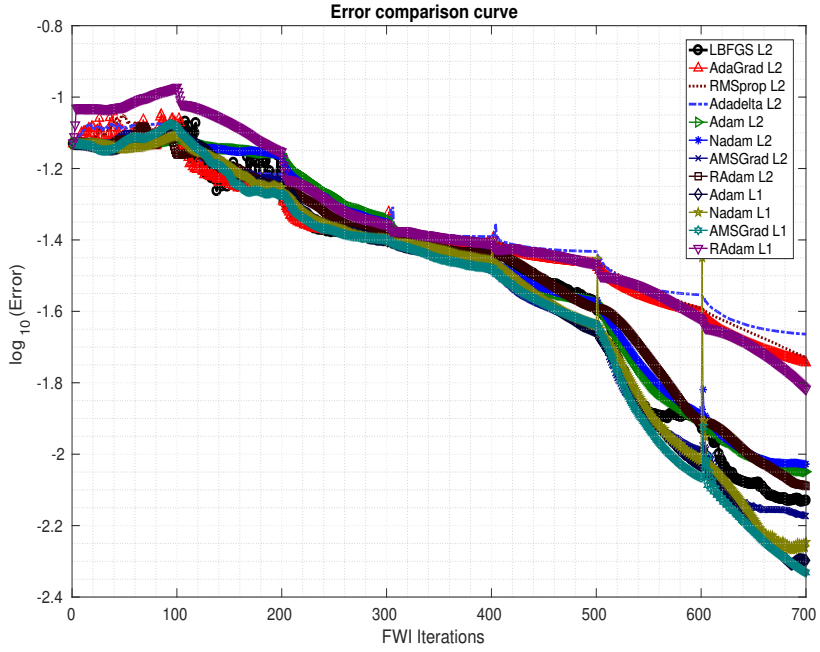

Figura 5.2: Curva de error en cada iteración de FWI (usando escala logarítmica) para cada uno de los métodos de optimización aplicados en la FWI basada en las normas L_2 y L_1.

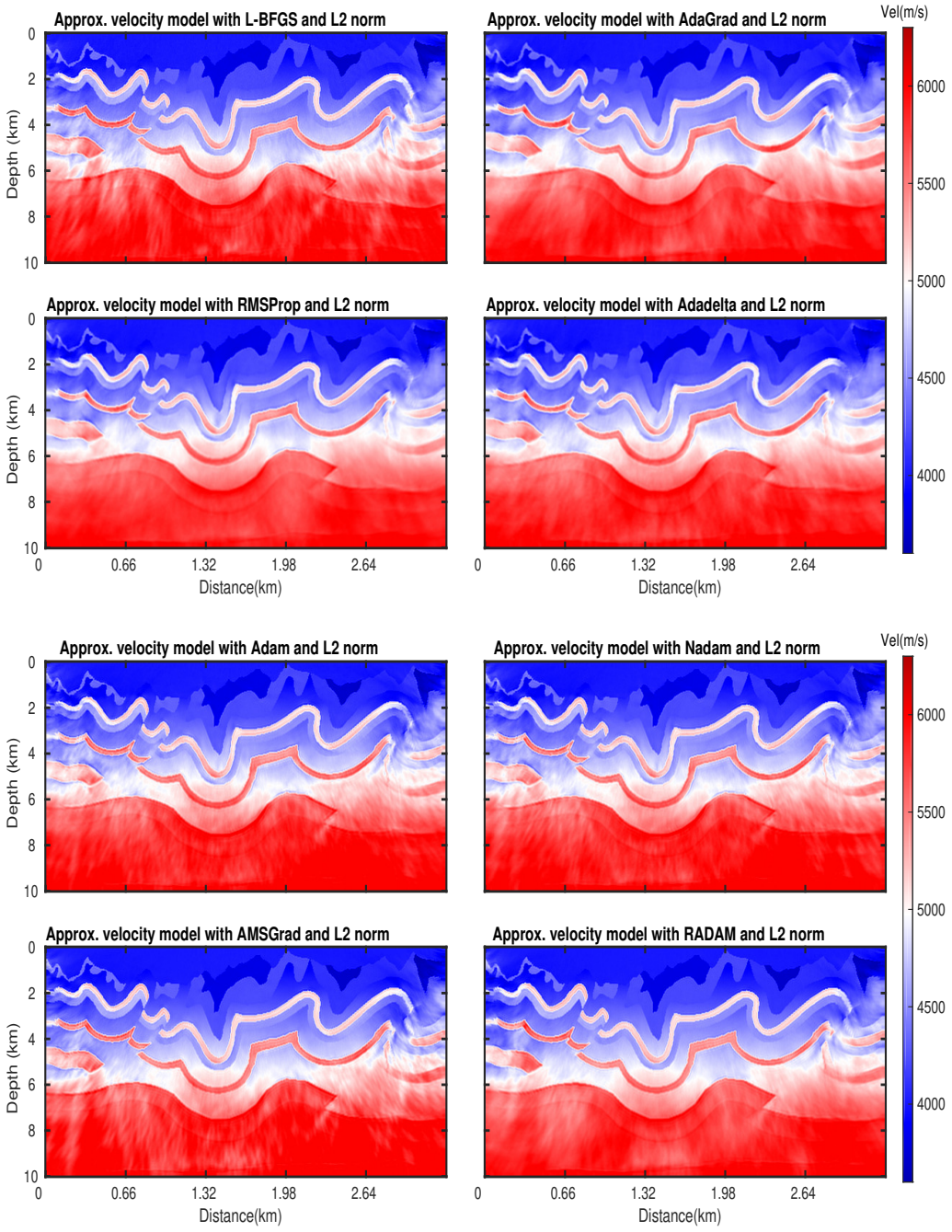

Figura 5.3: Modelos de velocidad finales obtenidos después de la inversión usando cada uno de los métodos de optimización aplicados en FWI basada en la norma L_2.

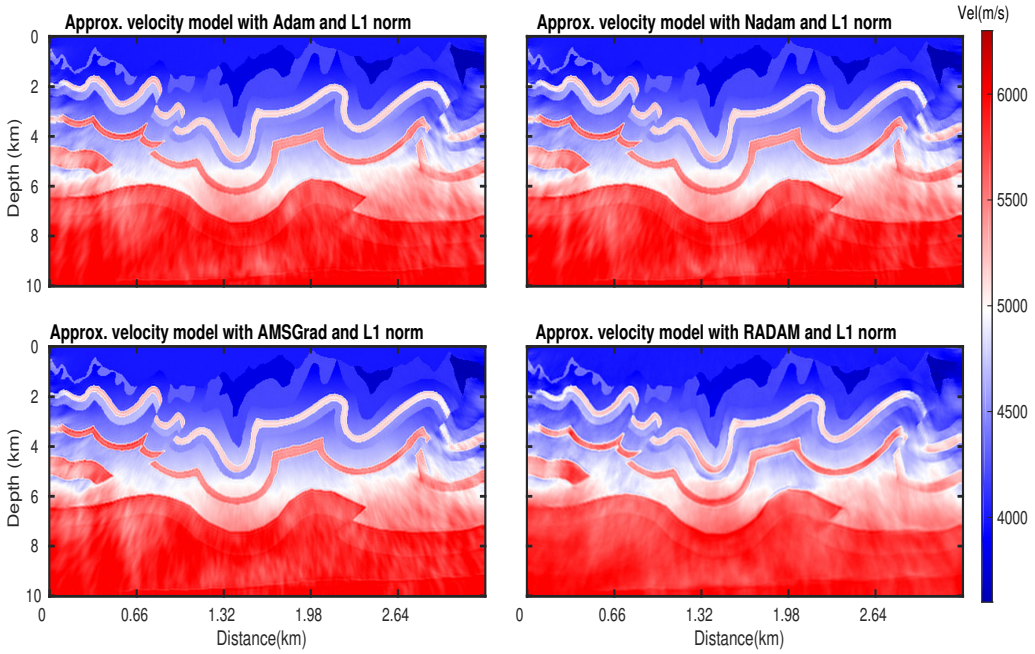

Figura 5.4: Modelos de velocidad finales obtenidos después de la inversión usando los métodos: Adam, Nadam, AMSGrad and RAdam; aplicados en FWI basada en la norma L_1.

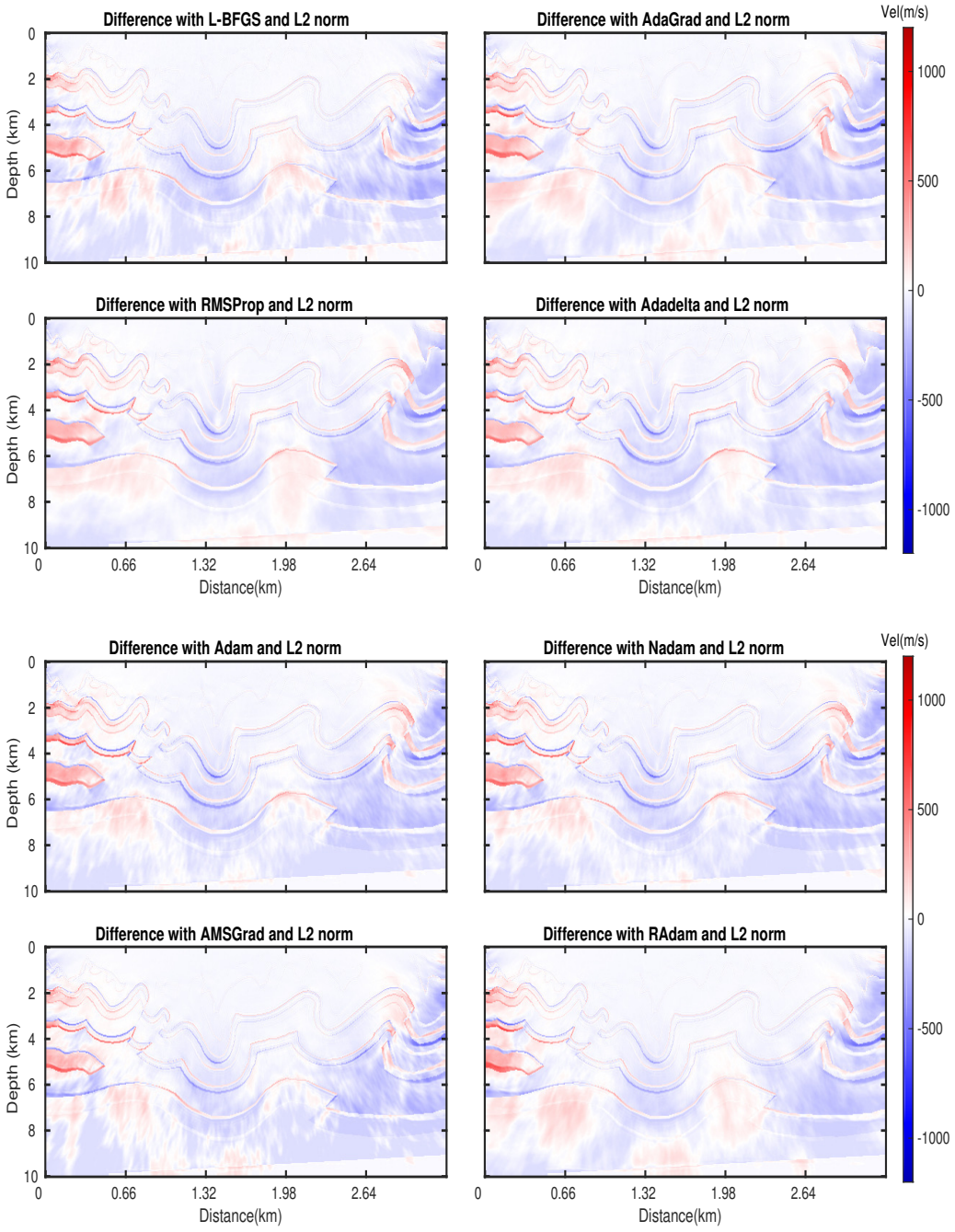

Figura 5.5: Diferencias entre los modelos de velocidades, real y aproximados, usando cada uno de los métodos de optimización aplicados en FWI basada en la norma L_2.

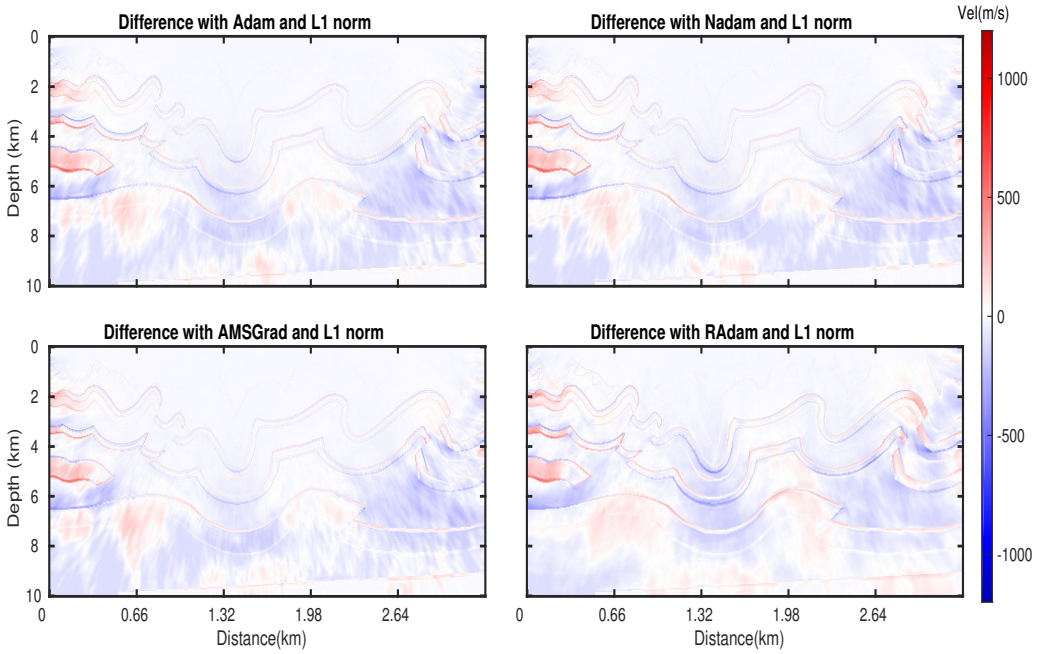

Figura 5.6: Diferencias entre los modelos de velocidades, real y aproximados, usando los métodos: Adam, Nadam, AMSGrad y RAdam; aplicados en FWI basada en la norma L_1.

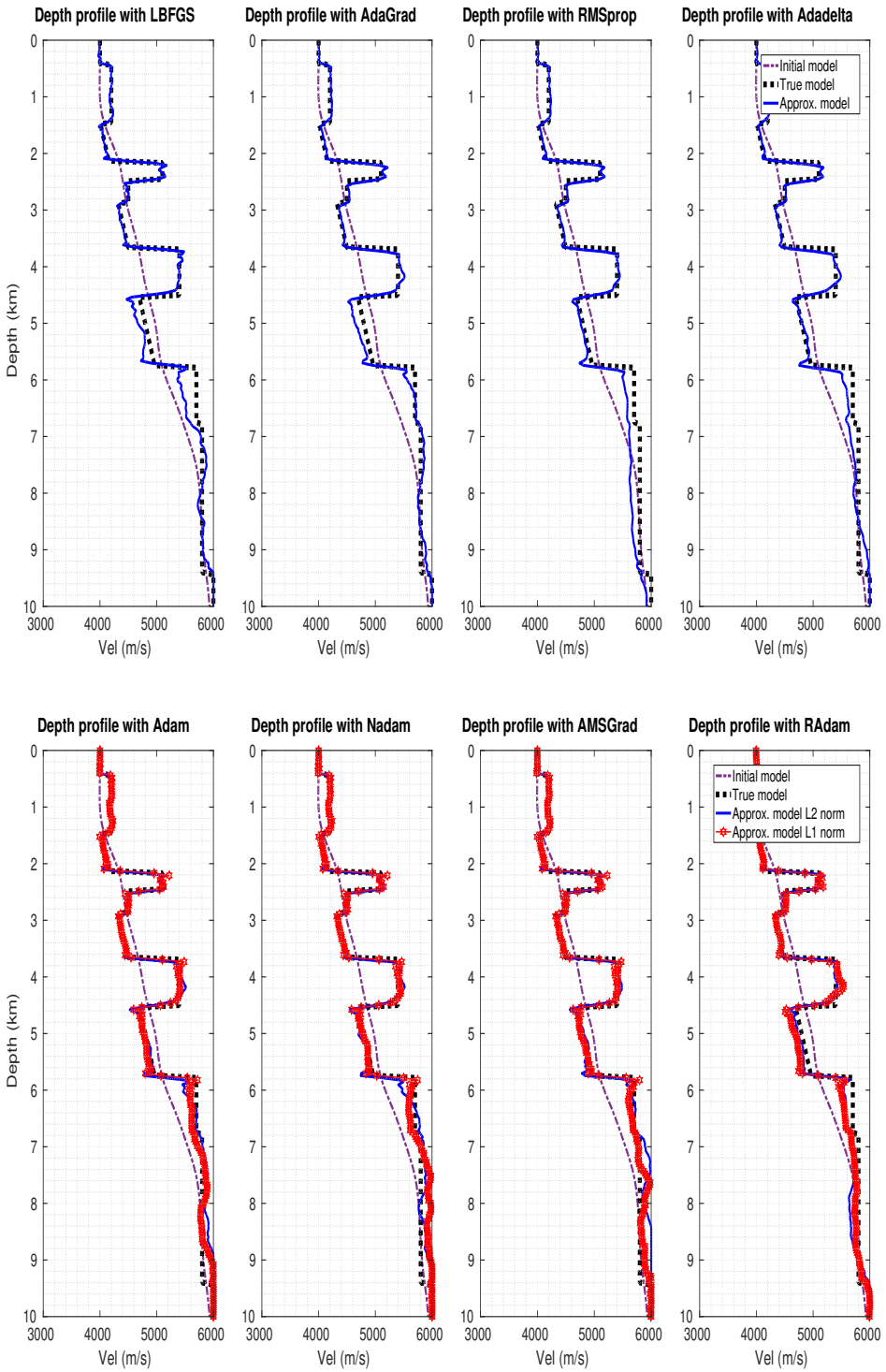

Figura 5.7: Comparación de los perfiles de profundidad, real y aproximados, en $x = 820$m; obtenidos al aplicar cada uno de los métodos de optimización en FWI basada en la norma L_2. Para la FWI basada en la norma L_1, sólo mostramos los resultados obtenidos con los métodos: Adam, Nadam, AMSGrad y RAdam.

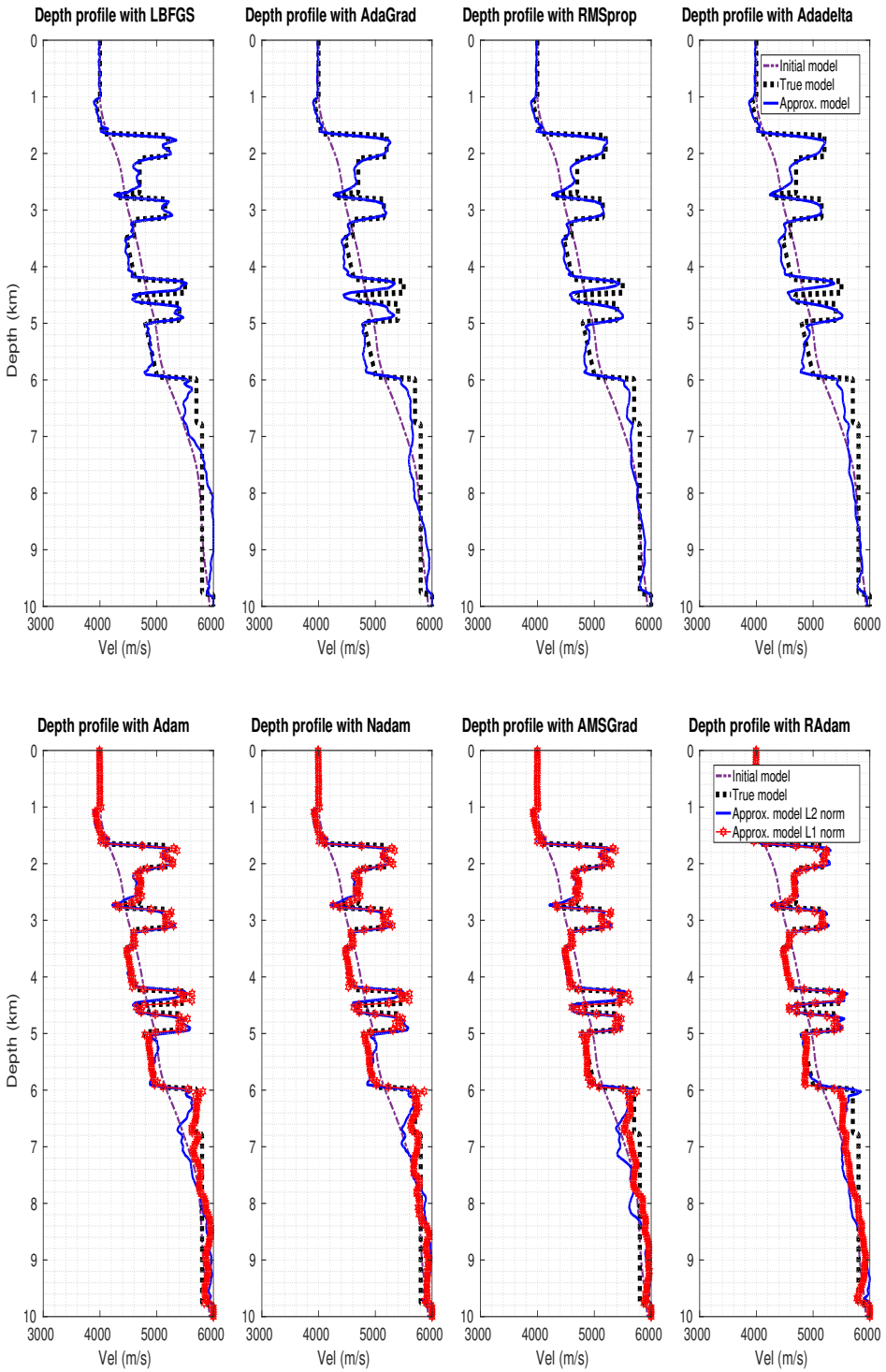

Figura 5.8: Comparación de los perfiles de profundidad, real y aproximados, en $x = 2$km; obtenidos al aplicar cada uno de los métodos de optimización en FWI basada en la norma L_2. Para la FWI basada en la norma L_1, sólo mostramos los resultados obtenidos con los métodos: Adam, Nadam, AMSGrad y RAdam.

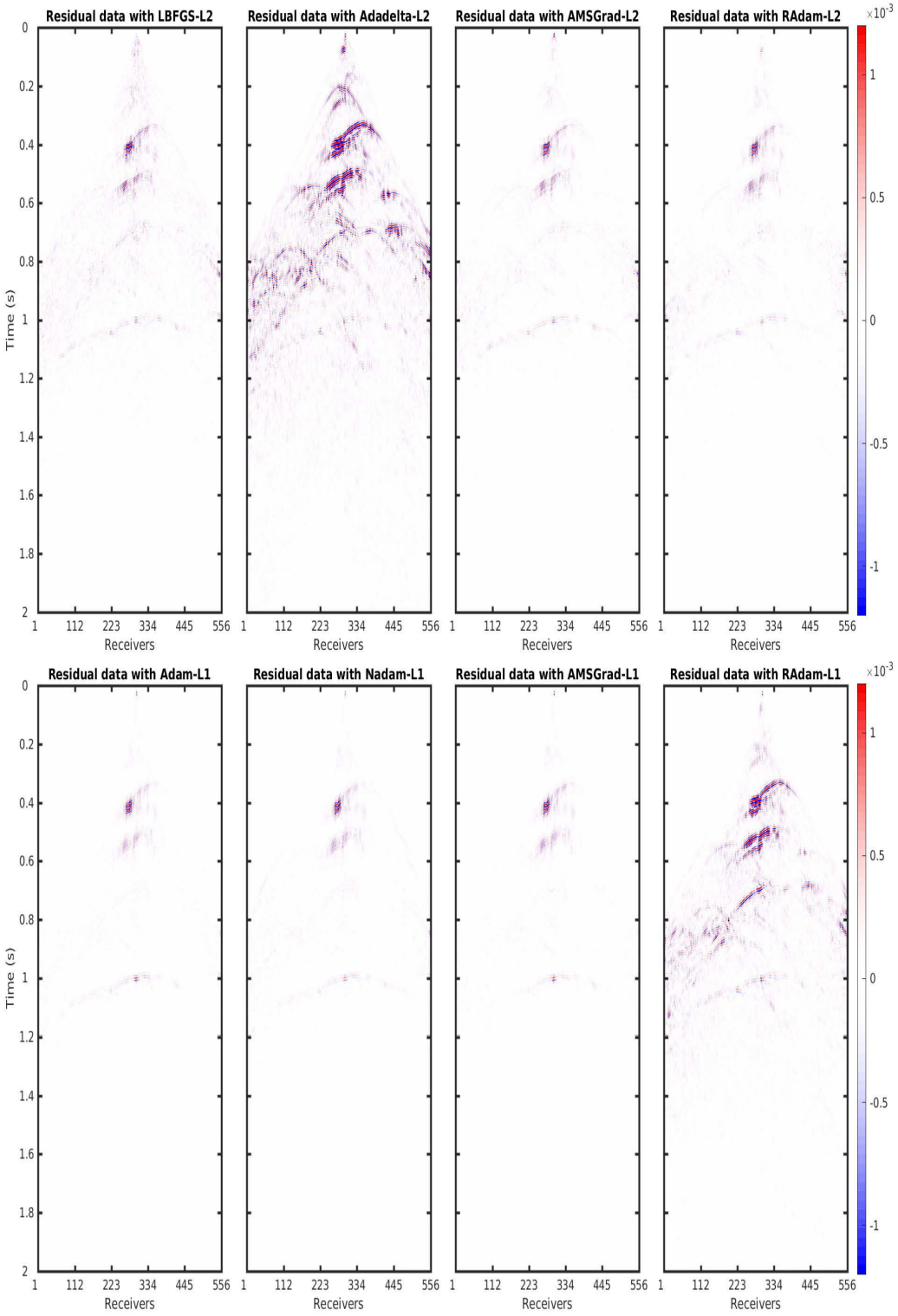

Figura 5.9: Resuduales (diferencia entre sismogramas observados y aproximados correspondientes a una fuente ubicada a la mitad de la línea de receptores) usando los modelos de velocidades finales obtenidos con L-BFGS y algunos métodos AGO, usando FWI basada en las normas L_2 y L_1.

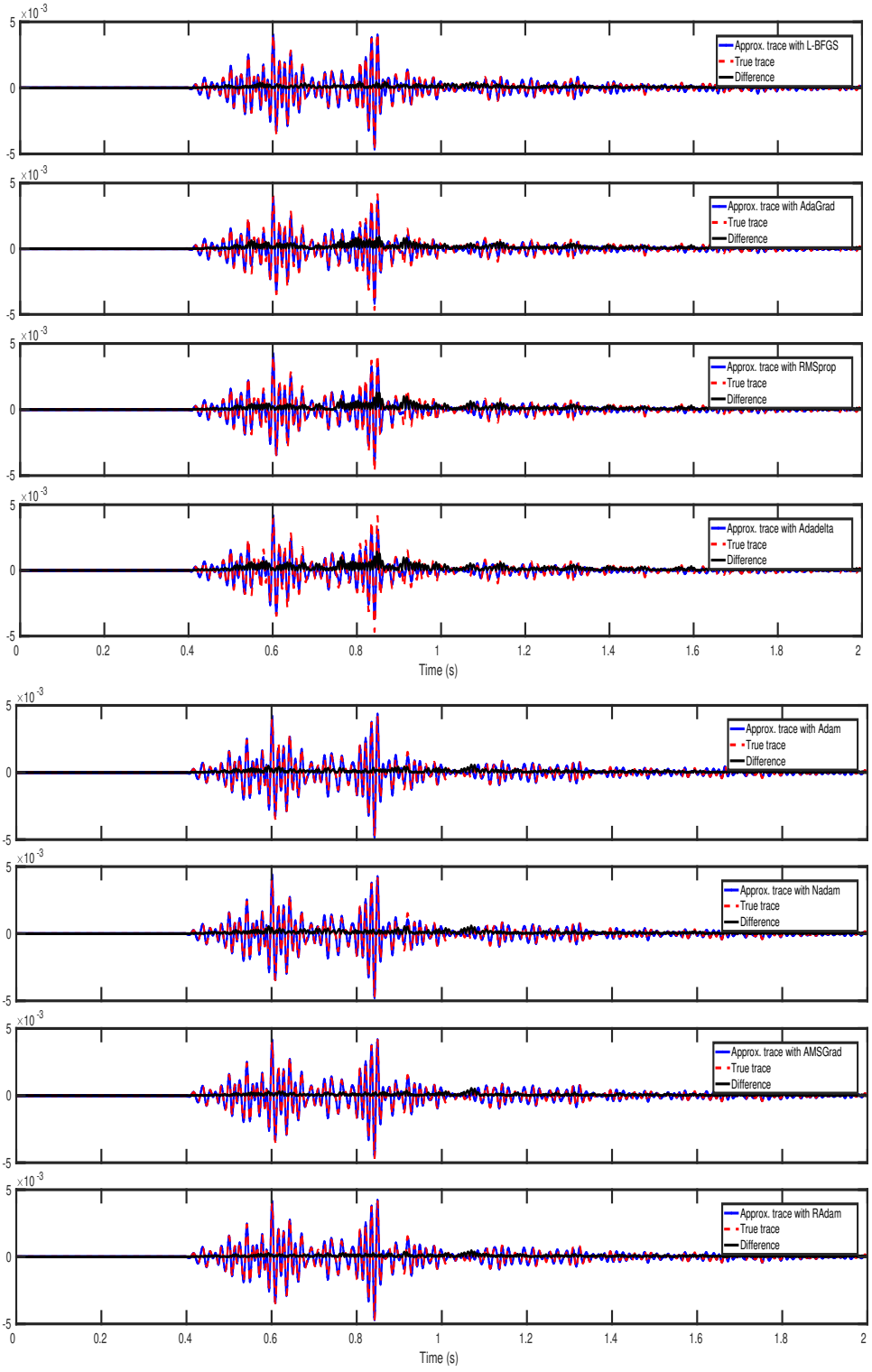

Figura 5.10: Comparación de trazas reales y aproximadas correspondientes al receptor No.140, usando cada uno de los métodos de optimización en FWI basada en la norma L_2.

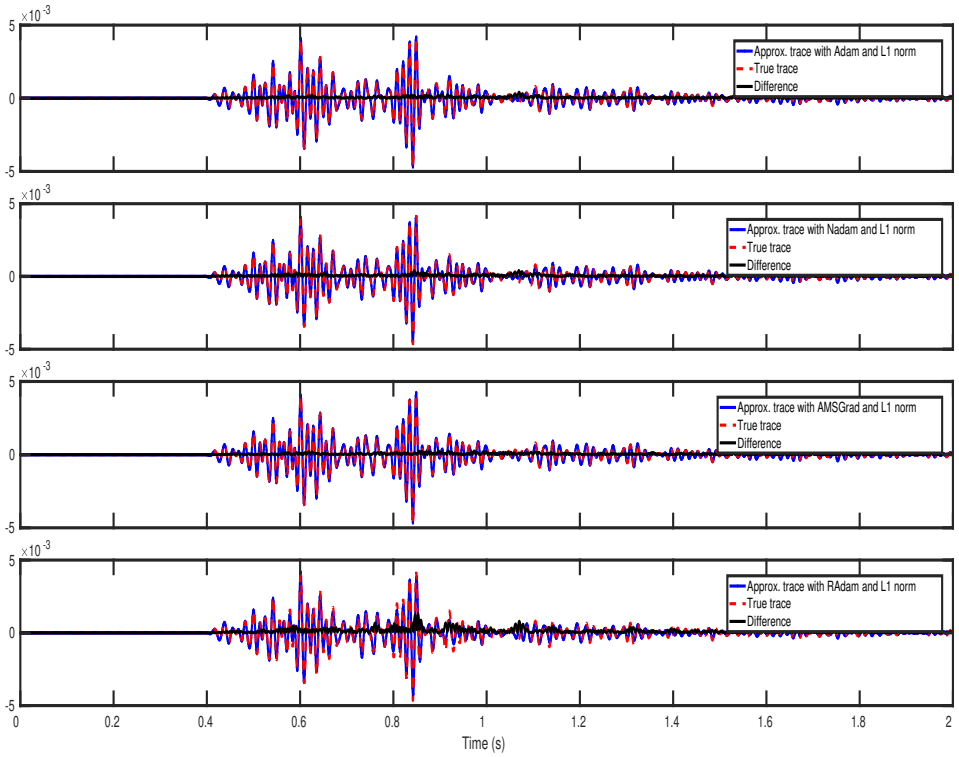

Figura 5.11: Comparación de trazas reales y aproximadas correspondientes al receptor No.140, usando los métodos: Adam, Nadam, AMSGrad y RAdam en FWI basada en la norma L_1.

5.0.3. Inversión del modelo de velocidades: Marmousi

Ahora aplicamos los métodos AGO a una muestra del modelo de Marmousi 2 (Martin et al., 2006) de tamaño $nz \times nx = 257 \times 522$ (sin incluir fronteras C-PML) que se muestra en la figura 5.12 (izquierda). De la misma forma como en la sección anterior consideramos un modelo inicial de capas planas que se muestra en la figura 5.12 (derecha). El campo de onda acústico es registrado por 522 receptores ubicados a 10m bajo la superficie libre, durante un tiempo de $T = 7{,}5$s lo cual requiere 7500 pasos de tiempo. Aunque usamos un incremento de $dx = 10$, las escalas en las figuras 5.12, 5.14, 5.15, 5.16 y 5.17, son las mismas que se muestran en Martin et al. (2006) por lo que el modelo tiene una longitud de 10km y una profundidad de 3,5km.

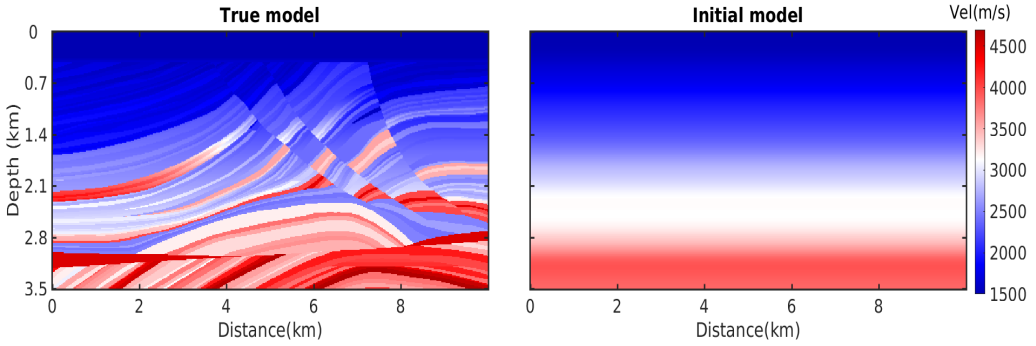

Figura 5.12: Modelo de velocidades: Marmousi 2 (izquierda); Modelo de velocidades inicial (derecha).

Durante el proceso de multiscaling usamos las frecuencias: 1.5Hz, 3Hz, 5.25Hz, 7.5Hz, 9Hz, 12Hz y 15Hz. Siguiendo el mismo orden como en la sección anterior, las figuras 5.14 y 5.15 muestran los modelos finales generados con la FWI basada en las normas L_2 y L_1, respectivamente. La diferencia entre el modelo real y algunos modelos de velocidades obtenidos con la FWI (basada en las normas L_2 y L_1) se muestran en las figuras 5.16 y 5.17, respectivamente. La curva de error en cada iteración de FWI (usando escala logarítmica) obtenida con la ecuación (5.1), se muestra en la figura 5.13. En

las figuras 5.18 y 5.19 se muestran los perfiles de profundidad a $x = 2{,}5$km y a $x = 5$km, respectivamente. Como en la sección anterior sólo usamos algunos métodos AGO para mostrar la comparación entre el residual de datos con la FWI basada en la norma L_2 (con L-BFGS, Adadelta, AMSGrad y RAdam) y con la FWI basada en la norma L_1 (con Adam, Nadam, AMSGrad y RAdam), correspondientes a una fuente ubicada en el punto medio de la línea de receptores; dichos residuales se muestran en la figura 5.20. Para visualizar la precisión en las formas de onda, comparamos las trazas reales con las trazas aproximadas (correspondientes al receptor No. 130) generadas con los modelos obtenidos de la FWI basada en la norma L_2 (figura 5.21) y basada en la norma L_1 (figura 5.22).

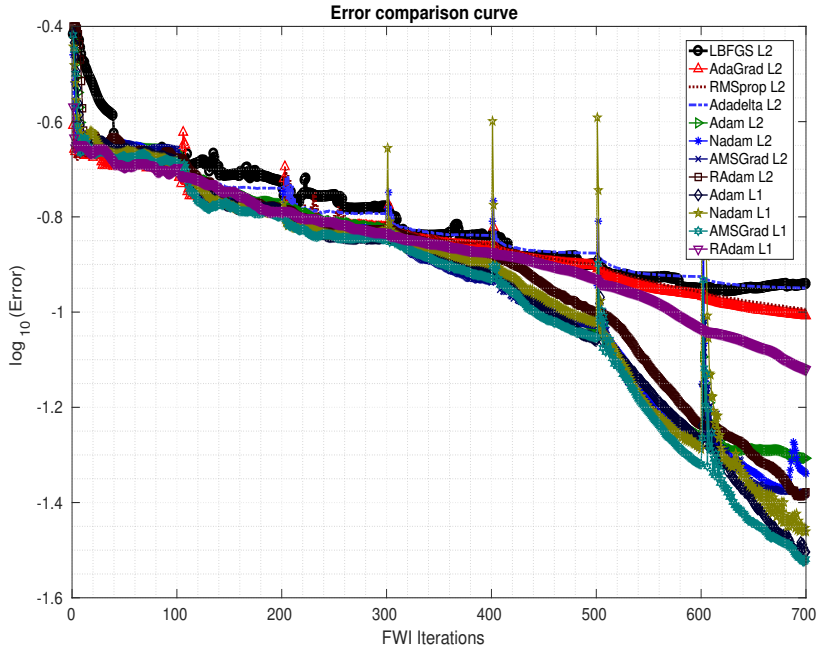

Figura 5.13: Curva de error en cada iteración de FWI (usando escala logarítmica) para cada uno de los métodos de optimización aplicados en la FWI basada en las normas L_2 y L_1.

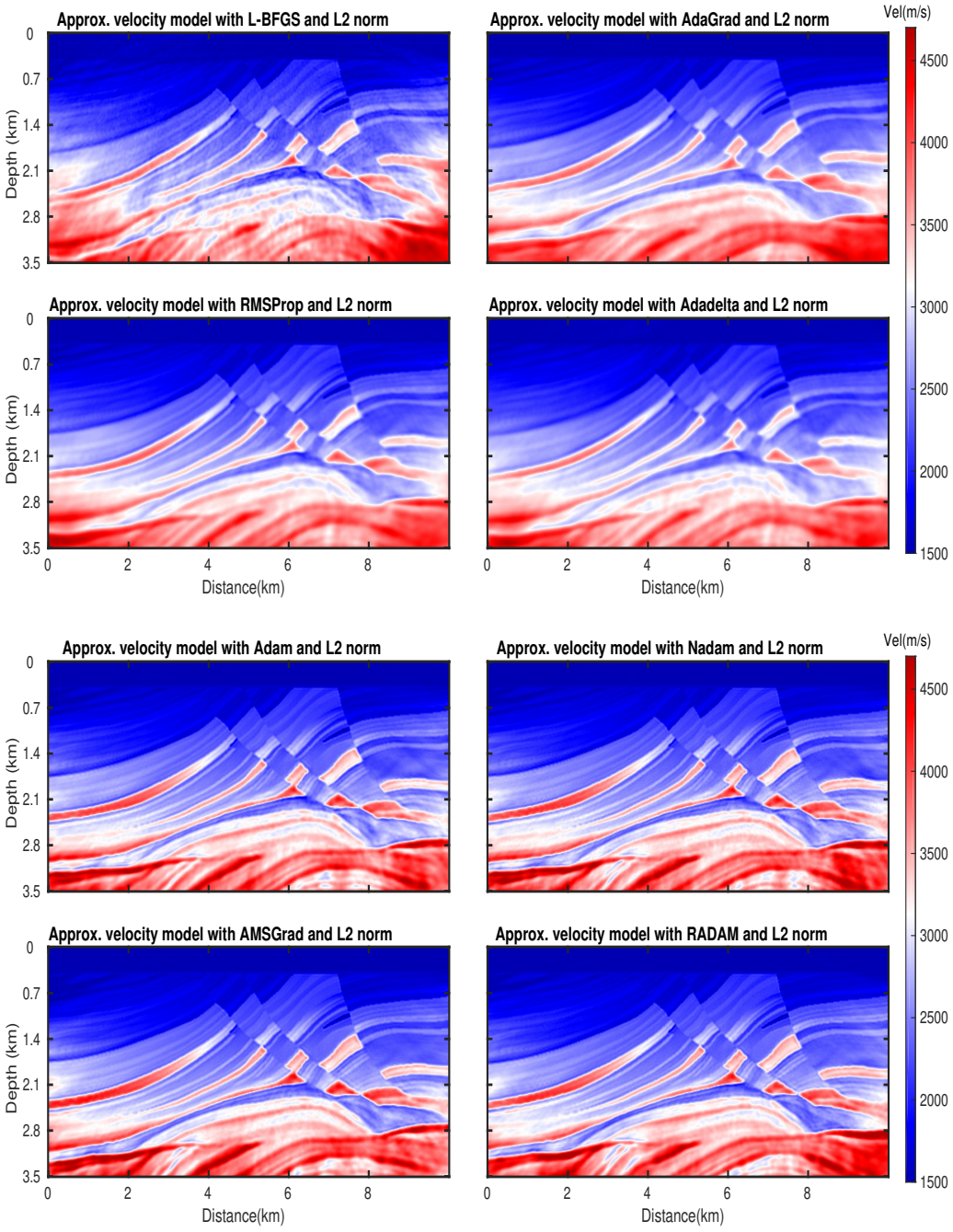

Figura 5.14: Modelos de velocidad finales obtenidos después de la inversión usando cada uno de los métodos de optimización aplicados en FWI basada en la norma L_2.

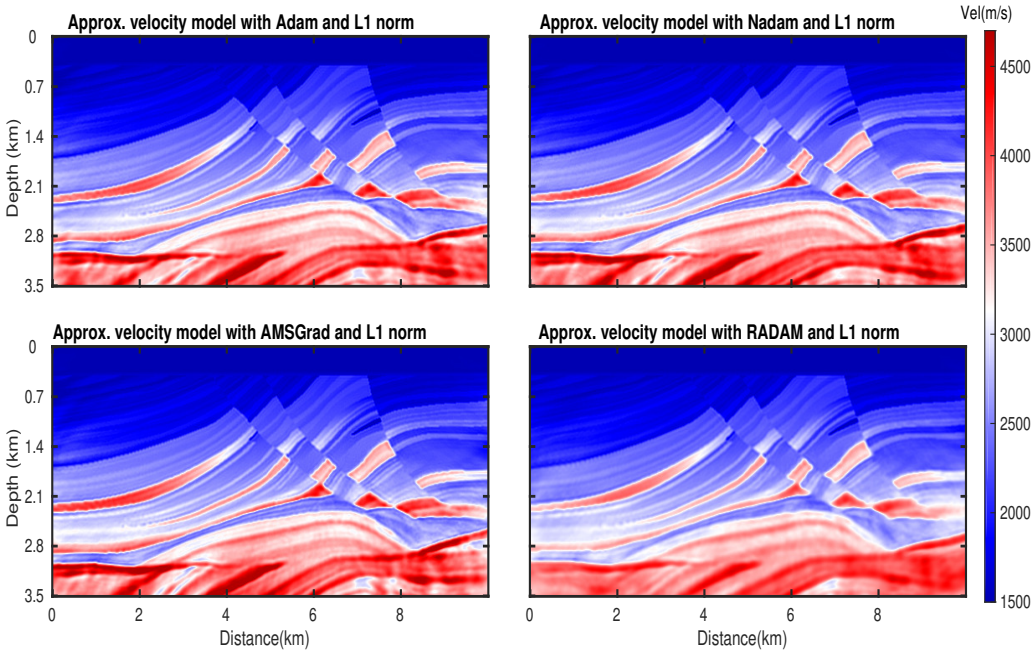

Figura 5.15: Modelos de velocidad finales obtenidos después de la inversión usando los métodos: Adam, Nadam, AMSGrad and RAdam; aplicados en FWI basada en la norma L_1.

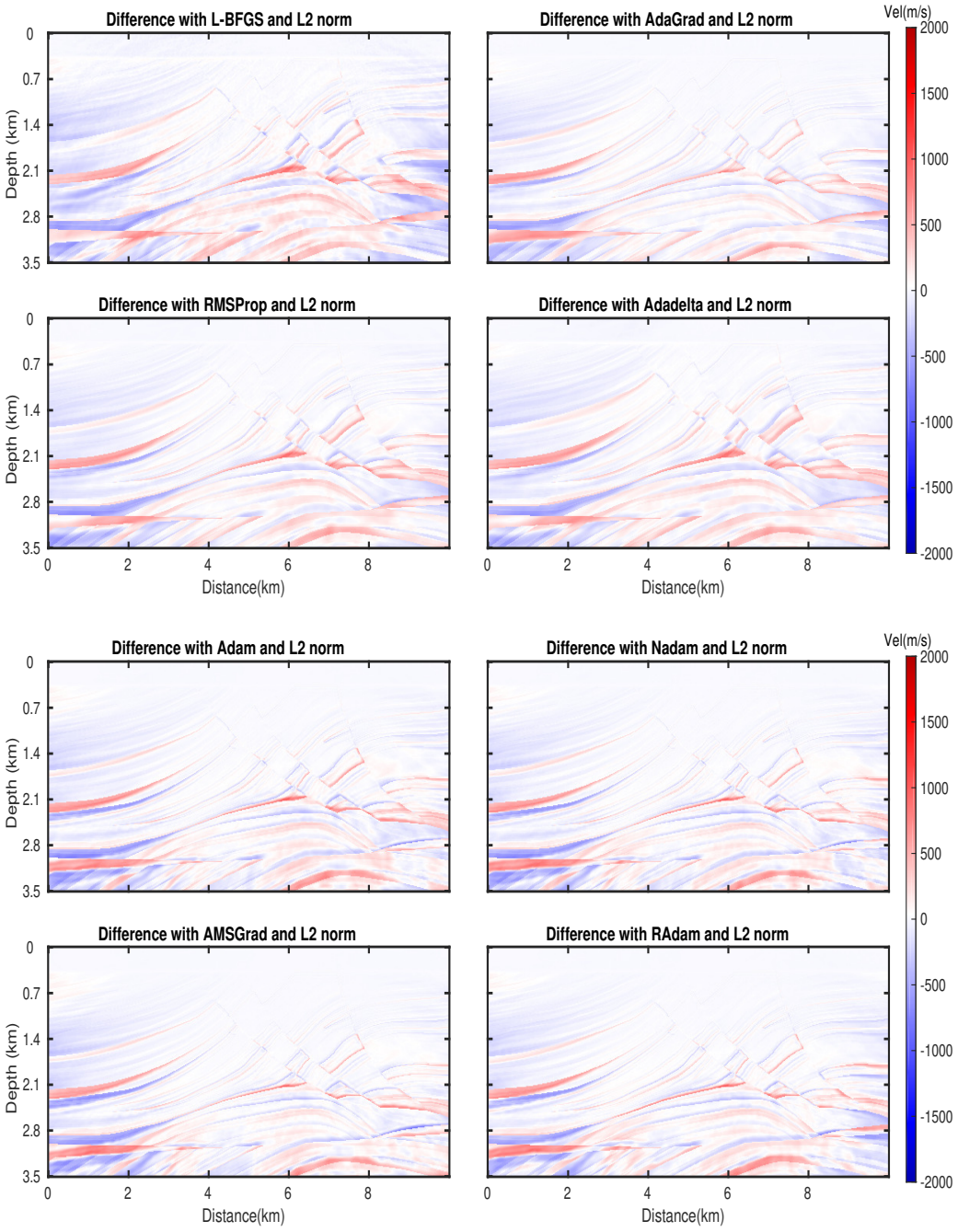

Figura 5.16: Diferencias entre los modelos de velocidades, real y aproximados, usando cada uno de los métodos de optimización aplicados en FWI basada en la norma L_2.

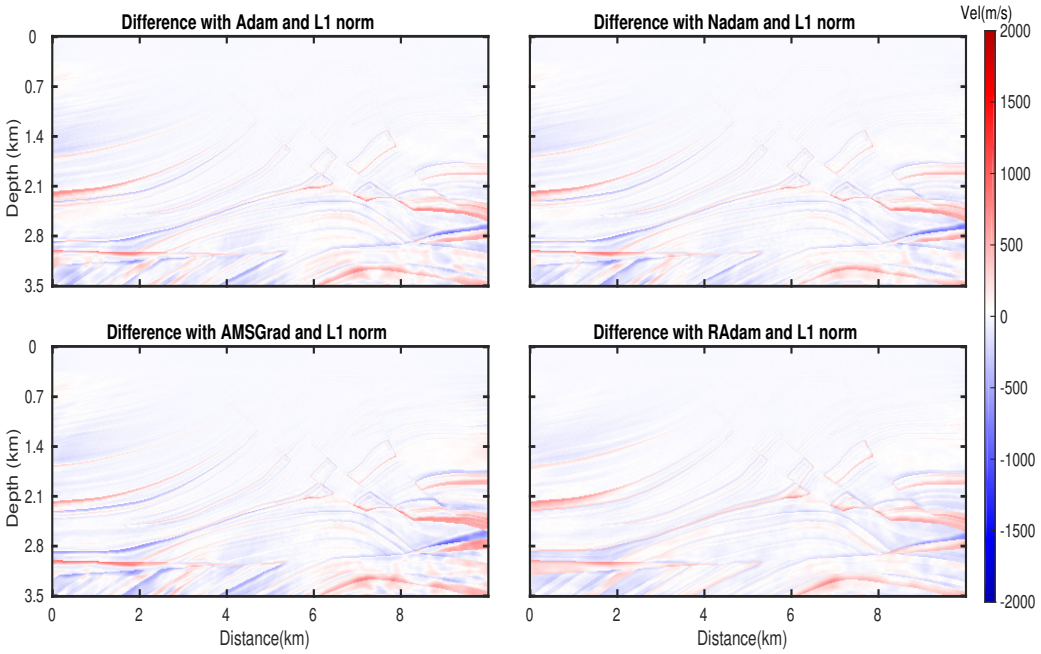

Figura 5.17: Diferencias entre los modelos de velocidades, real y aproximados, usando los métodos: Adam, Nadam, AMSGrad y RAdam; aplicados en FWI basada en la norma L_1.

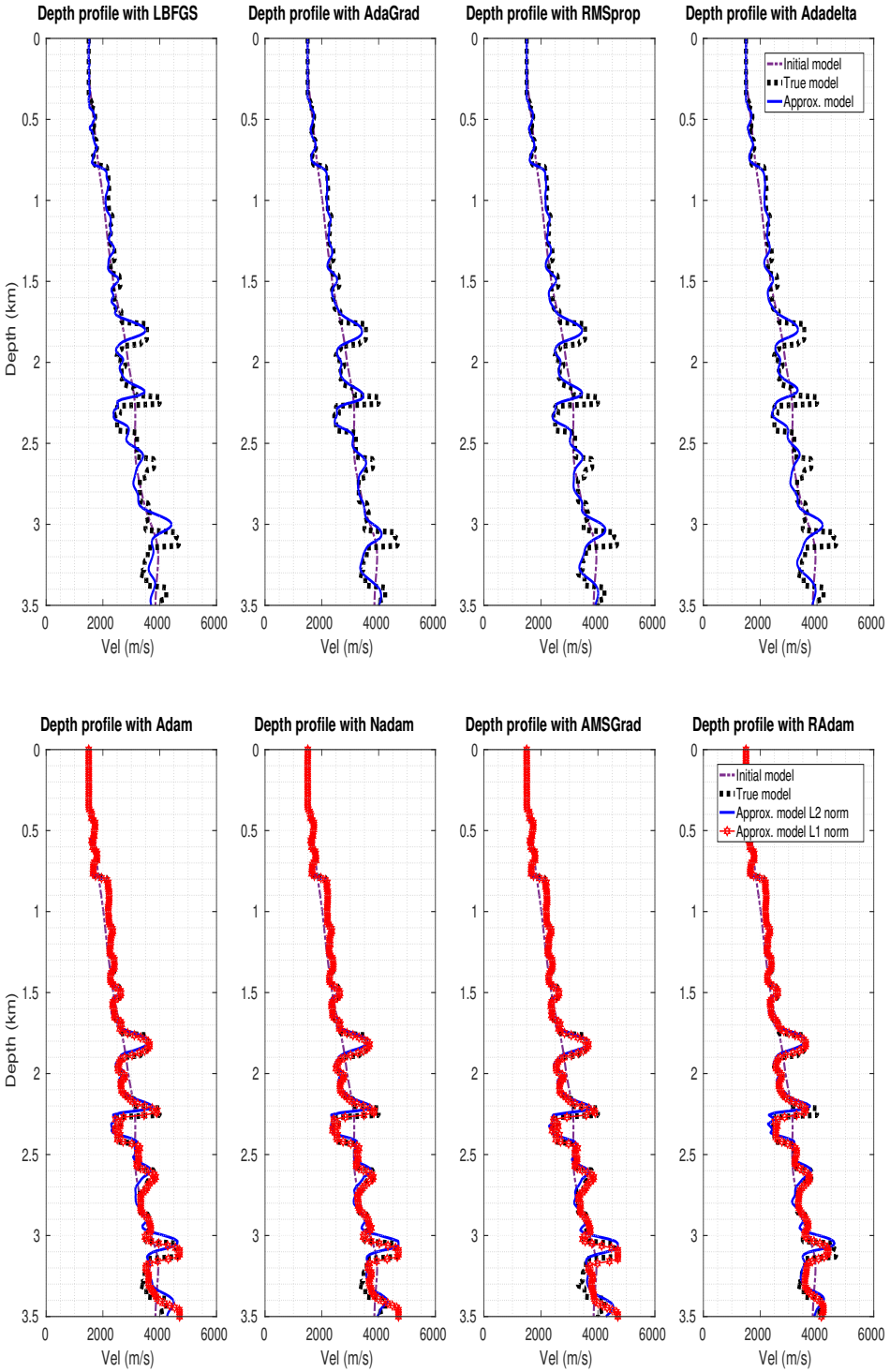

Figura 5.18: Comparación de los perfiles de profundidad, real y aproximados, en $x = 2{,}5$km; obtenidos al aplicar cada uno de los métodos de optimización en FWI basada en la norma L_2. Para la FWI basada en la norma L_1, sólo mostramos los resultados obtenidos con los métodos: Adam, Nadam, AMSGrad y RAdam.

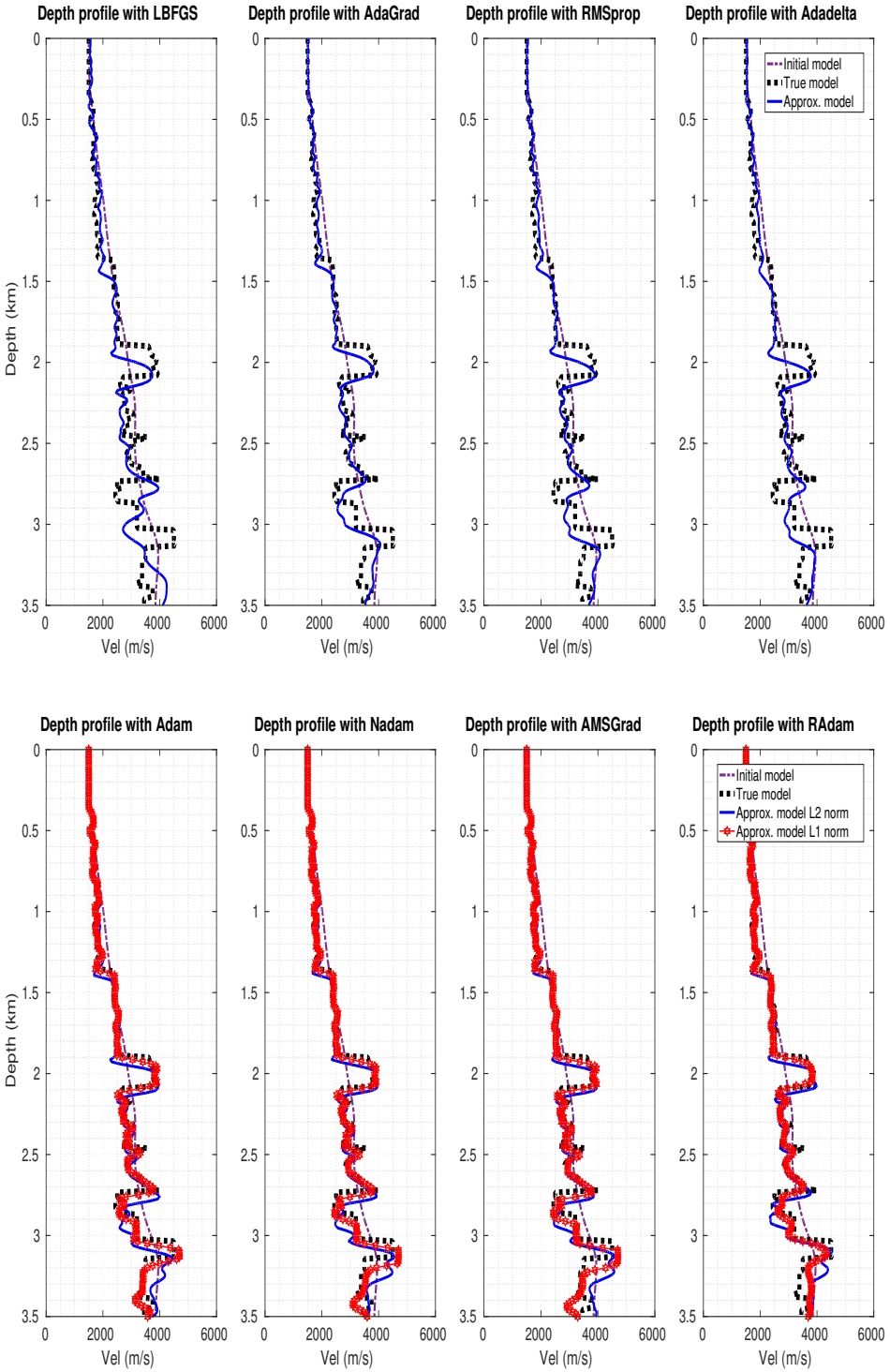

Figura 5.19: Comparación de los perfiles de profundidad, real y aproximados, en $x = 5$km; obtenidos al aplicar cada uno de los métodos de optimización en FWI basada en la norma L_2. Para la FWI basada en la norma L_1, sólo mostramos los resultados obtenidos con los métodos: Adam, Nadam, AMSGrad y RAdam.

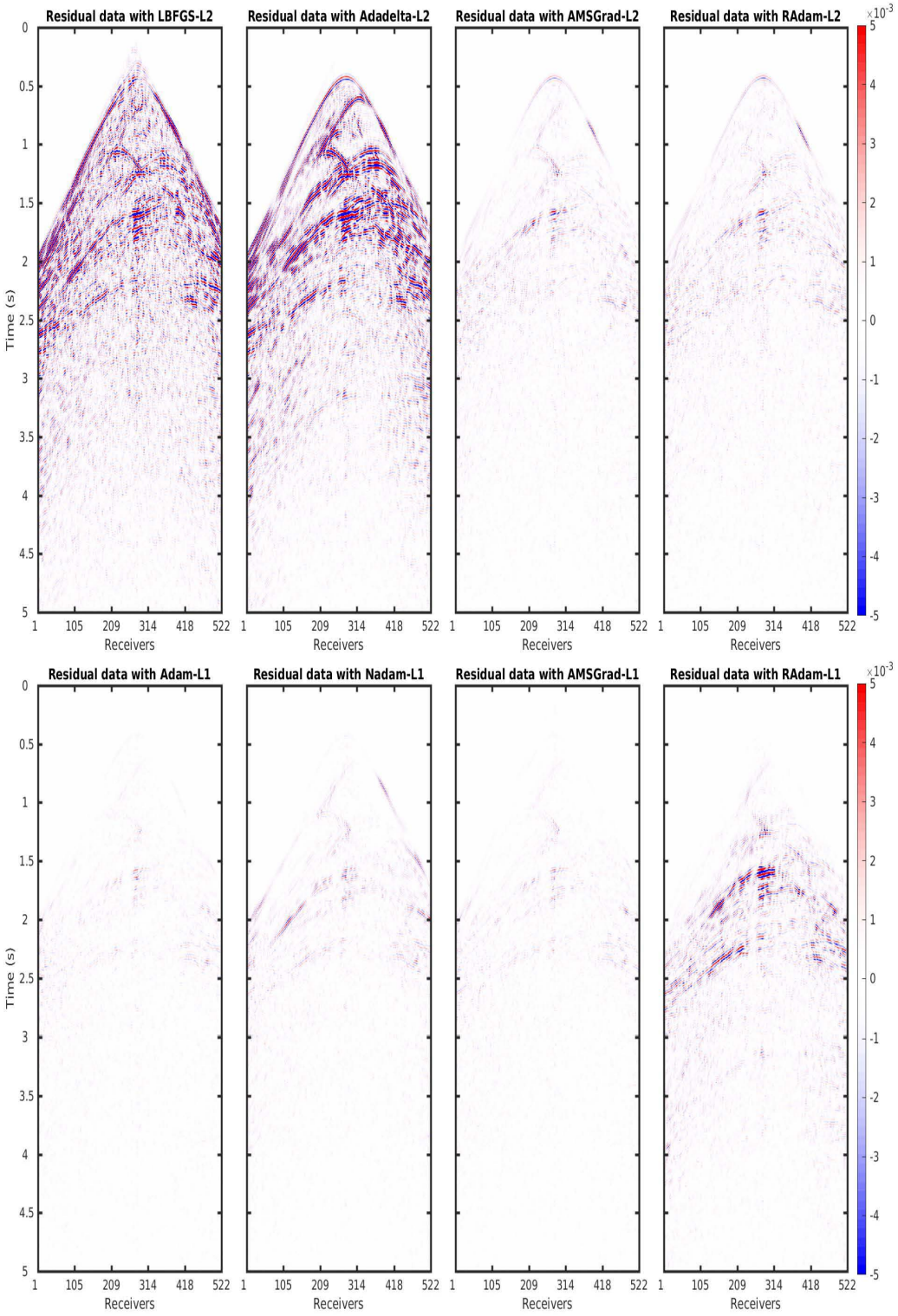

Figura 5.20: Resuduales (diferencia entre sismogramas observados y aproximados correspondientes a una fuente ubicada a la mitad de la línea de receptores) usando los modelos de velocidades finales obtenidos con L-BFGS y algunos métodos AGO, usando FWI basada en las normas L_2 y L_1.

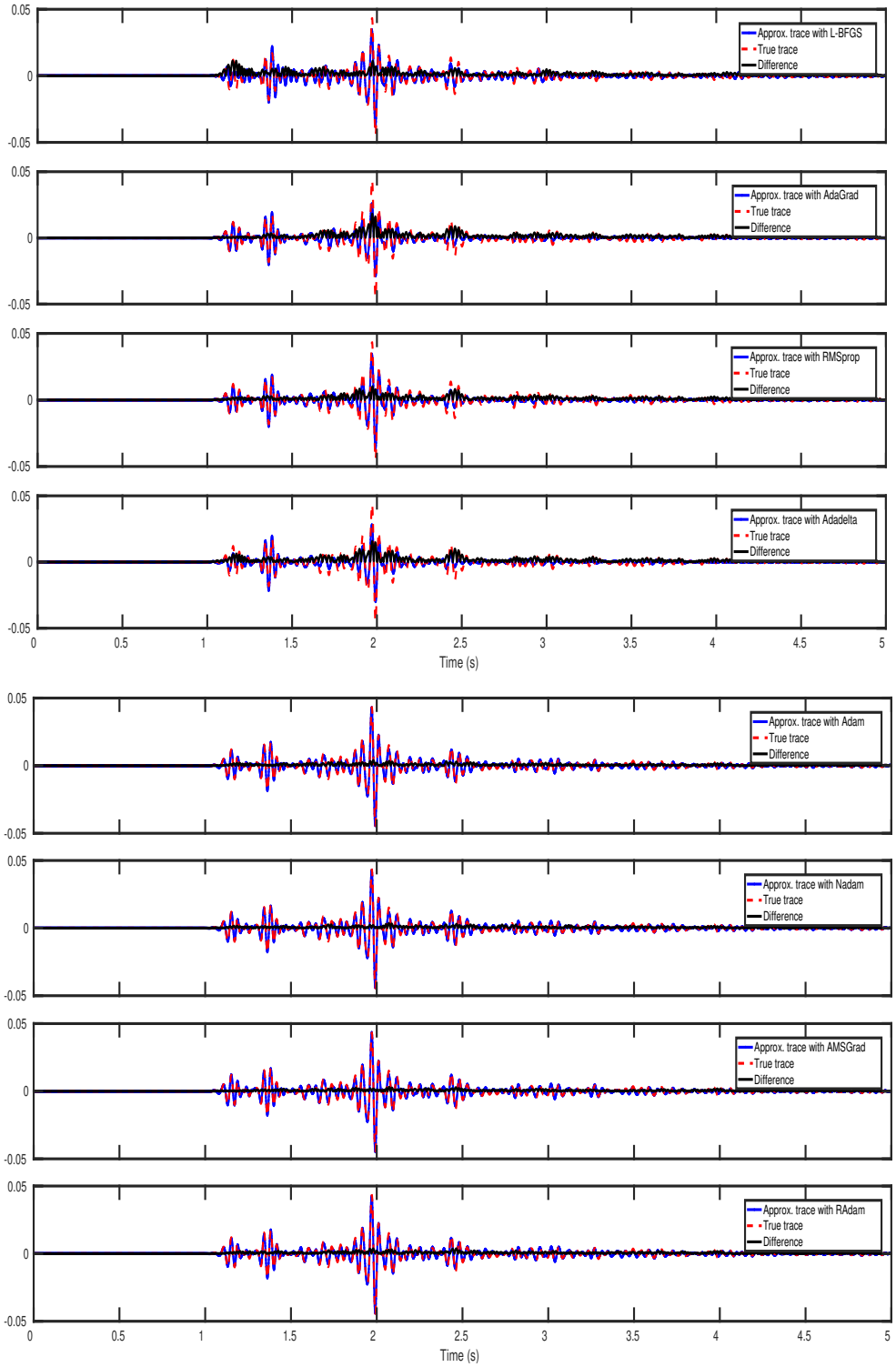

Figura 5.21: Comparación de trazas reales y aproximadas correspondientes al receptor No.130, usando cada uno de los métodos de optimización en FWI basada en la norma L_2.

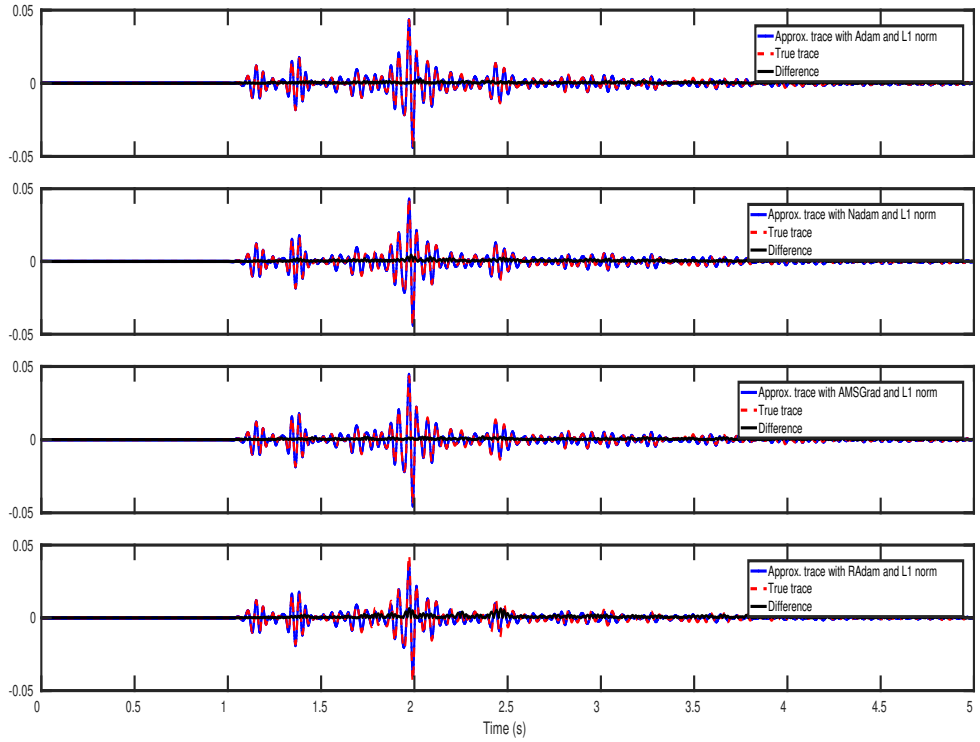

Figura 5.22: Comparación de trazas reales y aproximadas correspondientes al receptor No.130, usando los métodos: Adam, Nadam, AMSGrad y RAdam en FWI basada en la norma L_1.

Capítulo 6

Discusión y Conclusiones

En ambos experimentos numéricos se realizaron 100 iteraciones de FWI por cada una de las siete frecuencias del proceso de multiscaling. En el Cuadro 6.1 mostramos la comparación del desempeño entre el método L-BFGS y los métodos AGO, en donde se mide el tiempo en que se realiza una sola iteración de FWI y el número de propagaciones directas que se realizan en ella. El tiempo de cómputo se midió usando un equipo con procesador Intel Xeon E5 12-Core a 2.7 Ghz. De esos resultados podemos notar que el proceso de FWI usando los métodos AGO es computacionalmente más rápido que al usar el método L-BFGS. Esto ocurre porque en general los métodos de line-search que calculan el step-length para el método L-BFGS difícilmente satisface la condición de Wolfe en el primer intento así que se debe recalcular un nuevo step-length y realizar una nueva propagación directa extra para volver a verificar la condición de Wolfe, lo cual incrementa el costo computacional. Sin embargo, en este caso hemos usado el método ASL (ecuación (5.4)) que ofrece un step-length óptimo, de tal forma que sólo requiere una propagación directa extra en cada iteración de FWI. A diferencia de esto, los métodos AGO con nuestra regla de step-length (ecuación (4.2)) no requieren realizar propagaciones directas adicionales así que evitan el proceso de line-search lo que reduce el costo computacional. Se debe notar que los saltos en las curvas de error (figuras 5.2 y 5.13) se deben a los cambios discontinuos en las frecuencias. Cuando se realiza la FWI con bajas frecuencias se van recuperando estructuras grandes en los modelos de velocidades y conforme van incrememntando las frecuencias se van recuperando las estructuras más finas.

Cuadro 6.1: Tiempo en que tarda una iteración completa de FWI y número de propagaciones directas que se realizan en dicha iteración usando los métodos L-BFGS y AGOs. El tiempo de cómputo se midió usando una máquina con procesador Intel Xeon E5 12-Core a 2.7 GHz.

	Methods / Models	L-BFGS	AGOs
Forward Propagations	Canadian	2	1
	Marmousi	2	1
Time	Canadian	82sec	67sec
	Marmousi	210sec	170sec

De acuerdo a las curvas de error para el modelo Canadiense, mostradas en la figura 5.2, el alcance del método L-BFGS es similar al de los métodos Adam y Nadam, sin embargo, el método AMSGrad es el que más reduce el valor de la función de costo ofreciendo un modelo de velocidades más preciso. Por otro lado, debido a que el modelo de Marmousi tiene una estructura más compleja que el modelo Canadiense, por la evolución de la curva de error mostrada en la figura 5.13, observamos que los métodos L-BFGS y Adadelta reducen el valor de la función de costo de forma más lenta comparado con el resto de los métodos AGO.

En las figuras 5.3 y 5.14, observamos que los modelos de velocidad finales obtenidos con Adam, Nadam, AMSGrad y RAdam muestran una resolución más alta cuando nos restrigimos a la FWI basada en la norma L_2. Sin embargo, mientras algunos métodos AGO (particularmente los métodos: AdaGrad, RMSProp y Adadelta) son incapaces de ofrecer buenos resultados cuando se aplican en FWI basada en la norma L_1, los métodos Adam, Nadam y AMSGrad con la norma L_1 superan la resolución que ofrecen todos los métodos con la norma L_2. Más aún, ofrecen una mejor aproximación de las zonas profundas como se muestra en las figuras 5.4 y 5.15. Estas afirmaciones también se pueden verificar al observar la precisión en los perfiles de profundidad que se muestran en las figuras 5.7, 5.8, 5.18 y 5.19. En las figuras 5.6 y 5.17, se muestra la mejoría que ofrecen los métodos AGO con la FWI basada en la norma L_1, al reducir la diferencia (entre los modelos reales y aproximados) comparados con las diferencias mostradas en las figuras 5.5 y 5.16, respectivamente.

Concluimos de su rápida convergencia; la evolución de las curvas de error

y los modelos de velocidades finales, que los mejores optimizadores para la FWI basada en la norma L_2, son los métodos: AMSGrad y RAdam. El método AMSGrad ofrece alta velocidad de convergencia y el método RAdam ofrece una mejor estabilidad durante el proceso de FWI. Además, los resultados que ofrecen los métodos AMSGrad y RAdam son similares a los que ofrecen los métodos Adam y Nadam. Sin embargo, los mejores resultados se obtienen al aplicar la FWI basada en la norma L_1 combinada con los métodos Adam, Nadam y AMSGrad.

Los resultados que ofrece el método L-BFGS son similares a los que ofrecen algunos métodos AGO pero pueden ser superados por los métodos Adam, Nadam y AMSGrad, además el método L-BFGS resulta computacionalmente más costoso y más difícil de implementar que los métodos AGO. En general, las estructuras profundas son más difíciles de recuperar, sin embargo, los métodos AGO funcionan mejor. Debemos notar que el modelo de Marmousi es más difícil de recuperar que el modelo Canadiense, sin embargo, los métodos AGO ofrecieron resultados aceptables a pesar de que iniciamos el proceso de inversión con un simple modelo de velocidades inicial de capas planas.

En Bollapragada et al. (2018), los autores (entre ellos, Nocedal el autor del método L-BFGS) afirman que el método estándar L-BFGS está basado en aproximaciones de gradientes que no contienen ruido. Esto puede ser una desventaja si se combina con la técnica de fuentes simultáneas dinámicas que proponemos, pues introduce cierta cantidad de ruido en el gradiente (el crosstalk noise). Más aún, L-BFGS no es algoritmo adecuado para tratar con problemas de optimización estocástica (Bollapragada et al., 2018; Adolphs et al., 2019); mientras que los métodos AGO lo son. Debido a que la selección aleatoria de fuentes que se realiza con la técnica que hemos usado en este trabajo, el problema de FWI se convierte en un problema de optimización estocástica (Moghaddam et al., 2013), por lo que L-BFGS no se considera la mejor opción dentro de este marco de trabajo. Esta es la razón del por qué algunos métodos AGO muestran un mejor desempeño, es decir, mejores resultados y convergencia más rápida, que el método L-BFGS.

Los métodos de descenso estocástico tienen la ventaja de evitar la costosa realización de simulaciones directas adicionales que se requieren con los métodos convencionales de line-search. Sin embargo, debemos ser cuidadosos al calibrar el step-length ya que esto podría ser una espada de doble filo, pues el precio que uno debe pagar para evitar el proceso de line-search es la disminución del los step-lengths, lo que eventualmente podría resultar en un número significativamente mayor de iteraciones. Otro aspecto importante de

los métodos de descenso estocástico es que los algoritmos convencionales de step-length no son directamente aplicables debido a que la función de costo es diferente en cada iteración de FWI (comportamineto dinámico) como resultado de la activación aleatoria de las fuentes.

6.1. Conclusiones

Recientemente se ha mostrado que la FWI también se puede abordar con técnicas de *deep learning* mediante redes neuronales recurrentes y redes neuronales convolucionales supervisadas, en donde los optimizadores por elección son algunos métodos AGO, véanse: Richardson (2018); Yang and Ma (2019); Sun et al. (2020). Mientras el enfoque con deep learning realiza la búsqueda en el espacio de parámetros (pesos) de una red neuronal, nuestro enfoque realiza la búsqueda directamente en el espacio de los parámetros físicos en cuestión. En Richardson (2018), el autor muestra que el cálculo del gradiente de la función de costo mediante el método de estado adjunto es equivalente al método de diferenciación automática, por lo que nuestro enfoque es muy similar al proceso de entrenamiento que se realiza en el enfoque con deep learning. Ambos enfoques tienen sus ventajas y desventajas, por ejemplo, para el enfoque con deep learning existen herramientas computacionales como Keras, PyTorch, TensorFlow, etc., que simplifican la construcción de la arquitectura con redes neuronales e implementan el proceso de diferenciación automática, sin embargo, para lograr que una red neuronal logre predecir un modelo de parámetros físicos del subsuelo correspondiente a cualquier conjunto de sismogramas, para su entrenamiento se requeriría una gran cantidad de datos geofísicos correspondientes a distintos campos de exploración, mientras que nuestro enfoque permite trabajar con la información disponible correspondiente a un solo campo de exploración.

Hemos estudiado los efectos de aplicar algunos métodos AGO, combinados con una técnica de fuentes simultáneas dinámicas al proceso de FWI (con multiscaling) basada en las normas L_1 y L_2. Esto nos llevó a proponer una nueva fórmula de step-length en la ecuación (4.2) que considera explícitamente las frecuencias usadas en el multiscaling y permite evitar costosos procesos de line-search cuando se aplica a los métodos AGO. Esto ofrece una mejora del costo computacional en FWI y en algunos casos obteniendo mejores resultados que los proporcionados por el método L-BFGS. Esto supera la aplicación de métodos cuasi-Newton comúnmente utilizados en FWI (los cuales dependen de un proceso de line-search). Adicionalmente, hemos reducido el efecto del ruido corsstalk noise aplicando el método de fuentes simultáneas

dinámicas la cual es una combinación de las técnicas: random-in-subgroup shot sub-sampling, desfazamientos temporales aleatorios y polaridades aleatorias. La combinación de estas estrategias en un solo flujo de trabajo da como resultado una aceleración del proceso de FWI y llevando a una inversión de parámetros físicos de alta resolución realizando solo una propagación directa por cada iteración de FWI. Concluimos que los mejores resultados se obtienen cuando se aplican los métodos Adam, Nadam y AMSGrad en la FWI basada en la norma L_1. La teoría mostrada en este trabajo se puede generalizar a los casos multiparamétricos de FWI 3D y con el uso de GPUs, se podría acelerar aún más este proceso, además el uso de paquetes y herramientras de machine learning pueden simplificar la implementación computacional.

El resumen de este trabajo fue aceptado y publicado por la revista *Geophysical Journal International* (Bernal-Romero and Iturrarán-Viveros, 2020) y se muestra en el Apéndice C.

Apéndice A

Deducción del Método de Newton

Sea $f(\vec{x})$ una función doblemente-continuamente diferenciable en una vecindad de un punto $\vec{x}_0$, de tal forma que $f(\vec{x}_0 + \Delta\vec{x})$ se puede expandir alrededor de $\vec{x}_0$ como,

$$f(\vec{x}_0 + \Delta\vec{x}) = f(\vec{x}_0) + \vec{g}^T\Delta\vec{x} + \frac{1}{2}\Delta\vec{x}^T\nabla\nabla^T f(\vec{x}_0)\Delta\vec{x} + o(\| \Delta\vec{x} \|^3), \quad \text{(A.1)}$$

donde $\vec{g}$ representa el gradiente, $\vec{g} = \nabla f(\vec{x}_0)$.

Truncando después del tercer término, obtenemos el modelo de *aproximación cuadrática*,

$$f(\vec{x}_0 + \Delta\vec{x}) \approx f(\vec{x}_0) + \vec{g}^T\Delta\vec{x} + \frac{1}{2}\Delta\vec{x}^T H\Delta\vec{x}, \quad \text{(A.2)}$$

donde H es la matriz Hessiana dada por, $H = \nabla\nabla^T f(\vec{x}^*)$. Se sabe que si $\vec{x}^*$ es un mínimo local o global de f, entonces,

$$\nabla f(\vec{x}^*) = 0 \quad \text{y} \quad \Delta\vec{x}^T H(\vec{x}^*)\Delta\vec{x} > 0, \quad \text{(A.3)}$$

donde la última condición asegura la propiedad de que $H(\vec{x}^*)$ sea definida positiva Kelley (1999), esto quiere decir que los eigenvalores de $H(\vec{x}^*)$ son todos positivos, es decir, f es convexa.

Definiendo $\Delta\vec{x} = \vec{x} - \vec{x}_0$ y tomando la k-ésima componente del gradiente en la ecuación (A.2), obtenemos,

$$g_k = \frac{\partial f(\vec{x})}{\partial x_k} = \frac{\partial f(\vec{x}_0 + \Delta\vec{x})}{\partial x_k} \approx \frac{\partial f(\vec{x}_0)}{\partial x_k} + \sum_{j=1}^{N} \frac{\partial^2 f(\vec{x}_0)}{\partial x_k \partial x_j}\Delta x_j. \quad \text{(A.4)}$$

Si $\vec{x}$ es un punto mínimo de f, entonces $\frac{\partial f(\vec{x})}{\partial x_k} = 0$, por lo que la ecuación (A.4), se reduce a,

$$g_k = -\sum_{j=1}^{N} \frac{\partial^2 f(\vec{x}_0)}{\partial x_k \partial x_j} \Delta x_j \quad \text{ó} \quad \vec{g} = -H\Delta\vec{x}, \tag{A.5}$$

donde $g_k = \frac{\partial f(\vec{x})}{\partial x_k}|_{\vec{x}=\vec{x}_0}$ y $\Delta\vec{x}$ es el vector incógnita.

Resolviendo la ecuación (A.5) para $\Delta\vec{x}$, obtenemos,

$$\Delta\vec{x} = -H^{-1}\vec{g}, \tag{A.6}$$

es decir,

$$\vec{x} - \vec{x}_0 = -[H(\vec{x}_0)]^{-1}\vec{g}(\vec{x}_0), \tag{A.7}$$

de donde,

$$\vec{x} = \vec{x}_0 - [H(\vec{x}_0)]^{-1}\vec{g}(\vec{x}_0), \tag{A.8}$$

La versión iterativa de la ecuación (A.8), se conoce como método de Newton y tiene la forma

$$\vec{x}^{(k+1)} = \vec{x}^{(k)} - \alpha_k[H(\vec{x}^{(k)})]^{-1}\vec{g}(\vec{x}^{(k)}), \tag{A.9}$$

donde α_k es el step-length que debe ser actualizado en cada iteración.

Apéndice B

Adjunto de Operadores Diferenciales

Como vimos en el Capítulo 2.1, para calcular el gradiente de la función de costo usando el método de estado adjunto, es necesario resolver las ecuaciones del modelo adjunto (ecuación (2.13)). En este apartado mostraremos a grandes rasgos qué es el adjunto de un operador (en particular de algunos operadores diferenciales) y cómo intervienen en la ecuación de onda.

Definición B.0.1. *La generalización del producto escalar de dos funciones integrables $f(x)$ y $g(x)$ con $\int_{-\infty}^{\infty} f(x)dx < \infty$ y $\int_{-\infty}^{\infty} g(x)dx < \infty$, se define como:*

$$(f, g) = \int_{-\infty}^{\infty} f(x)g(x)dx. \tag{B.1}$$

Definición B.0.2. *Sean H y G dos espacios de Hilbert (esto es, un espacio normado completo cuya norma es inducida por un producto escalar) cuyo producto escalar se denota por $(\cdot, \cdot)$ y sea $L : H \to G$ un operador lineal. Se llama operador adjunto de L al único operador lineal $L^{\dagger} : G \to H$ que cumple,*

$$(d, Lm)_G = (L^{\dagger}d, m)_H \qquad \forall m \in H, \qquad \forall d \in G. \tag{B.2}$$

Si $H = G$ y $L = L^{\dagger}$, se dice que L es autoadjunto.

Ejemplo B.0.3. *Sea H el espacio de funciones cuadrado integrables, es decir, si $f(x) \in H$, entonces $\int_{\infty}^{\infty} \parallel f(x) \parallel^2 dx < \infty$ y $f(x) \to 0$ cuando $x \to \infty$. Si $\rho^{-1}(x) \in H$ y $L : H \to H$ el operador diferencial dado por,*

$$L(\cdot) = \rho(x)^{-1}\frac{\partial(\cdot)}{\partial x}, \tag{B.3}$$

entonces:

$$L^{\dagger}(\cdot) = -\frac{\partial\{\rho(x)^{-1}(\cdot)\}}{\partial x}. \tag{B.4}$$

En efecto, si $m(x), d(x) \in H$*, aplicando integración por partes en la definición de producto escalar de funciones:*

$$\begin{aligned}(d, Lm) &= \int_{-\infty}^{\infty} d(x)\rho(x)^{-1}\frac{\partial m(x)}{\partial x}dx, \\ &= d^*(x)\rho(x)^{-1}m(x)\mid_{-\infty}^{\infty} - \int_{-\infty}^{\infty}\frac{\partial\{\rho(x)^{-1}d(x)\}}{\partial x}m(x)dx \\ &= \int_{-\infty}^{\infty} -\frac{\partial\{\rho(x)^{-1}d(x)\}}{\partial x}m(x)dx \\ &= (L^{\dagger}d, m). \end{aligned} \tag{B.5}$$

En particular si $\rho(x) = 1$*, se tiene que el adjunto del oprador* $\frac{\partial}{\partial x}$ *es* $-\frac{\partial}{\partial x}$*.*

Ejemplo B.0.4. *Con las mismas hipótesis del ejemplo anterior y aplicando dos veces integración por partes se puede mostrar que el operador diferencial de segundo orden,* $L(\cdot) = \frac{\partial^2(\cdot)}{\partial x^2}$ *es autoadjunto, es decir,* $L^{\dagger} = L$*.*

En general,

$$\left(\frac{\partial^n(\cdot)}{\partial x^n}\right)^{\dagger} = (-1)^n\left(\frac{\partial^n(\cdot)}{\partial x^n}\right).$$

La ecuación del modelo directo que estamos implementando corresponde a la ecuación de onda,

$$\frac{\partial^2 p}{\partial t^2} - \nu^2\left(\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial z^2}\right) = \vec{f}, \tag{B.6}$$

que se basa en operadores diferenciales de segundo orden los cuales son autoadjuntos, entonces para resolver la ecuación del modelo adjunto (ecuación (2.13)), podemos utilizar la misma implementación computacional que resuelve la ecuación del modelo directo usando la fuente adjunta Δd (cuando la función de costo se basa en la norma L_2) o $\frac{\Delta d}{|\Delta d|}$ (cuando la función de costo se basa en la norma L_1).

Apéndice C

Artículo: Accelerating full-waveform inversion through adaptive gradient optimization methods and dynamic simultaneous sources

A continuación se presenta la versión publicada del artículo que incluye los resultados más importantes que hemos mencionado. Dicho artículo se puede entender como el resumen de esta tesis.

Geophysical Journal International

Geophys. J. Int. (2021) **225,** 97–126
Advance Access publication 2020 December 08
GJI General Geophysical Methods

doi: 10.1093/gji/ggaa583

Accelerating full-waveform inversion through adaptive gradient optimization methods and dynamic simultaneous sources

Marcos Bernal-Romero and Ursula Iturrarán-Viveros

Facultad de Ciencias, Universidad Nacional Autónoma de México, Circuito Escolar S/N, Coyoacán C.P. 04510*, México D.F., Mexico.*
E-mail: ursula.iturraran@gmail.com

Accepted 2020 December 4. Received 2020 November 30; in original form 2020 June 4

SUMMARY

Full-waveform inversion (FWI) is a procedure based on the minimization of a misfit (or cost) function applied to the difference between synthetic waveforms and real seismic traces that derives high-resolution velocity models. This is achieved through the iterative adjustment of the velocity model and/or some other physical parameters of the Earth's subsurface, which generally implies large computational effort. In order to minimize this cost function, we explore the use of adaptive gradient optimization (AGO), a variant of stochastic gradient descent (SGD) methods, combining them with a dynamic simultaneous sources strategy that allow us to reduce the computational cost involved in this process. AGO methods are computationally efficient, have little memory requirements and have the capability of adapting the step length according to the optimization process' evolution. Since a precise calibration of the step length is needed to ensure efficiency, the AGOs are well suited for this task because they are able to adapt the step length according to the optimization's development. In this work, we propose a simple nonlinear relationship that allows an adjustment of the step length with respect to the frequencies used in the multiscale FWI, avoiding the line-search strategy's high computational burden. Additionally, the application of this new step length rule into the AGO methods with a dynamic simultaneous sources strategy, allow us to concurrently accelerate and significantly improve the FWI's numerical performance and results. We compare the performance and final results of seven AGO methods, using two different FWI misfit functionals (based on L_1 and L_2 norms) applied to estimate the final velocity models of two benchmark acoustic models: the Marmousi and the Canadian overthrust BP velocity models.

Key words: Inverse theory; Neural networks, fuzzy logic; Numerical modelling; Numerical solutions; Waveform inversion; Computational seismology.

1 INTRODUCTION

Full-waveform inversion (FWI) aims at deriving high-resolution velocity models by minimizing the difference between observed (or acquired) and modelled (or synthetic) seismic waveforms, measured with a metric, mostly the L_2-norm. The synthetic data are generated with a forward waveform simulation for a given set of physical parameters ($\rho =$ density and velocities V_p and V_s). The result of the FWI process is a set of high-resolution images, see Tarantola (1984) and Virieux & Operto (2009). From a mathematical's perspective, FWI is a nonlinear inverse problem that can be interpreted as a local optimization (Tarantola 1984; Virieux *et al.* 2014) or global optimization process (Jin & Madariaga 1993; Sambridge & Mosegaard 2002; Datta & Sen 2016). Since global optimization FWI require thousands of iterations, it is more common in practice to use local optimization techniques, see Schuster (2017).

Commonly used globalization strategies for iterative optimization methods can be classified into two different kinds: line search methods and trust region methods (Yuan 1999; Nocedal & Writght 1999; Conn *et al.* 2000). Some of the most attractive attractive methods to compute search directions used in FWI are: the nonlinear conjugate-gradient (NLCG) method; the quasi-Newton (QN) methods such as the Limited-memory Broyden–Fletcher–Goldfarb–Shanno algorithm (L-BFGS), or the more robust Truncated Newton (TN, Métivier *et al.* 2017) and the Gauss-Newton (GN) method (Tarantola 2005; Brossier *et al.* 2009; Ma & Hale 2012; Métivier *et al.* 2013). In theory, Newton's method converges using less iterations than the above mentioned. In practice however, due to the fact one needs computing the Hessian matrix's inverse, large-scale problems have a very expensive computational cost. In particular, Newton's method for real FWI applications become forbidden, see, for example, Virieux & Operto (2009) and Schuster (2017).

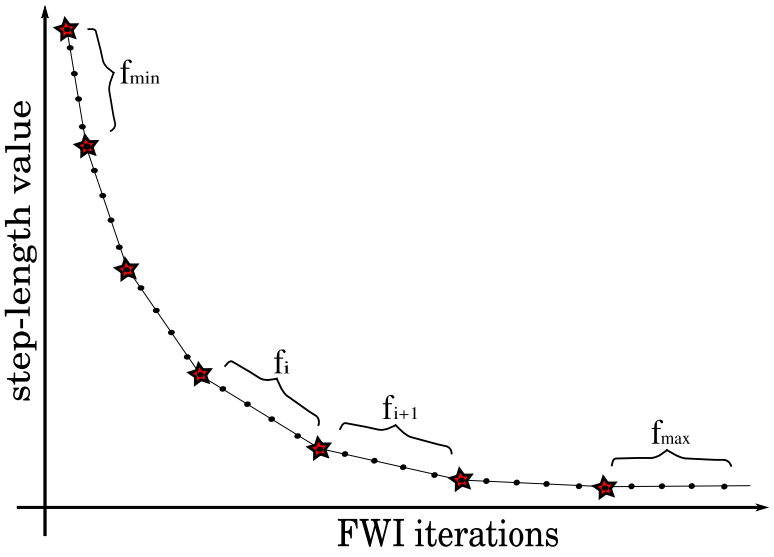

Figure 1. Decay of step-length values at each multiscale batch FWI iteration with increasing frequencies applied to the AGOs.

QN requires a computational approximation of the Hessian matrix's inverse. When these QN methods are applied to FWI, they strongly depend on the step length to guarantee a considerable decrease in the cost function. In order to accomplish this and guarantee convergence to a local minimum, QN methods usually require a line-search strategy that allows them to calculate a step length that satisfies the Wolfe (1969, 1971) conditions. Even though the L-BFGS method (one of the most popular QN methods used in FWI) provides a scaled search direction such that a unitary step length could satisfy the Wolfe conditions (Nocedal & Writght 1999), in FWI (based on line search) at least one extra forward simulation might be required to calculate an analytical step length (ASL), see Ma *et al.* (2019). Since FWI requires hundreds of iterations, mainly when applying FWI in its conventional form: source by source, a single extra forward simulation implies high computation costs. Therefore, in this work, we propose the application of adaptive gradient optimization (AGO) methods (variants of stochastic gradient descent, SGD) as misfit function minimizers in FWI, combined with a dynamic simultaneous sources strategy.

AGOs are computationally efficient; have little memory requirements; have the capability of adapting the step length according to the advancement of the optimization process and they are easy to implement (compared to QN methods), see Kingma & Ba (2015) AND Ruder (2016). Since AGOs do not require any forward modelLing simulation to compute the step length with each iteration, this results in a reduction of the computational cost, thereby improving the computational performance in FWI. Nevertheless, the step length for AGOs needs to be very well calibrated to ensure a decrease in the cost function with each iteration to reach a local minimum. Part of the contribution of this work is to show a simple nonlinear relationship between the step length and the frequencies used in the FWI. This allows us to avoid the expensive line searching, sometimes required by some QN methods, simplifying the conventional FWI workflow and drastically reducing computational time, obtaining similar (or in some cases outperforming) results to those offered by the popular L-BFGS method.

Mulder *et al.* (2006) compared the performance of the L-BFGS method and a kind of AGO method (Nesterov method, one of the first AGOs methods, Nesterov 1983) applied to a test FWI problem. The authors conclude that the L-BFGS results can overcome the Nesterov results. However in this paper, we show that when combining the use of dynamic simultaneous sources FWI and new AGOs, the outcome reverts previous results.

An important observation is that some of the AGO methods can be interpreted as first-order trust region methods with ellipsoidal constraints. In practice, this outperforms its spherical counterpart, see Adolphs *et al.* (2019). Recently, variants of the trust-region optimization method, have been applied to FWI based on SGD with dynamic mini-batches, see van Herwaarden *et al.* (2020).

In Richardson (2018), the author proved that the adjoint state method (used in conventional FWI) is equivalent to the reverse-mode automatic differentiation (commonly used in artificial neural networks). FWI can be seen as a recurrent neural network (RNN) or big-data training where recurrent or supervised deep fully convolutional neural networks are applied and some AGO methods are the optimizers by choice, see Sun & Alkhalifah (2019), Yang & Ma (2019) and Sun *et al.* (2020). Considering that in FWI there is a much higher degree of redundancy in the encoded sources than one would usually find in training data of a neural network, the AGO methods should be tailored to FWI. Hence, we apply some of the most recent AGO methods in a conventional FWI workflow (without performing wave propagation like an RNN), where the FWI misfit function is based on the L_1 and L_2 norms, using a dynamic simultaneous sources method and computing the gradient via the adjoint state method, see Plessix (2006).

The use of the conventional FWI workflow enables us to apply the multiscaling technique: (Boonyasiriwat *et al.* 2009). In order to speed up this process, we employ a dynamic simultaneous sources strategy (which is a combination of several popular source-encoding methods) that allow us to reduce the crosstalk noise. In this work, we apply the combination of these FWI techniques with some AGO's methods (using our step-length assignment rule) to two different synthetic cases: the Canadian overthrust BP velocity model, see Gray & Marfurt (1995) and

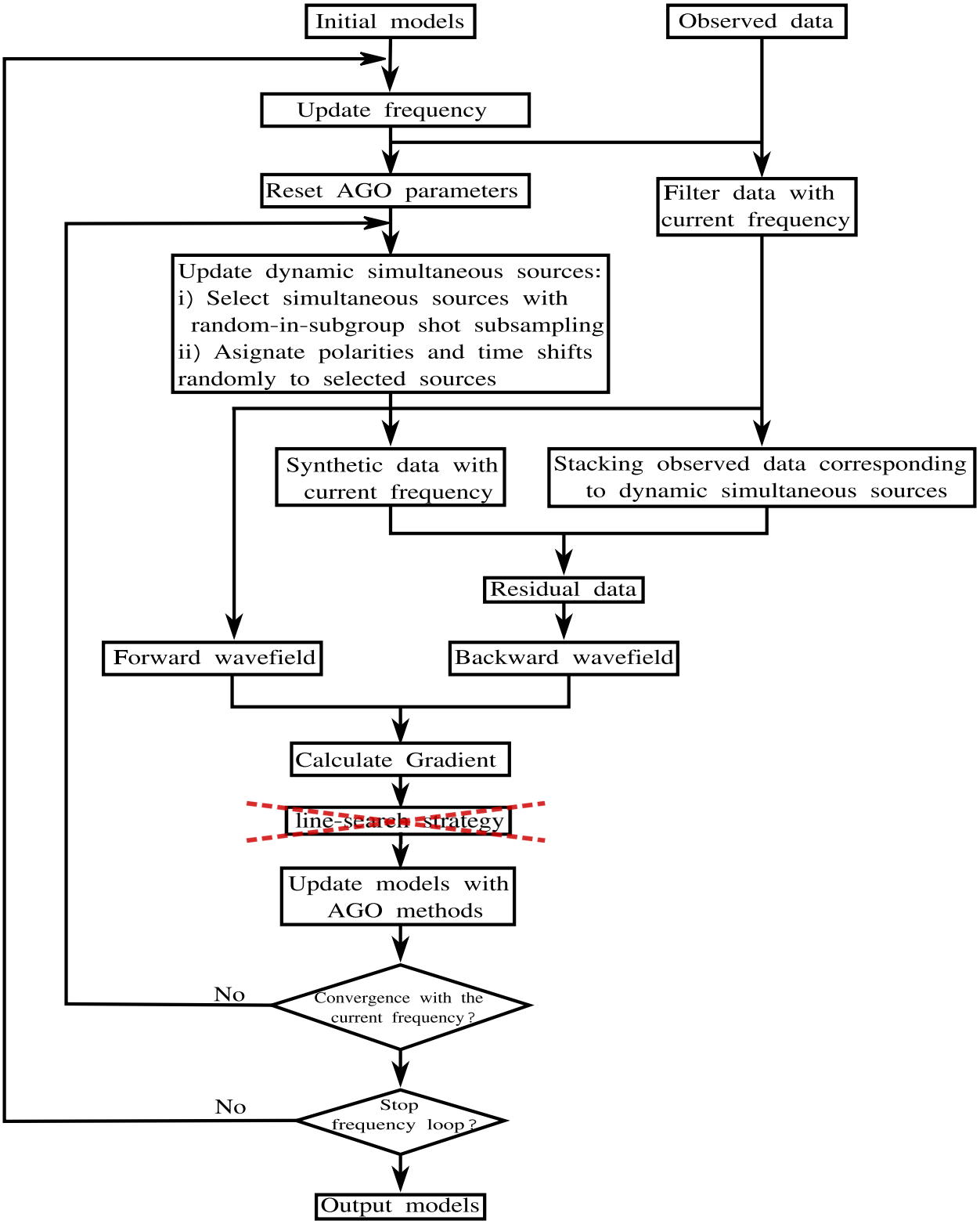

Figure 2. Multiscale dynamic simultaneous sources FWI workflow with AGO methods. Due to the fact that AGOs avoid the line-search strategy, we have crossed out that step.

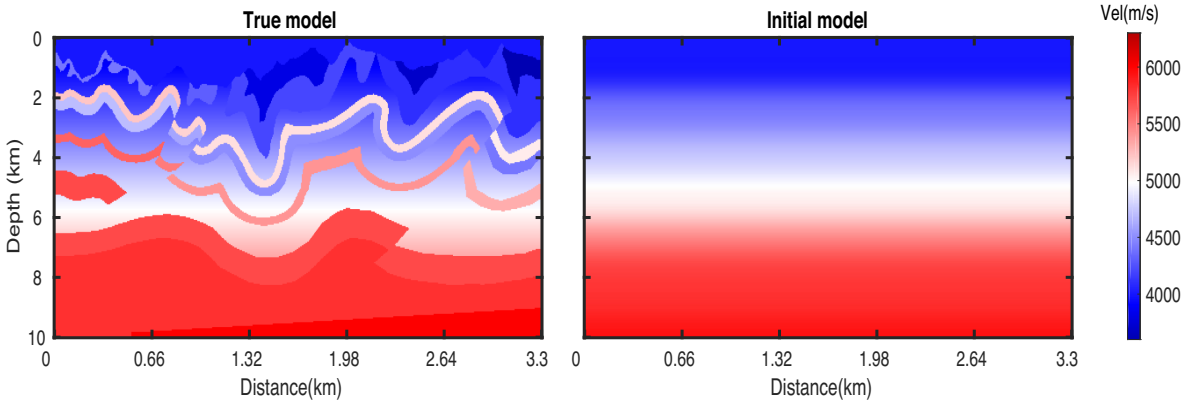

Figure 3. Canadian overthrust BP velocity model (left) and initial velocity model (right).

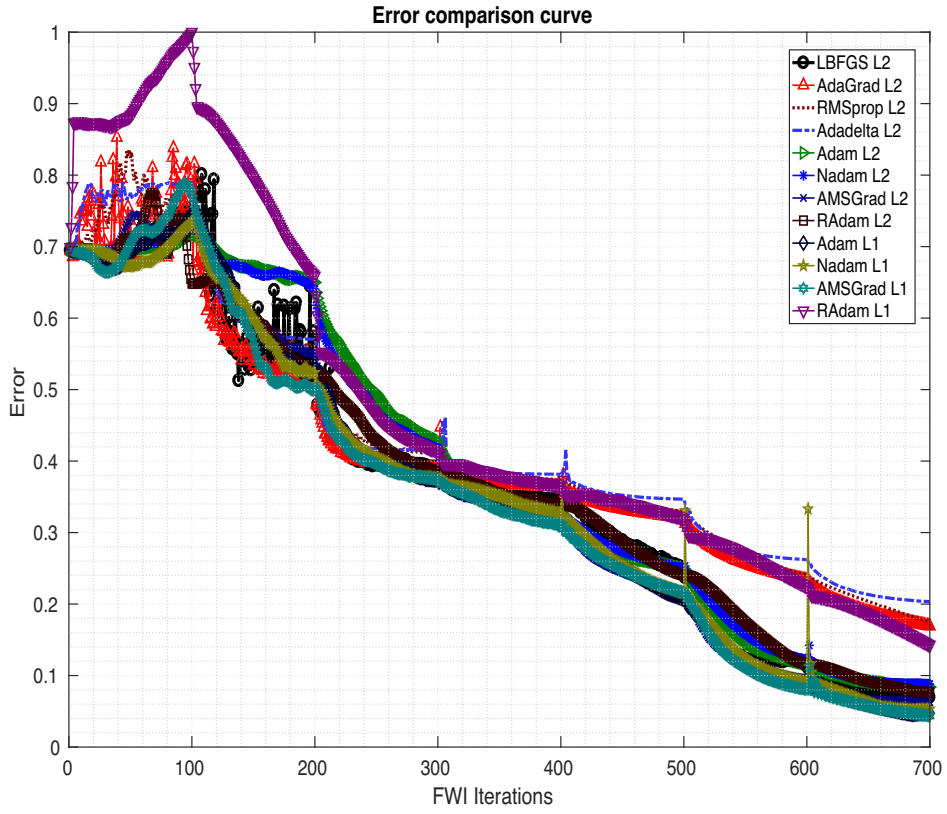

Figure 4. Normalized error curve for each one of the optimization methods applied into the FWI based on L_2 and L_1 norms.

the Marmousi 2 velocity model, see Martin *et al.* (2006). In order to compare the performance of the AGOs we compare their results with results obtained applying the L-BFGS QN method described in Nocedal & Writght (1999).

In this paper, we begin with a review of the FWI inversion process in the time domain, including a glance to dynamic simultaneous sources techniques. We continue, describing some of the AGOs used in this work. Next, we propose a formula to assign a step length in the AGO methods for multiscale FWI. This new formula allow us to avoid the line search in the FWI iterative process and we show how this affects the implementation of a conventional workflow. We follow with the description of the forward model for the acoustic case. Next, we applied the different AGOs to the FWI (based on L_2 and L_1 norms), combined with a dynamic simultaneous sources technique to the Canadian overthrust BP and the Marmousi 2 velocity models. We then discuss the results before our concluding remarks. Lastly, in Appendix A, we give a brief description about what are the AGOs and where they are popularly applied. Appendix B reviews how to obtain the gradient for misfit functions based on the L_1 and L_2 norms, using the adjoint-state method.

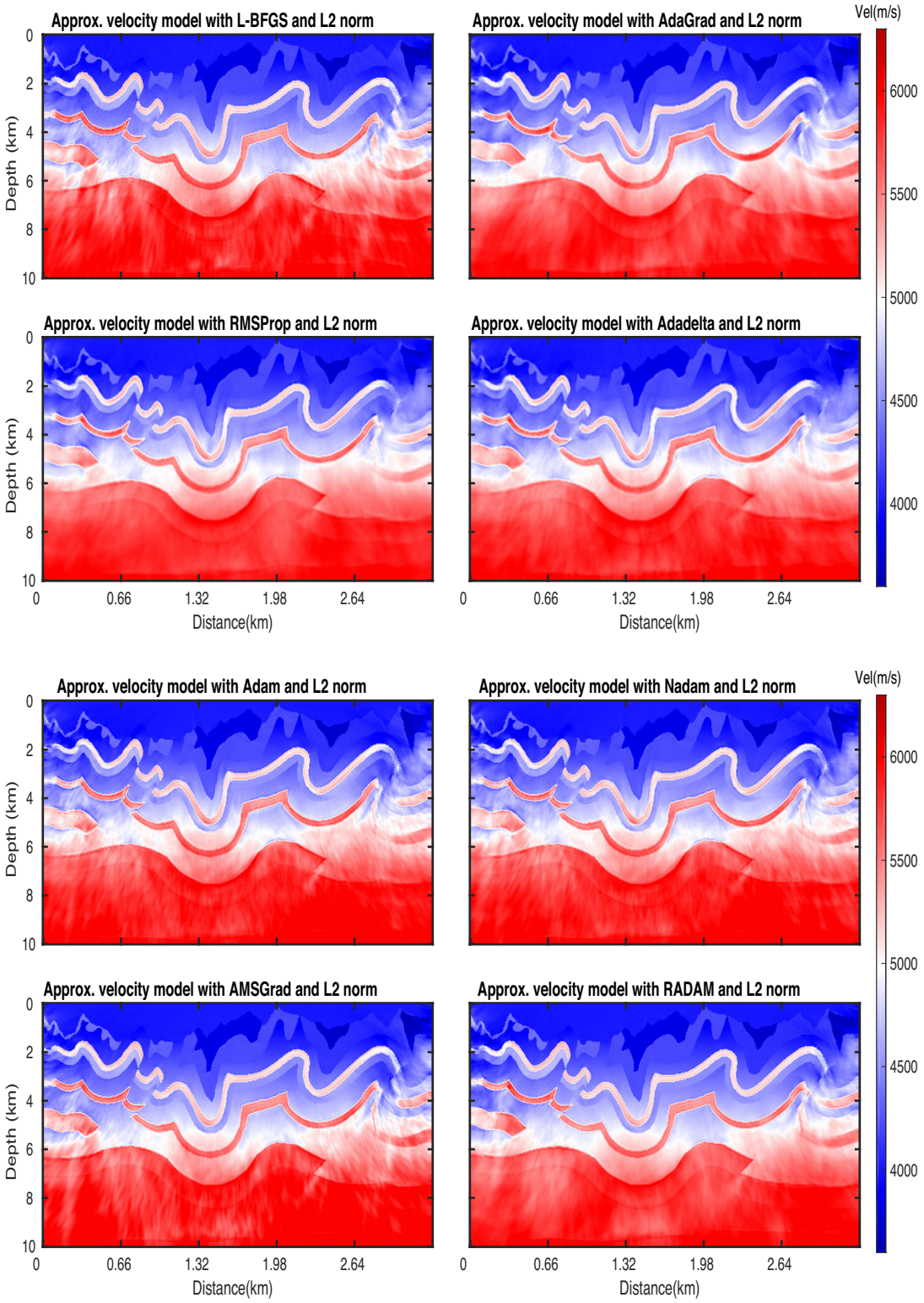

Figure 5. Final velocity models obtained after the inversion using each one of the optimization methods applied into the FWI based on L_2-norm.

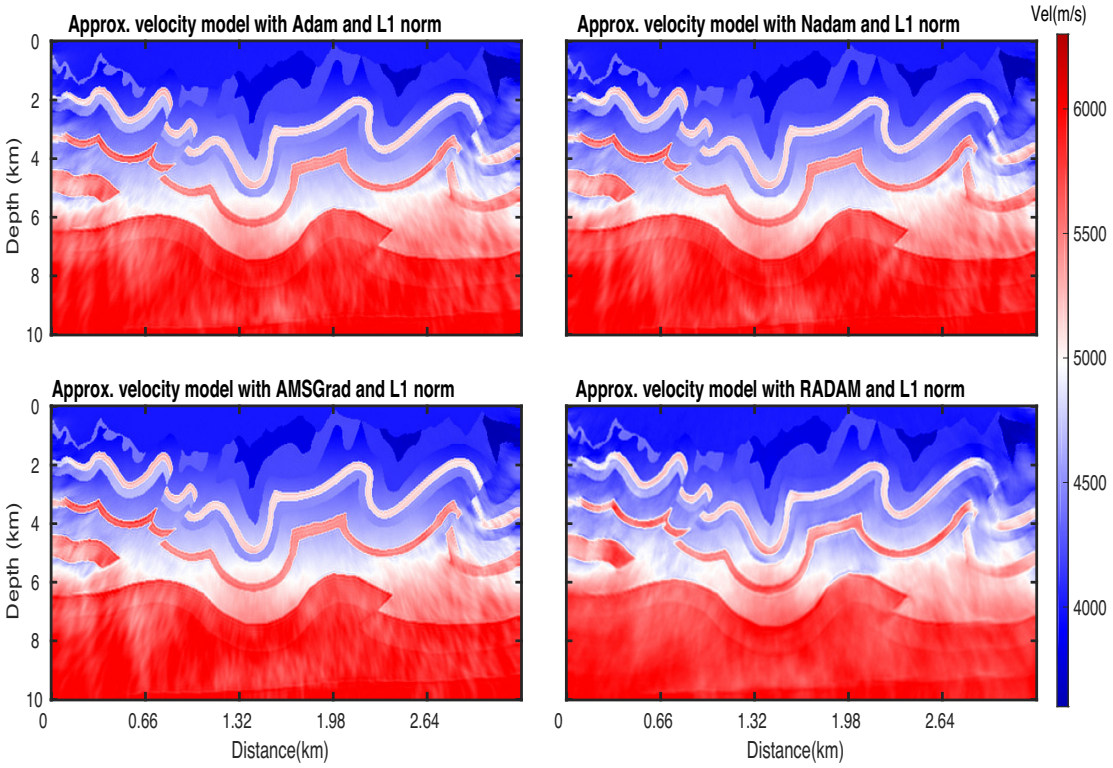

Figure 6. Final velocity models obtained after the inversion using the methods: Adam, Nadam, AMSGrad and RAdam; applied into the FWI based on L_1-norm.

2 TIME-DOMAIN FWI REVIEW

In this section, we give a brief account of FWI in the time domain. Let us define $V_{\text{obs}}(X_g, t|X_s)$ and $V_{\text{aprx}}(X_g, t|X_s)$ the observed and the approximated (or synthetic) data, respectively. The observed data are acquired by a source located at coordinates X_s, recorded by a geophone located at coordinates X_g at a time t. The approximated data are generated by a forward model that depends on $\vec{m}(X) = [m_1, ..., m_n]^T$, being $m_i(X)$ a physical parameter (e.g. velocity or density).

The goal of FWI is to minimize a misfit function, generally based on the L_2-norm of the difference between synthetic and observed data (using the least-squares criterion), see Tarantola (1984) and Virieux & Operto (2009):

$$\epsilon = \frac{1}{2}\sum_{s=1}^{N_s}\sum_{g=1}^{N_g}\int_0^T \parallel V_{\text{res}}(X_g, t|X_s) \parallel^2 dt, \tag{1}$$

or the non-quadratic misfit function like the one based on the L_1-norm of the difference between synthetic and observed data (using the least-absolute criterion), see Tarantola (1987), Crase *et al.* (1990), Brossier *et al.* (2010) and Jeong *et al.* (2013):

$$\epsilon = \sum_{s=1}^{N_s}\sum_{g=1}^{N_g}\int_0^T |V_{\text{res}}(X_g, t|X_s)|dt, \tag{2}$$

$V_{res}(X_g, t|X_s)$ is the residual between the observed and the synthetic data, given by:

$$V_{res}(X_g, t|X_s) = V_{aprx}(X_g, t|X_s) - V_{obs}(X_g, t|X_s), \tag{3}$$

where N_s and N_g are the number of sources and geophones, respectively. T is the duration of the forward simulation for each source. Note that ϵ is a function of $\vec{m}(X)$ since V_{obs} is fixed and V_{aprx} depends on $\vec{m}(X)$.

In order to reduce the cost function in eq. (1) or (2) to approximate $\vec{m}(X)$, the simplest way is to start from an initial model $\vec{m}_0(X)$ and begins an iterative search process as follows:

$$\vec{m}_{k+1} = \vec{m}_k - \alpha_k \vec{G}(\vec{m}_k), \tag{4}$$

where α_k is the kth step length and $\vec{G}(\vec{m}_k)$ is the kth misfit function's gradient. The specific form of eq. (4) will depend on the adopted optimization method. When QN methods are applied, the misfit function gradient in eq. (4) needs a multiplication by an approximation of the

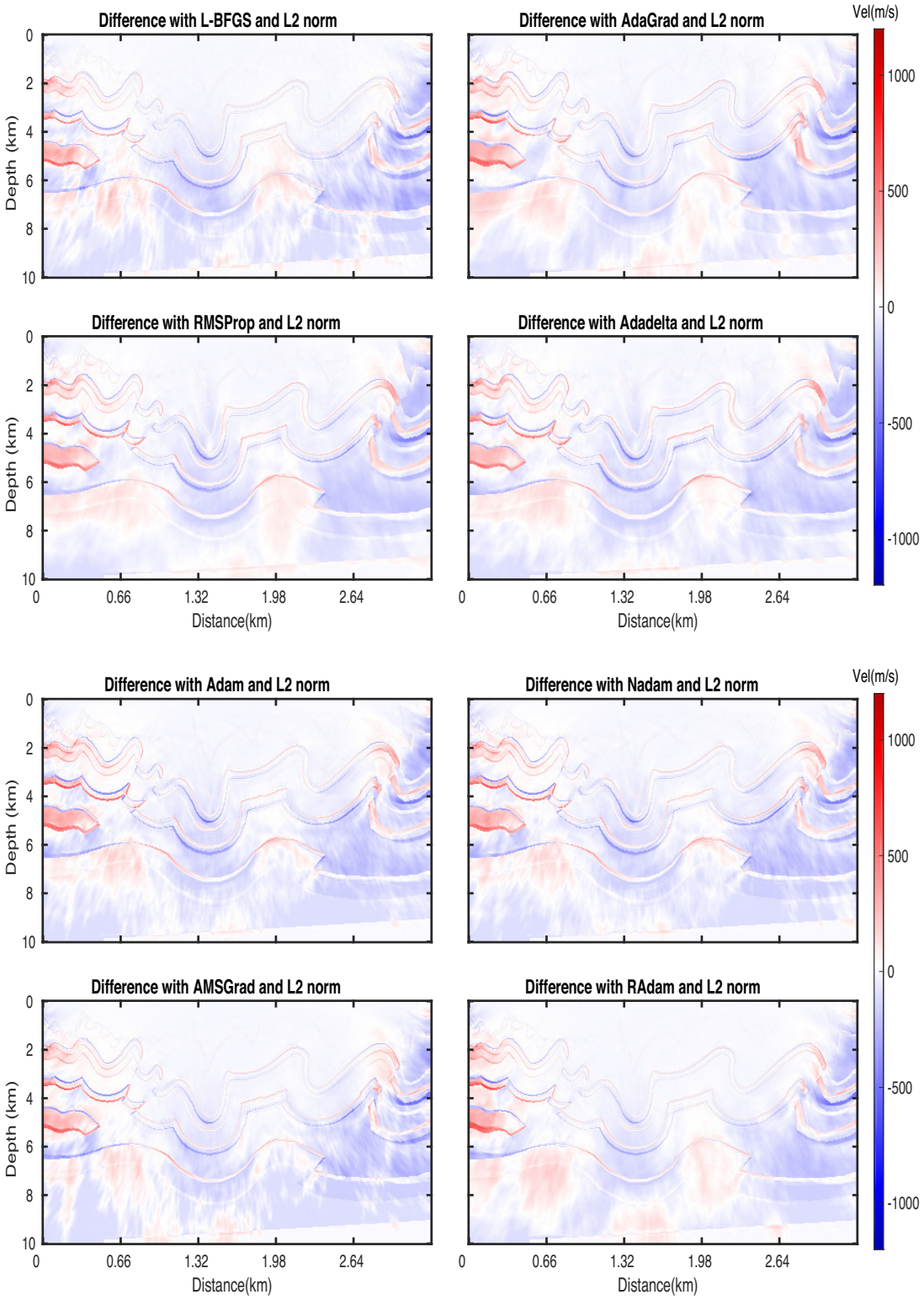

Figure 7. Differences between the true and the approximated velocity models using each one of the optimization methods applied into the FWI based on L_2-norm.

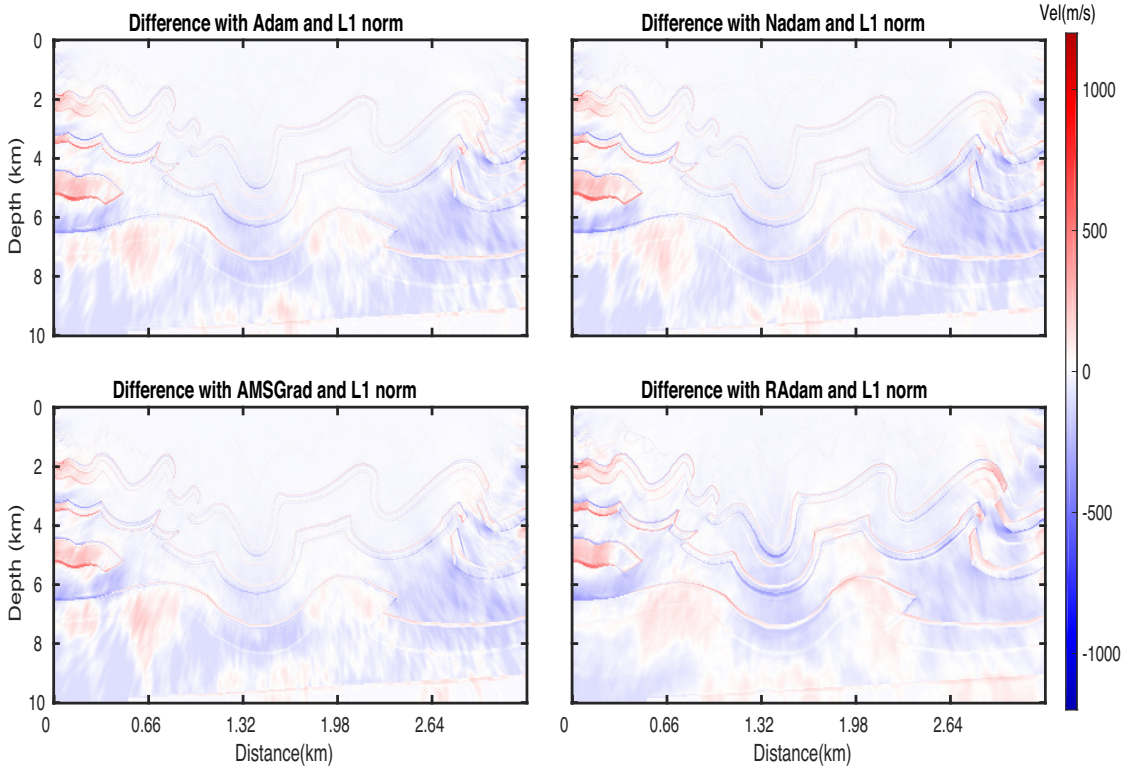

Figure 8. Differences between the true and the approximated velocity models using the methods: Adam, Nadam, AMSGrad and RAdam; applied into the FWI based on L_1-norm.

Hessian's inverse H_k^{-1} to get:

$$\vec{m}_{k+1} = \vec{m}_k - \alpha_k \underbrace{H_k^{-1}\vec{G}(\vec{m}_k)}_{d_k}. \tag{5}$$

Each gradient component $G_i = \frac{\partial \epsilon(\vec{m})}{\partial m_i}$ is computed through the zero-lag correlation between the forward-propagated source wavefield, $D(X, t|X_s)$ and the backward-propagated residual wavefield, $U(X, t|X_s)$, see Tarantola (1984, 1987) and Boonyasiriwat *et al.* (2009), as follows:

$$G_i = \frac{\partial \epsilon(\vec{m}(X))}{\partial m_i(X)} = \sum_{s=1}^{N_s} \int_0^T D(X, t|X_s) U(X, t|X_s) dt. \tag{6}$$

The fields D and U depend on the forward model used to generate the approximated data.

2.1 Dynamic simultaneous sources FWI to crosstalk noise reduction

Conventional FWI is a very expensive computational process. The reason being that, the computation of the gradient is done through the zero-lag correlation between the fields D and U for each source. This requires thrice the time used for a forward simulation and according to eq. (6) this needs to be done for N_s sources. To reduce the computational cost of this process, encoding techniques were initially applied in pre-stack migration (Romero *et al.* 2000; Jing *et al.* 2000) and subsequently in FWI (Krebs *et al.* 2009). Variants of these techniques for FWI are based on phase reversal, phase shifting, time-shifting, see Herrmann *et al.* (2009) and Ben-Hadj-Ali *et al.* (2011). These techniques take advantage of the forward model's linearity with respect to the sources. It activates various sources at a one single forward simulation, this is known as a supershot.

It is well known that source-encoding techniques introduce an amount of noise called crosstalk noise, which if it is not greatly reduced, it can degrade the accuracy of the physical parameters obtained from the FWI, see Romero *et al.* (2000). To reduce the crosstalk noise, in this work, we use a dynamic simultaneous sources technique that consists on the combination of three popular source-encoding strategies: random-in-subgroup shot subsampling (Díaz & Guitton 2011; Ha & Shin 2013); random time-shifting (Zhan *et al.* 2013; Schuster *et al.* 2011) and random polarities (Boonyasiriwat & Schuster 2010); with the property that these settings change randomly in each iteration, so that the term dynamic simultaneous sources has been coined to describe this method (Krebs *et al.* 2009). These strategies are implemented in the following order:

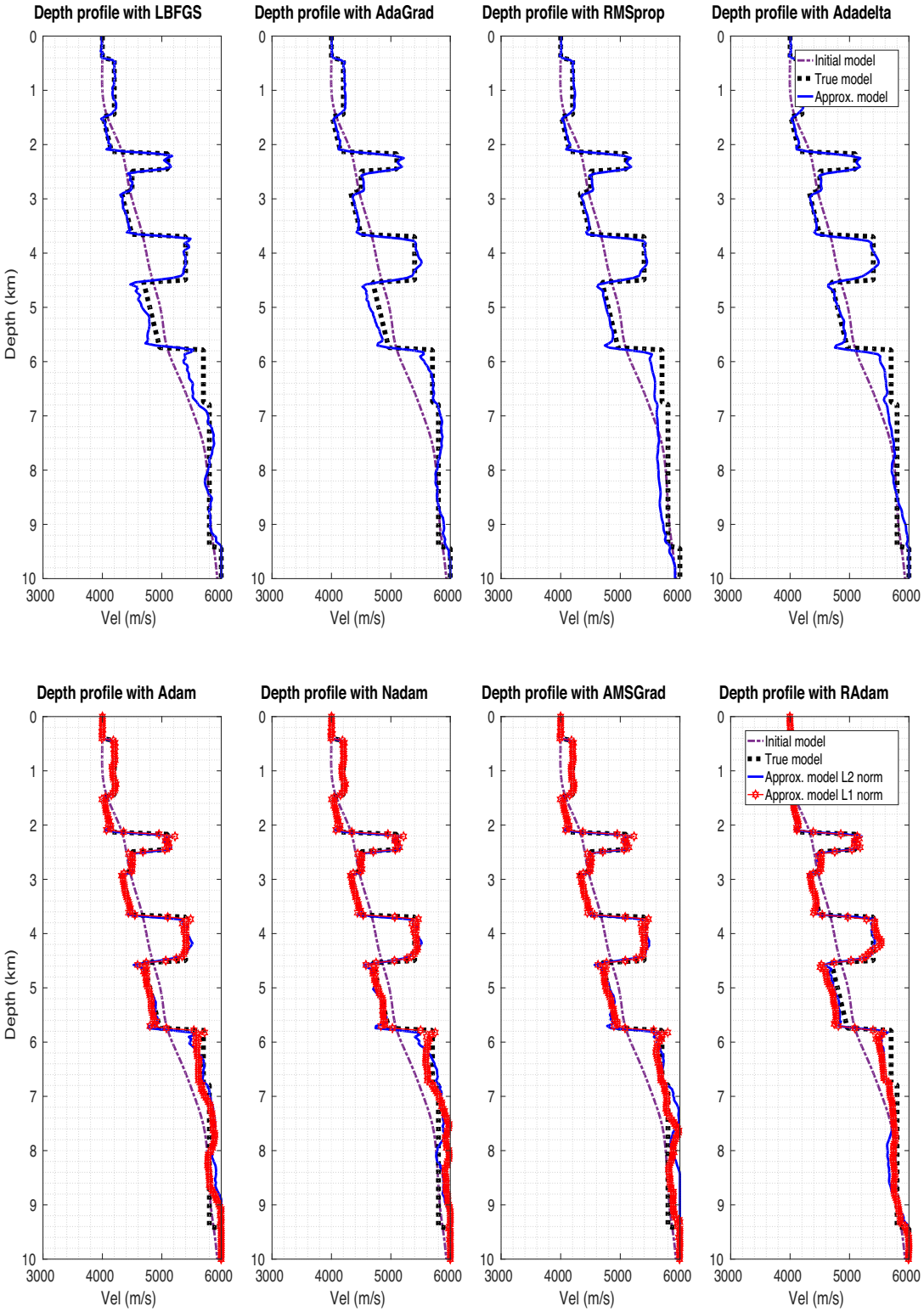

Figure 9. Comparison of the true velocity profile and the recovered velocity profiles at $x = 820$ m, obtained using each one of the optimization methods into the FWI based on L_2-norm. For the FWI based on L_1-norm, we only show the results obtained with the methods: Adam, Nadam, AMSGrad and RAdam.

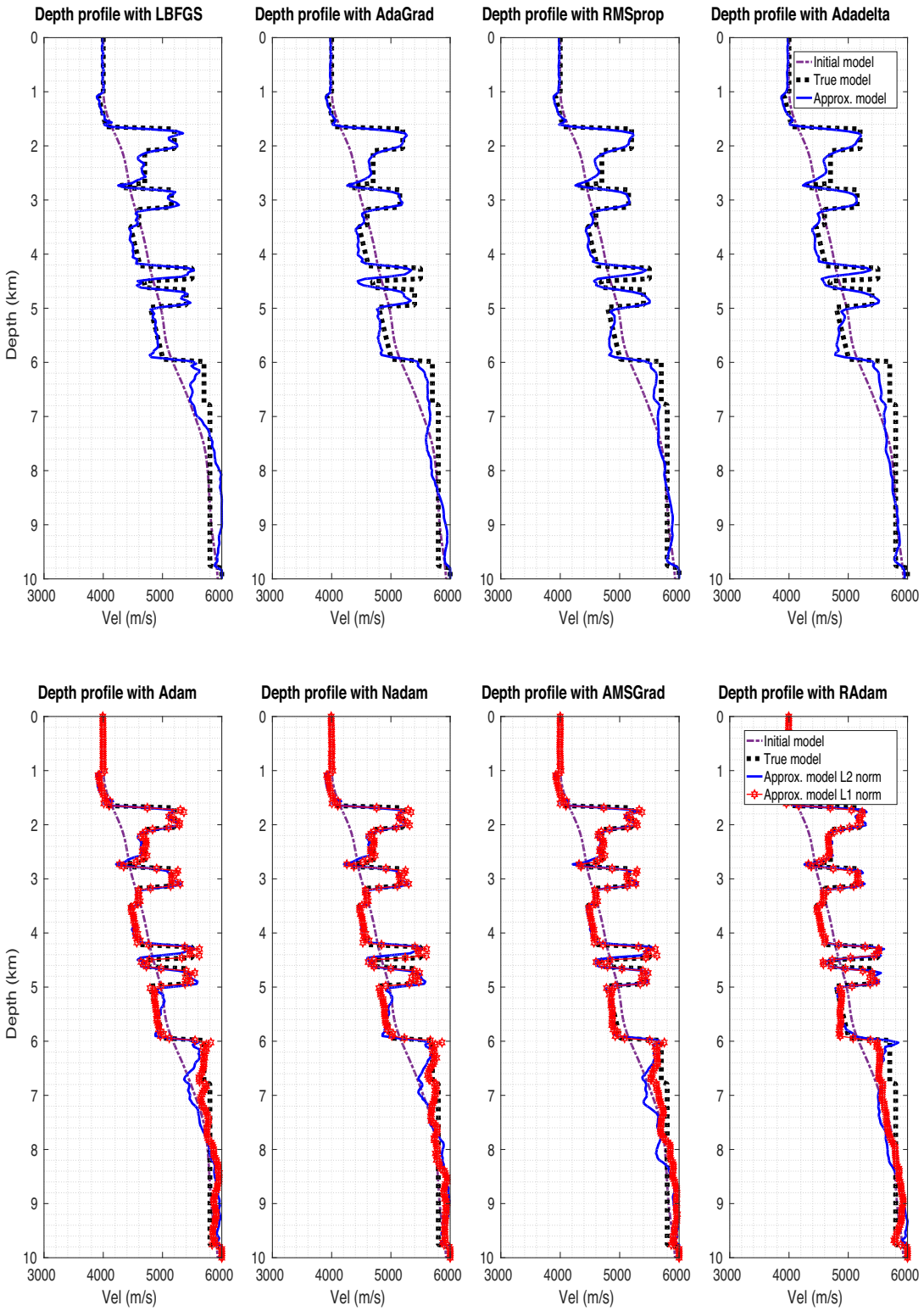

Figure 10. Comparison of the true velocity profile and the recovered velocity profiles at $x = 2$ km, obtained using each one of the optimization methods into the FWI based on L_2-norm. For the FWI based on L_1-norm, we only show the results obtained with the methods: Adam, Nadam, AMSGrad and RAdam.

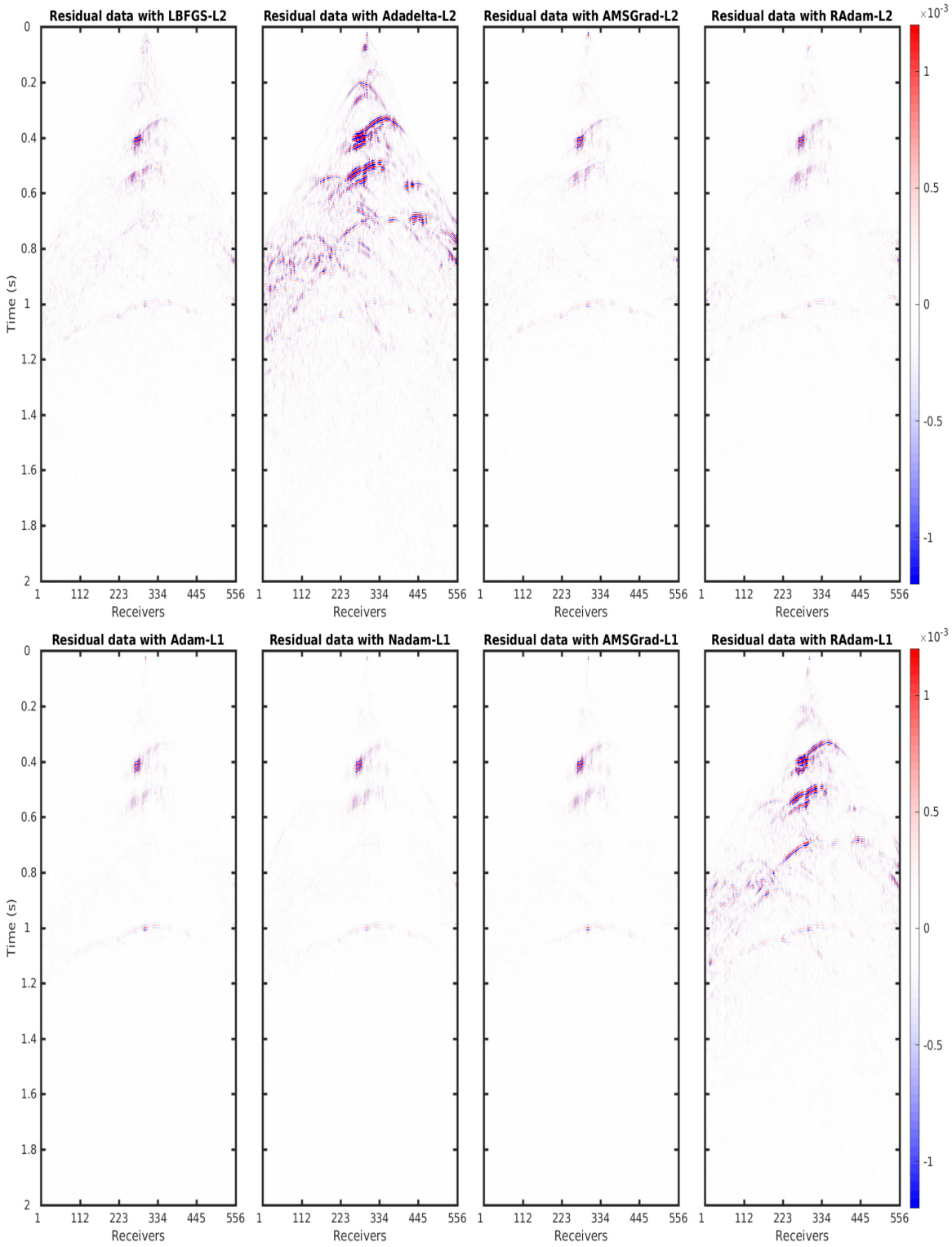

Figure 11. Residuals (difference between observed and approximated shot gathers for a source located at the middle of the receivers' axis.) using the final inverted velocity models obtained with L-BFGS and some AGO's methods, using the FWI based on L_2 and L_1 norms.

(i) Consider $X_{ss} = \{X_s : s = 1, 2, \ldots, N_s\}$ the set of sources used in the experiment. In each FWI iteration, we will randomly select a subset $\gamma(X_{ss}) \subset X_{ss}$ through what is referred as *random-in-subgroup shot subsampling*, see Díaz & Guitton (2011). The number of sources in the supershot composed by $\gamma(X_{ss})$, is given by $N_{ss} = \text{round}(\frac{f}{\lambda})$, where f is the current frequency of the multiscaling and $\lambda = \frac{f_{\max}}{n_s}$, the ratio of the maximum number of sources activated n_s for the maximum frequency $f_{\max}$, see Shi & He (2018). N_{ss} therefore increases as the frequencies increase in the multiscaling process.

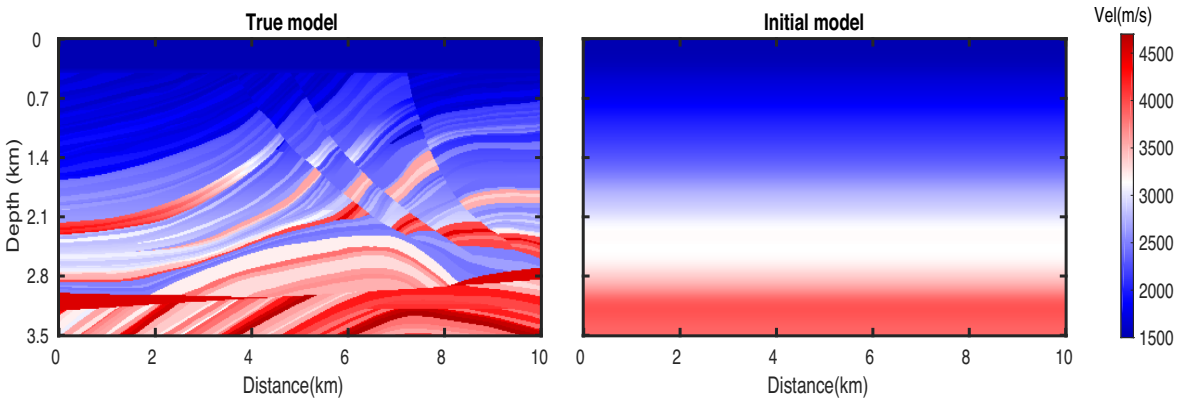

Figure 12. Marmousi 2 velocity model (left) and initial velocity model (right).

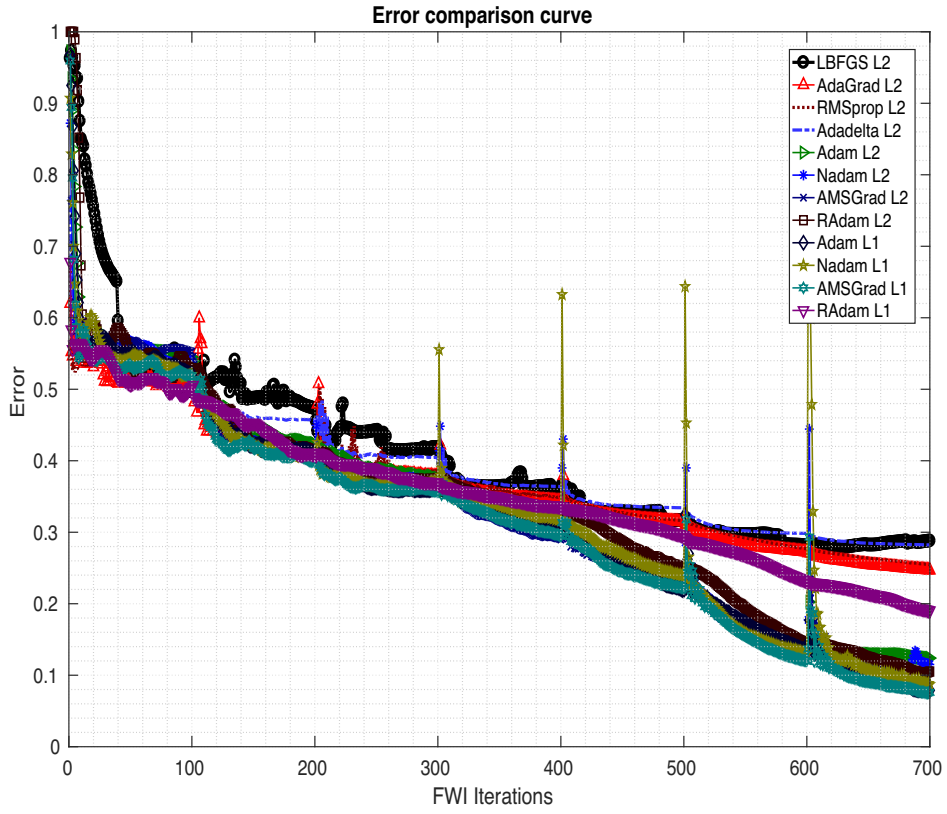

Figure 13. Normalized error curve for each one of the optimization methods applied into the FWI based on L_2 and L_1 norms.

(ii) A polarity $(-1)^{p(s)}$ and a time-shift τ_s will be randomly assigned, see Schuster *et al.* (2011), to have a signal source $(-1)^{p(s)}w(t-\tau_s)$ that corresponds to each source $X_s \in \gamma(X_{ss})$. Both quantities are random although $p(s)$ is a positive integer whereas τ_s comes from the equally spaced partition: the N_{ss} partition belonging the interval $[0, 0.2T]$. T is the forward simulation's duration.

In this way, if $\gamma(X_{ss}) = \{X_s : s = 1, 2, ..., N_{ss} < N_s\}$ is the set of selected sources for a current supershot, the forward-propagated sources and back-propagated residuals wavefields for this source-encoding strategies in the time domain become:

$$\tilde{D}(X, t|\gamma(X_{ss})) = \sum_{s=1}^{N_{ss}} (-1)^{p(s)} D(X, t - \tau_s|X_s) \tag{7}$$

and

$$\tilde{U}(X, t|\gamma(X_{ss})) = \sum_{k=1}^{N_{ss}} (-1)^{p(k)} U(X, t - \tau_k|X_k). \tag{8}$$

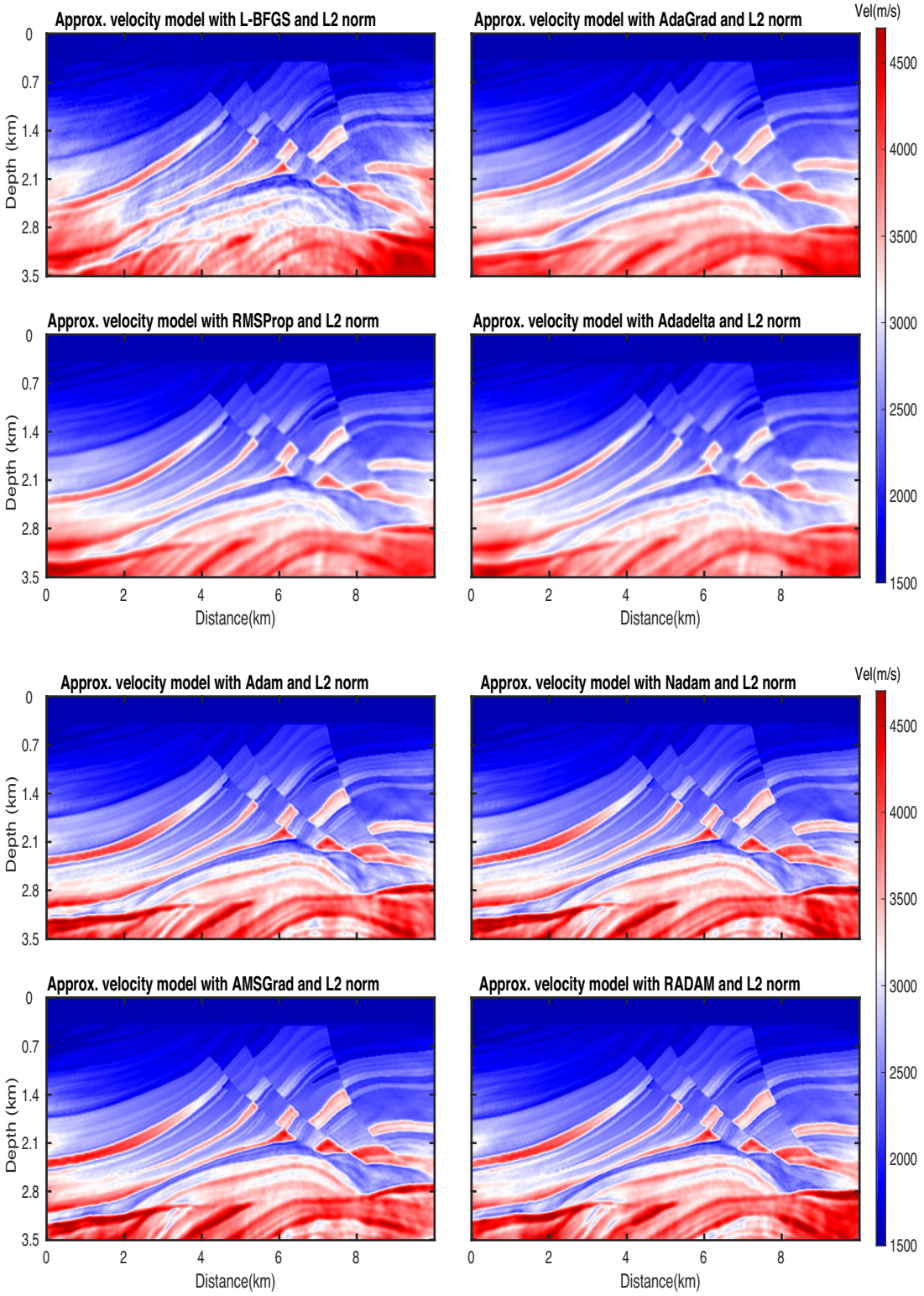

Figure 14. Final velocity models obtained after the inversion using each one of the optimization methods applied into the FWI based on L_2-norm.

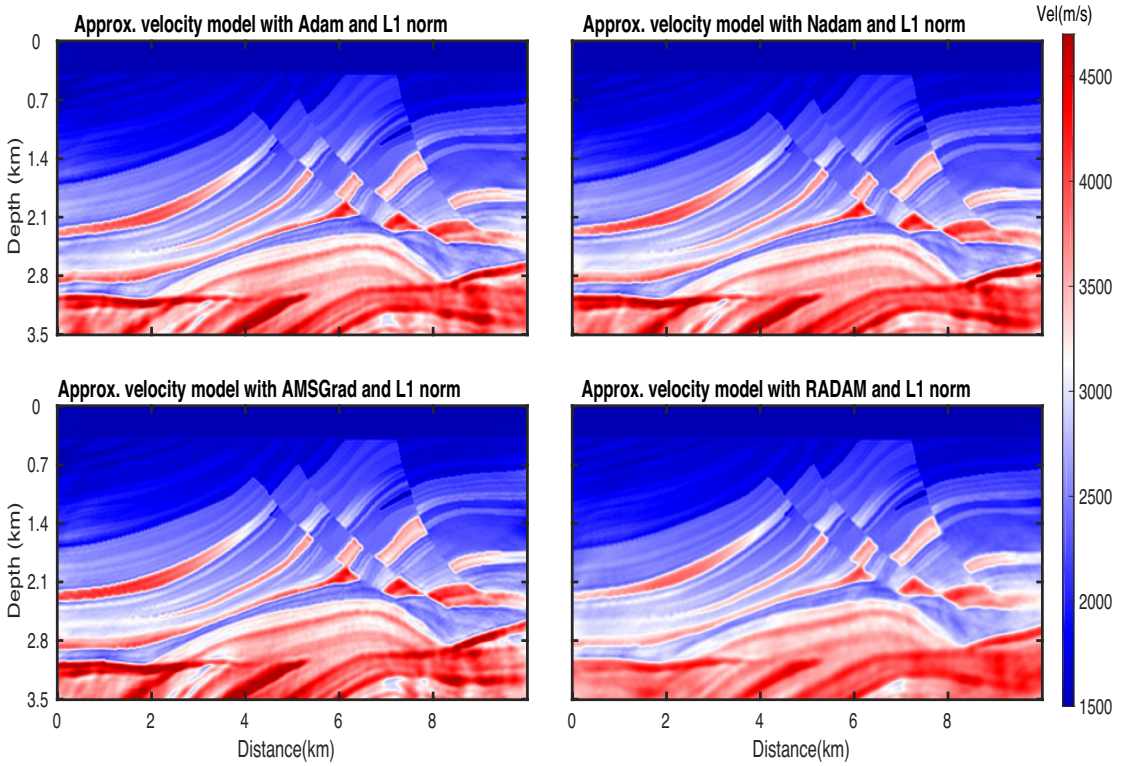

Figure 15. Final velocity models obtained after the inversion using the methods: Adam, Nadam, AMSGrad and RAdam; applied into the FWI based on L_1-norm.

Hence, a gradient component with this dynamic simultaneous source technique becomes:

$$\begin{aligned}\tilde{G} &= \int_0^T \tilde{D}(X,t|\gamma(X_{ss}))\tilde{U}(X,t|\gamma(X_{ss}))dt \\ &= \underbrace{\int_0^T \sum_{s=k}^{N_{ss}} D(X,t-\tau_s|X_s)U(X,t-\tau_k|X_k)dt}_{G} \\ &+ \underbrace{\int_0^T \sum_{s\neq k}^{N_{ss}} (-1)^{p(s)}D(X,t-\tau_s|X_s)(-1)^{p(k)}U(X,t-\tau_k|X_k)dt}_{\delta G} = G+\delta G. \end{aligned} \tag{9}$$

Note that G in eq. (9) corresponds to a gradient component obtained conventionally and δG corresponds to the crosstalk noise generated from the zero-lag correlation between the fields D and U that come from different sources. Due to the fact that in the crosstalk noise term, the D and U fields have different polarities and different time-shiftings, this leads to incoherent stack computations for the zero-lag correlation, reducing the value of the crosstalk noise term.

3 SOME ADAPTIVE GRADIENT OPTIMIZATION METHODS

In this paper, we apply some AGO methods that come from the machine learning (ML) community (Erickson *et al.* 2017) to reduce the misfit function in the FWI. Appendix A gives a brief general description of AGOs. Next we give a brief account of the AGO methods applied in this work: AdaGrad, RMSprop, Adadelta, Adam, Nadam, AMSGrad and RAdam. These have been the most used by the ML community in recent years. We provide suitable references for each method. Theoretical justification of the AGO's structure is designed to guarantee the local or global minimum convergence for convex misfit functionals. Since the FWI misfit functions are highly nonlinear, they can be locally convex, so the AGOs applied in FWI only guarantee convergence to a local minimum. However, this depends on a suitable calibration of the step length for each iteration. The step-length rule that we propose for AGOs in FWI does not necessarily guarantee reaching the global minimum. However, a correct calibration of the parameters of this step-length rule guarantees a local minimum. A bad selection/calibration of the step length could lead to high fluctuations around the local minimum or the worst case scenario, that the method diverges (Ruder 2016).

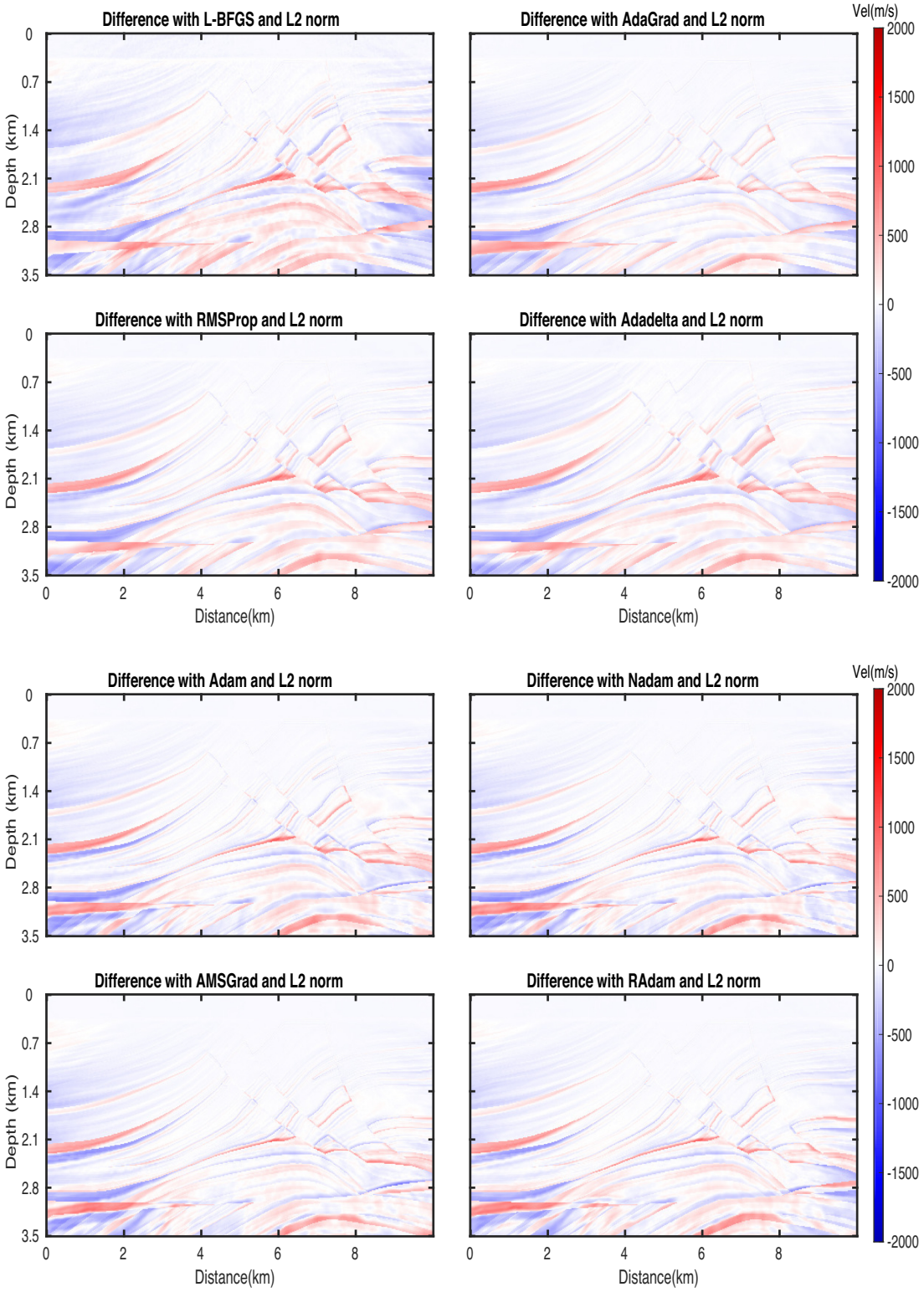

Figure 16. Differences between the true and the approximated velocity models using each one of the optimization methods applied into the FWI based on L_2-norm.

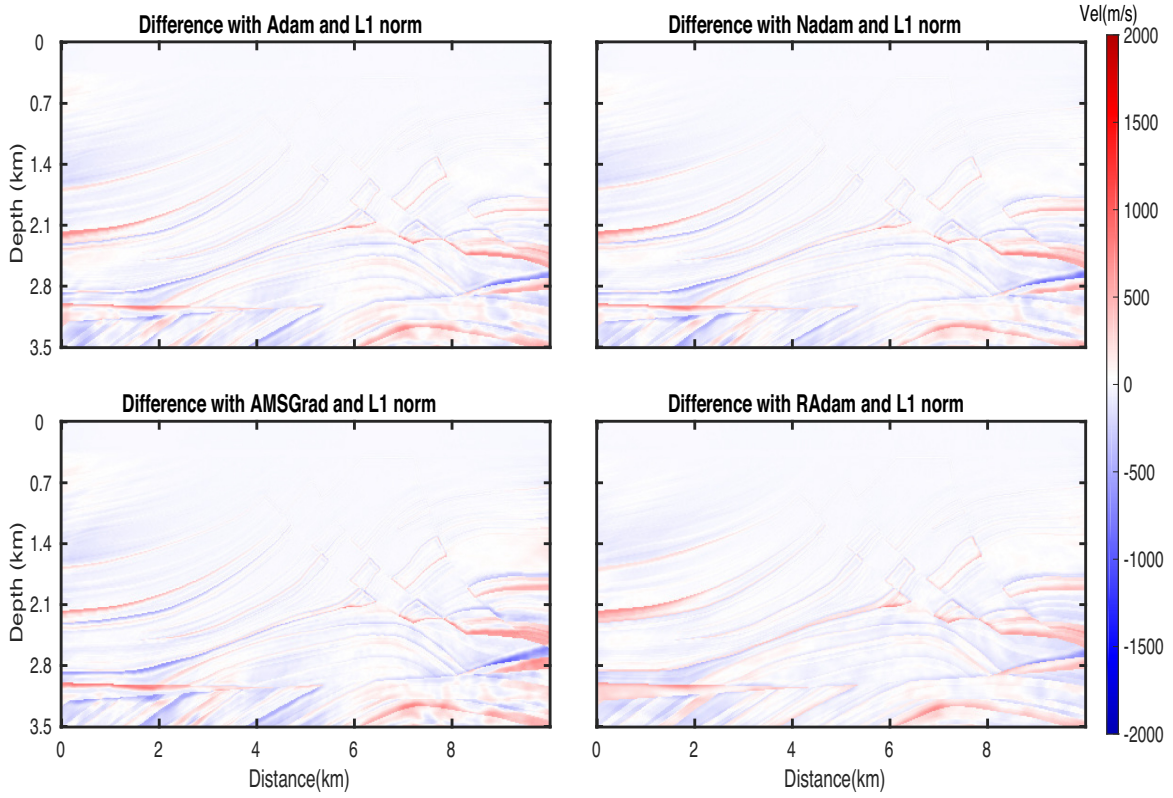

Figure 17. Differences between the true and the approximated velocity models using the methods: Adam, Nadam, AMSGrad and RAdam; applied into the FWI based on L_1-norm.

The choice for α will be explained in Section 4. Hence, in order to chose the step length, in Section 4 we propose a new formula that adapts to the different frequencies used during the FWI multiscaling FWI.

Next, we briefly describe a collection of the AGO methods used in this work. Let us consider $G(m_k) = \frac{\partial \epsilon(\vec{m})}{\partial m_k}$, a gradient component of a misfit function, computed with the adjoint-state method for the physical parameter m_k at the kth iteration of the FWI and α a given step length. The parameters in each of the AGOs formulae: G, S, D and V have the same dimension as the parameter model m_k. The following AGOs refer to this function.

3.1 AdaGrad

Adaptive gradient or AdaGrad method (Duchi *et al.* 2011) is an algorithm for gradient-based optimization. In each iteration the goal is to adapt the step length, performing smaller updates for parameters associated with frequently occurring features, and larger updates for parameters associated with infrequent features. This makes it suitable to deal with sparse data and is one of the most used by Google Inc. to train large-scale neural networks, see Ruder (2016). The step length is divided by the square root of S that corresponds to the cumulative sum of the square gradients, the formula is given by:

$$m_{k+1} = m_k - \frac{\alpha}{\sqrt{S_k} + \eta} \cdot G(m_k), \tag{10}$$

with

$$S_k = S_{k-1} + [G(m_k)]^2, \tag{11}$$

where S is initialize to 0 and η is a smoothing term to avoid division by zero (we use $\eta = 1 \times 10^{-7}$).

3.2 RMSprop

Root Mean Square prop or RMSprop method (Tieleman & Hinton 2012) is an adaptive gradient method. It differs from Adagrad since, this method considers the exponential moving average of the gradients as in the momentum method (Polyak 1964), instead of the cumulative sum

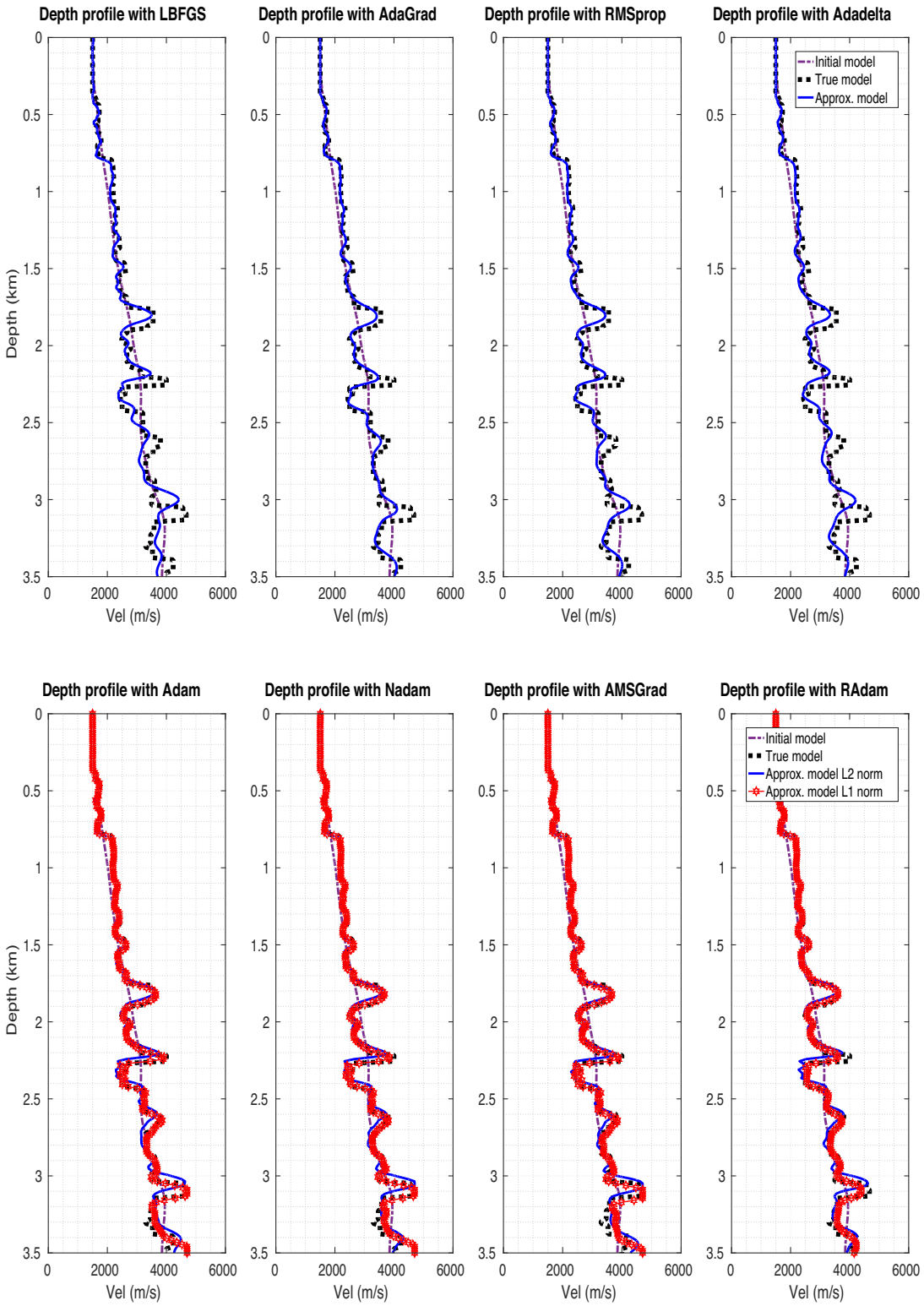

Figure 18. Comparison of the true velocity profile and the recovered velocity profiles at $x = 2.5$ km, obtained using each one of the optimization methods into the FWI based on L_2-norm. For the FWI based on L_1-norm, we only show the results obtained with the methods: Adam, Nadam, AMSGrad and RAdam.

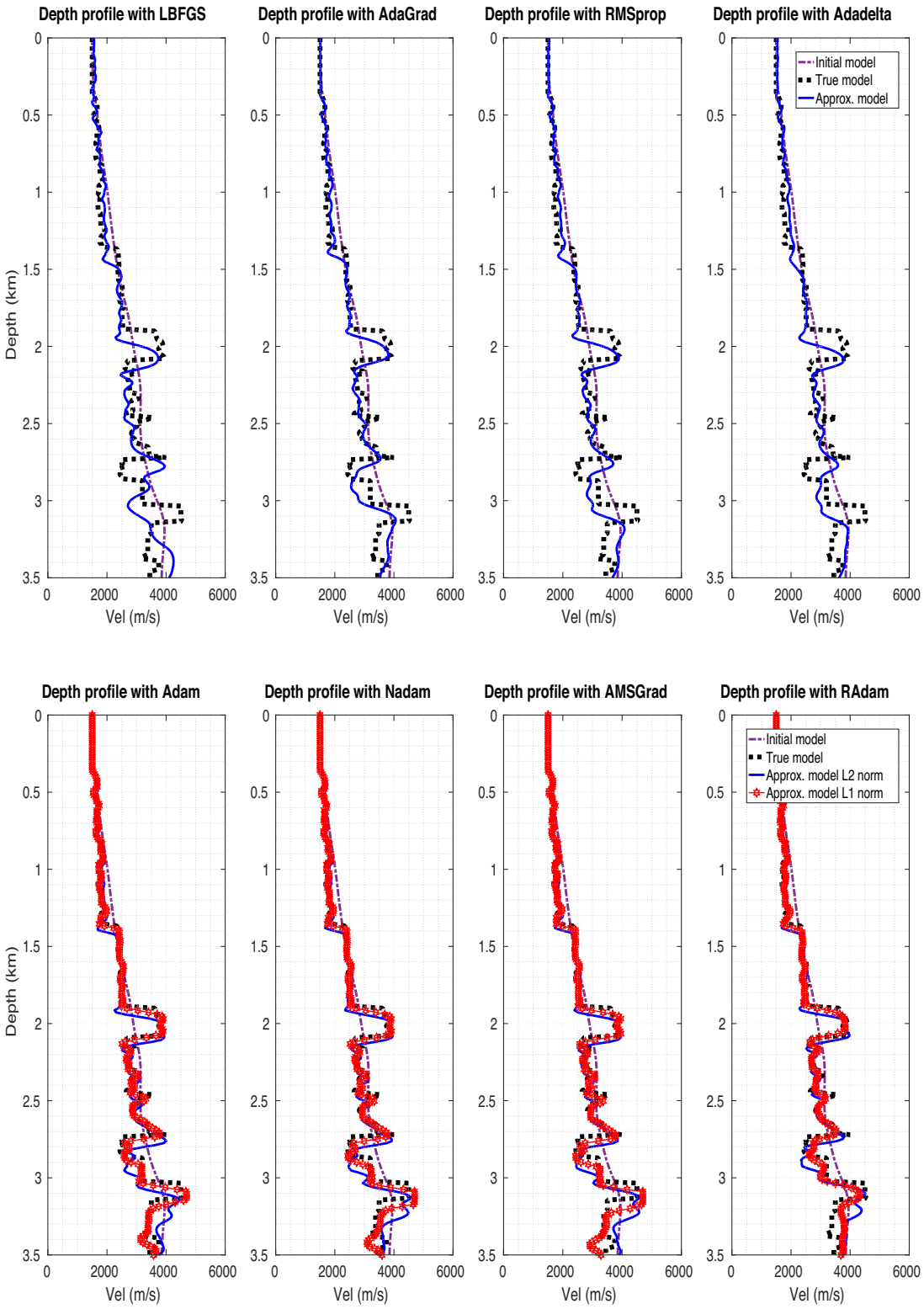

Figure 19. Comparison of the true velocity profile and the recovered velocity profiles at $x = 5$ km, obtained using each one of the optimization methods into the FWI based on L_2-norm. For the FWI based on L_1-norm, we only show the results obtained with the methods: Adam, Nadam, AMSGrad and RAdam.

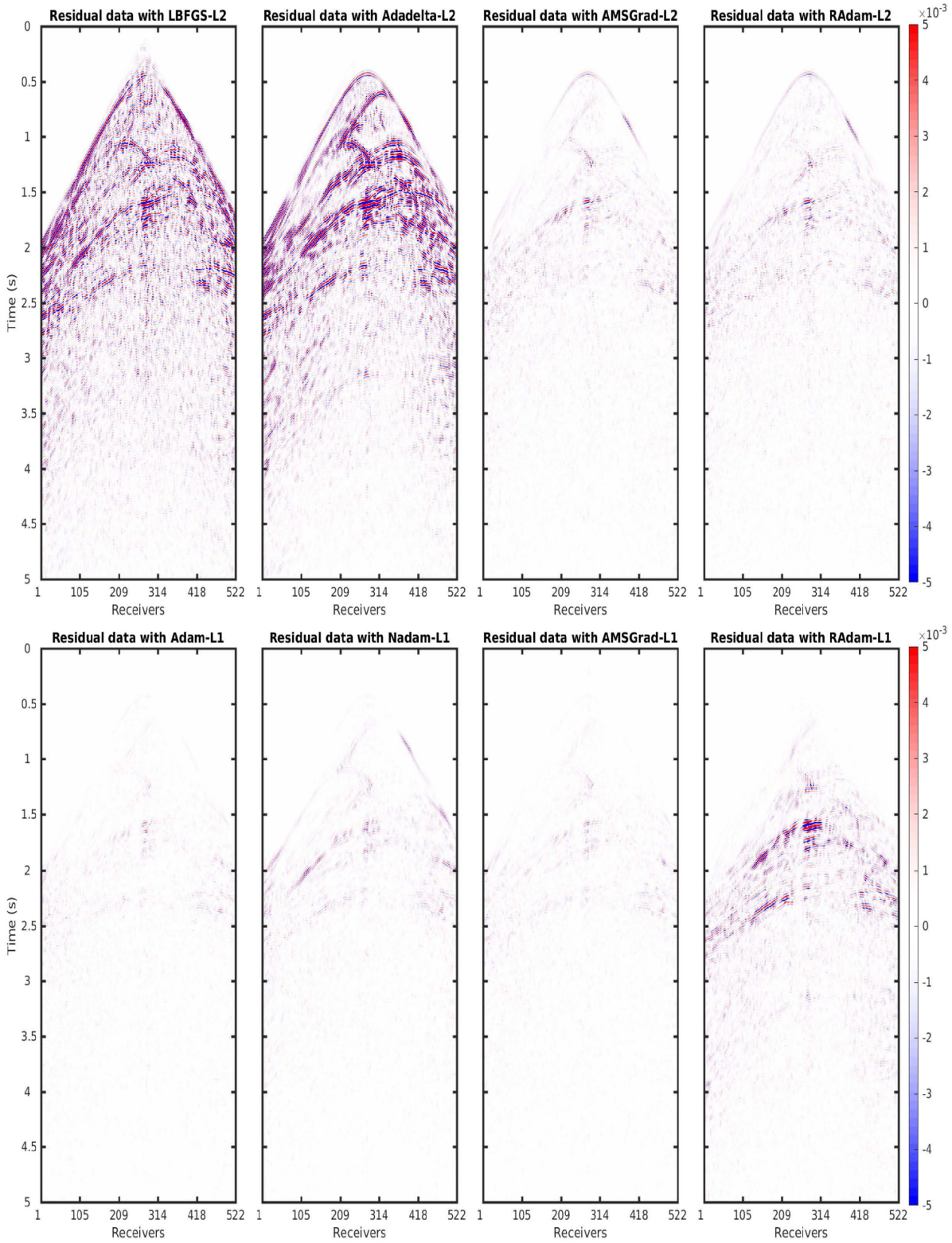

Figure 20. Residuals (difference between observed and approximated shot gathers for a source located at the middle of the receivers' axis.) using the final inverted velocity models obtained with L-BFGS and some AGOs methods, using the FWI based on L_2 and L_1 norms.

of the square gradient. The formula is given as follows:

$$m_{k+1} = m_k - \frac{\alpha}{\sqrt{S_k} + \eta} \cdot G(m_k), \tag{12}$$

with

$$S_k = \beta S_{k-1} + (1-\beta)[G(m_k)]^2 \tag{13}$$

where S is initialize to 0, η is a smoothing term to avoid division by zero (we use $\eta = 1 \times 10^{-6}$) and $\beta = 0.9$ is a value recommended by the authors of this method.

3.3 Adadelta

Adadelta method (Zeiler 2012), is an extension to the AdaGrad method but instead of accumulating the gradient square from the beginning (this could reduce the step length of some parameters), it gradually reduces the contribution of previous terms. The formula is given by:

$$m_{k+1} = m_k - \alpha \frac{\sqrt{D_{k-1} + \eta}}{\sqrt{S_k} + \eta} \cdot [G(m_k)], \tag{14}$$

with

$$D_k = \beta D_{k-1} + (1-\beta)[m_k - m_{k-1}]^2, \tag{15}$$

$$S_k = \beta S_{k-1} + (1-\beta)[G(m_k)]^2, \tag{16}$$

where S and D are initialize to 0; $\beta = 0.95$ and $\eta = 1 \times 10^{-6}$ are values used in the Keras library (Chollet 2017).

This method was originally designed to have a fixed unitary step length which becomes incompatible to the frequency-dependent step length required in the multiscaling FWI. Without this frequency-dependent step length, the method could diverge as we will see in Section 4.

3.4 Adam

Adaptive moment estimation, or Adam method (Kingma & Ba 2015), is a combination of the momentum method (Polyak 1964) and the RMSprop method, since the main gradient is computed through the exponential moving average (like in the momentum method). In addition, the size of the step considers the square root of the exponential moving average of the square gradients (like in RMSprop). The formula is given by:

$$m_{k+1} = m_k - \frac{\alpha}{\sqrt{\hat{S}_k} + \eta} \cdot \hat{V}_k, \tag{17}$$

where

$$\hat{V}_k = \frac{V_k}{1-\beta_1^k} \tag{18}$$

$$\hat{S}_k = \frac{S_k}{1-\beta_2^k}, \tag{19}$$

with

$$V_k = \beta_1 V_{k-1} + (1-\beta_1) G(m_k) \tag{20}$$

$$S_k = \beta_2 S_{k-1} + (1-\beta_2)[G(m_k)]^2, \tag{21}$$

the superindex k, in β_1 and β_2, represents a power that corresponds to the iteration number; V and S are set to 0 at the beginning. $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\eta = 1 \times 10^{-8}$ are values recommended by the authors of this method.

3.5 Nadam

Nesterov-accelerated Adaptive Moment Estimation or Nadam method, see Dozat (2016), is a combination of the Nesterov accelerated gradient (NAG) method (Nesterov 1983) and the Adam method. In order to describe this combination, note that the Adam method can be written as follows:

$$m_{k+1} = m_k - \frac{\alpha}{\sqrt{\hat{S}_k} + \eta} \left[\beta_1 \hat{V}_{k-1} + \frac{1-\beta_1}{1-\beta_1^k} \cdot G(m_k) \right], \tag{22}$$

then the Nesterov method is used to update the gradient in the next step, substituting $\hat{V}_{k-1}$ by the current $\hat{V}_k$, obtaining:

$$m_{k+1} = m_k - \frac{\alpha}{\sqrt{\hat{S}_k} + \eta}\left[\beta_1 \hat{V}_k + \frac{1-\beta_1}{1-\beta_1^k} \cdot G(m_k)\right], \tag{23}$$

where

$$\hat{V}_k = \frac{V_k}{1-\beta_1^k} \tag{24}$$

$$\hat{S}_k = \frac{S_k}{1-\beta_2^k}, \tag{25}$$

with

$$V_k = \beta_1 V_{k-1} + (1-\beta_1) G(m_k) \tag{26}$$

$$S_k = \beta_2 S_{k-1} + (1-\beta_2)[G(m_k)]^2, \tag{27}$$

the superindex k, in β_1 and β_2, represents a power that corresponds to the iteration number; V and S are set to 0 at the beginning. $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\eta = 1 \times 10^{-7}$ are values used in the Keras' library (Chollet 2017).

3.6 AMSGrad

AMSGrad method (Reddi *et al.* 2018) is one of the latest methods developed by Google Inc. This is a variant of the Adam method and it modifies $\hat{S}$ in the Adam method to ensure that S will always be greater than its value in the previous iteration. This allows the step length to stabilize as the number of iterations increases. The formula is given by:

$$m_{k+1} = m_k - \frac{\alpha}{\sqrt{\hat{S}_k} + \eta} \cdot \hat{V}_k, \tag{28}$$

where

$$\hat{S}_k = \max(\hat{S}_{k-1}, S_k), \tag{29}$$

with

$$V_k = \beta_1 V_{k-1} + (1-\beta_1) G(m_k), \tag{30}$$

$$S_k = \beta_2 S_{k-1} + (1-\beta_2)[G(m_k)]^2, \tag{31}$$

where V and S are set to 0 at the beginning. $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\eta = 1 \times 10^{-7}$ are the values used by the Keras library (Chollet 2017). Note that $\hat{S}_{k-1}$ and S_k in eq. (29), are arrays that have the same size as the gradient. Therefore, we choose the maximum between these two arrays with respect to the matricial L_2-norm.

3.7 RAdam

Rectified Adam or RAdam method (Liyuan *et al.* 2019) is a variant of the Adam method. RAdam introduces a rectifying term that allows to stabilize the step-length (learning rate) variance during the first steps of the iterative process. The objective of this strategy is to reduce the variability of the step length during the first iterations of the Adam method. The formula is given by:

$$m_{k+1} = m_k - \frac{\alpha}{\sqrt{\hat{V}_k} + \eta} \cdot r_k \hat{S}_k, \tag{32}$$

where

$$\hat{S}_k = \frac{S_k}{1+\beta_1^k}, \tag{33}$$

$$\hat{V}_k = \begin{cases} \frac{V_k}{1-\beta_2^k}, & \text{if} p_k > 4 \\ 1, & \text{otherwise} \end{cases} \tag{34}$$

$$r_k = \begin{cases} \sqrt{\frac{(p_k-4)(p_k-2)p_\infty}{(p_\infty-4)(p_\infty-2)p_k}}, & \text{if} p_k > 4 \\ 1, & \text{otherwise} \end{cases} \tag{35}$$

$$S_k = \beta_1 S_{k-1} + (1-\beta_1) G(m_k), \tag{36}$$

$$V_k = \beta_2 V_{k-1} + (1-\beta_2)[G(m_k)]^2, \tag{37}$$

$$p_k = p_\infty - \frac{\beta_2^k}{1-\beta_2^k} \cdot 2k, \tag{38}$$

$$p_\infty = \frac{2}{1-\beta_2} - 1, \tag{39}$$

the superindex k, in β_1 and β_2, represents a power that corresponds to the iteration number; V and S are set to 0 at the beginning. $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\eta = 1 \times 10^{-8}$ are variable of the Adam method.

4 A NEW FORMULA TO ASSIGN A STEP LENGTH IN THE AGO METHODS FOR MULTISCALE FWI

The scalars α, β_1 and β_2 (known as hyperparameters in the ML literature) used in the AGO methods, play an important role in the optimization process. These parameters are very sensitive and tuning them is very difficult since they are problem dependent, see Sun *et al.* (2020). A slightly off value selection can drastically alter the convergence's velocity or in the worst case scenario, lead to divergence.

Most libraries that use AGOs, like TensorFlow, Keras, Caffe among others, have pre-set step-length (or learning-rate) values. Most of the deterministic step-length rules for AGO methods apply an exponential decline factor of the form $\alpha(k) \approx \varphi/\sqrt{k}$, where φ is a constant and k corresponds to the iteration number, see Reddi *et al.* (2018). However, these formulae do not consider that the frequency of the data could change dynamically, such as in the multiscaling FWI. In our first experiments with AGOs we observed that the increase of frequency might cause the FWI to diverge.

Effective alternative formulae to compute the step length are the Barzilai-Borwein (BB) methods, see dos Santos & Pestana (2015). Even though these BB methods are designed for gradient-based optimization these do not offer suitable step length for AGOs, since they do not preserve the exponential decay required by AGOs for multiscaling FWI. The ASL method (Pica *et al.* 1990) could provide a better option. Nevertheless, the ASL method does not provide either an exponential decay of the step length. Besides an extra forward simulation (per iteration) is needed to compute it, see Ma *et al.* (2019). Consequently, here we propose a new frequency-dependent formula for the step length, that meet the requirements of multiscaling FWI and has an exponential decay that satisfy the demands of AGOs. The construction of this formula is based on the following steps:

(i) *First step:* assign a step-length to each frequency preserving an exponential decay with respect to the frequency increase using the given relationship:

$$\hat{\alpha}(f) = Q\left(\frac{f_{\max}}{f}\right)^p, \tag{40}$$

where $f_{\max}$ is the maximum frequency of the data to be used during the multiscaling process. Constants $p > 0$ and $Q > 0$ depend on the choice of the AGO method in the FWI. Note that we can set the value for Q by applying eq. (40) to $f_{\max}$, obtaining $\alpha(f_{\max}) = Q$. Once we have the value for Q, we could use $f_{\min}$ to set a value for p. Values of eq. (40) are represented by the stars shown in Fig. 1.

(ii) *Second step:* build a re-parametrization that adds a linear decay (to the step-length) with respect to the number of iterations (to be performed to go form $\hat{\alpha}(f_i)$ to the next $\hat{\alpha}(f_{i+1})$). This reduces the fluctuations in the cost function and avoids high variability when updating the models. To this extent, the step length decreases as the FWI iterations and frequencies increase. This re-parametrization is given by:

$$\alpha(k, f_i) = \begin{cases} \left(\frac{\hat{\alpha}(f_{i+1}) - \hat{\alpha}(f_i)}{n_{f_i} - 1}\right)(k-1) + \hat{\alpha}(f_i), & \text{if } f_i < f_{\max}, \\ \hat{\alpha}(f_{\max}), & \text{if } f_i = f_{\max}, \end{cases} \tag{41}$$

k indicates the kth FWI iteration corresponding to the frequency f_i with $1 \leq k \leq n_{f_i}$. n_{f_i} is the maximum number of FWI iterations for each frequency. Therefore in the k-th FWI iteration that corresponds to the f_i frequency, we will have a specific step-length $\alpha(k, f_i)$. Note that eq. (41) comes from a linear correspondence between the values of eq. (40) and the number of FWI iterations. Values of eq. (41) are represented by the black points shown in Fig. 1. In this way we know the exact value of the step length to be used in each iteration prior to start the FWI, mainly because this step-length rule do not depend on the search direction (which needs to be approximated in each iteration). As a result of this process, we avoid the expensive line search in the conventional FWI's workflow. For each FWI iteration, we then only need one forward simulation to update the model in the right direction. The workflow we use is shown in Fig. 2.

4.1 Empirical criteria to assign values to p and Q

One of the most difficult problems for AGOs is the calibration of hyper-parameters. Consequently, assigning values to p and Q in eq. (40) could be complicated. From trial and error experiments, we found an empirical criteria that allow us to choose suitable values for p and Q obtaining final results that are comparable to the L-BFGS method. The criteria consist of two points:

(i) Use the maximum frequency $f_{\max}$ (knowing that $\alpha(1, f_{\max}) = Q$) to find a value Q such that the first model generated with the maximum frequency, denoted by $m_1(f_{\max})$ satisfies the following condition:

$$A_{\min} \leq \frac{\| m_1(f_{\max}) - m_0 \|}{\max\{\| m_1(f_{\max}) \|, \| m_0 \|\}} \times 100 \text{ per cent } \leq A_{\max}, \tag{42}$$

where m_0 is the initial model to start the FWI and $\|\cdot\|$ is the L_2-norm

(ii) Having fixed Q, use the minimum frequency $f_{\min}$ to find a value for p such that the first model generated with the minimum frequency denoted by $m_1(f_{\min})$ satisfies the following condition:

$$B_{\min} \leq \frac{\| m_1(f_{\min}) - m_0 \|}{\max\{\| m_1(f_{\min}) \|, \| m_0 \|\}} \times 100 \text{ per cent } \leq B_{\max}. \tag{43}$$

The values that we used for $A_{\min}$, $A_{\max}$, $B_{\min}$ and $B_{\max}$ are given in Section 6.

5 FORWARD MODEL AND GRADIENT FOR THE ACOUSTIC CASE

The theory described in the previous sections can be extended to more general multiparametric cases. However, here we only work with 2-D acoustic cases. The forward modelling that correspond to this case is given as follows:

$$\begin{aligned} &\frac{\partial^2 P}{\partial t^2} - v^2 \nabla^2 P = f, \\ &P(X, t=0) = \frac{\partial P}{\partial t}(X, t=0) = 0; f(X_s, t) = w(t)\delta(X - X_s), \end{aligned} \tag{44}$$

where $P(X, t)$ is the pressure field, $f(X_s, t)$ is the source that acts at coordinates X_s with a source-signal waveform $w(t)$, being δ the Dirac delta and $v(X)$ a velocity model. The selection of sources and time-shifting in $f(X_s, t)$ are set and described in Section 2.1.

We use staggered finite differences with a velocity-stress formulation of eq. (44) that can be written as follows:

$$\begin{aligned} \frac{\partial V}{\partial t} &= v^2 \nabla \cdot \vec{\sigma} + f, \\ \frac{\partial \vec{\sigma}}{\partial t} &= \nabla V, \\ V(X, t=0) = \frac{\partial P}{\partial t}(X, t=0) &= 0; f(X_s, t) = w(t)\delta(X - X_s), \end{aligned} \tag{45}$$

where $V = \frac{\partial P}{\partial t}$ and $\vec{\sigma} = \nabla P$.

Therefore, using the adjoint-state method (Plessix 2006) and this formulation, we obtain the gradient for the velocity model (see Appendix B):

$$G(v(X)) = \frac{\partial \epsilon(v)}{\partial v} = \frac{2}{v(X)} \sum_{s=1}^{N_s} \int_0^T \underbrace{\partial V / \partial t}_{D(X,t|X_s)} \underbrace{\check{V}(X, t)}_{U(X,t|X_s)} dt, \tag{46}$$

where V satisfies eq. (45) and $\check{V}$ is the backward-propagated residual velocity wavefield that satisfies:

$$\begin{aligned} \frac{\partial \check{V}}{\partial t} &= v^2 \nabla \cdot \check{\vec{\sigma}} + \sum_{g=1}^{N_g} \Psi(X_g, t|X_s)\delta(X - X_g), \\ \frac{\check{\vec{\sigma}}}{\partial t} &= \nabla \check{V}, \\ \check{V}(X, t=T) = 0; \check{\vec{\sigma}}(X, t=T) &= \vec{0}, \end{aligned} \tag{47}$$

where $\Psi(X_g, t|X_s) = V_{res}(X_g, t|X_s)$ when the gradient of the misfit function based on the L_2-norm (using the least-squares criterion) is computed and $\Psi(X_g, t|X_s) = \frac{V_{res}(X_g,t|X_s)}{|V_{res}(X_g,t|X_s)|}$ when the gradient of the misfit function based on the L_1-norm (using the least-absolute criterion), is computed (see Appendix B).

The next section shows the numerical experiments using these forward and backward simulations applied in the theory shown in the previous sections.

6 NUMERICAL EXPERIMENTS

We perform numerical experiments with forward and backward modelling simulations using a fourth-order staggered finite differences in space; second order in time (Levander 1988) and the simultaneous sources technique described in a previous section. We show results for seven AGO methods based on the L_2-norm. When we apply the FWI based in L_1-norm the results attained by the AdaGrad, RMSProp and Adadelta methods become unstable and the step length becomes hard to calibrate. Therefore for the FWI with the L_1-norm we only

Table 1. Values of constants p and Q used in eq. (40) for each AGO method applied into the FWI based on the L_2-norm for the two different velocity models.

AGO method	p	Q Canadian model	Q Marmousi model
AdaGrad	2	3	2.5
RMSprop	3	0.5	0.4
Adadelta	3	0.001	0.0009
Adam	0.05	4	6
Nadam	0.05	4	8
AMSGrad	0.05	1	2
RAdam	1.1	10	20

Table 2. Values of constants p and Q used in eq. (40) for some AGO methods applied into the FWI based on the L_1-norm for the two different velocity models.

AGO method	p	Q Canadian model	Q Marmousi model
Adam	0.05	4	6
Nadam	0.05	4	7
AMSGrad	0.05	1	1
RAdam	1.1	4	6.7

show results for the most recent AGO methods: Adam, Nadam, AMSGrad and RAdam. The maximum number of activated sources in the supershot for the maximum frequency is $n_s = (1/5)*nx$, being nx the number of gridpoints in the spatial mesh (in the horizontal direction of the velocity model). The sources are located 10 m below the free surface. The source signal is a Ricker wavelet with frequencies specified in each experiment. We use absorbing C-PML (Komatitsch & Martin 2007) boundaries with 25 gridpoints at the boundaries of the mesh. In order to guarantee stability and low numerical dispersion, we set $dx = dz = 10$ m and $dt = 0.001$ s. We fix β, β_1 and β_2 with the suitable values given in Section 3 for each one of the AGO methods considered here. For the multiscaling, we use seven different frequencies within a maximum of 100 iterations of FWI per frequency. To measure the error's evolution in each FWI iteration, we use the function:

$$f(m_k) = \frac{\parallel d_{\text{synt}}(m_k) - d_{\text{obs}} \parallel}{\parallel d_{\text{obs}} \parallel}, \tag{48}$$

being $d_{\text{syn}}(m_k)$ the synthetic seismogram generated with the kth velocity model m_k and d_{obs} is the real observed seismogram. These seismograms are generated with a supershot formed by a set of fixed sources per iteration. The separation between two contiguous sources is 100 m. The observed data come from the Canadian model or the Marmousi model.

Each time that the multiscaling produces a change of frequencies, we set the parameters S, D and V that appear in the AGO methods to zero. α, the step length, is computed as described in eq. (41), following the FWI workflow given in Fig. 2. The only preconditioner we use is the illuminating factor as in, Kaelin & Guitton (2006) given by:

$$\text{Illum}(X) = \int_0^T V^2(X, t)dt. \tag{49}$$

$V(X, t)$ is the velocity wavefield generated with the current supershot and the current velocity model, hence the gradient becomes:

$$G(X) \leftarrow \frac{G(X)}{[\text{Illum}(X) + \epsilon]}. \tag{50}$$

$G(X)$ is the gradient, see eq. (46) and $\epsilon = 1 \times 10^{-20}$ to avoid division by zero.

The decaying rate: the constant p in (40) has shown to be independent of the problem (either the Marmousi or the Canadian velocity models) but it changes depending on the AGO method. However, the constant value of Q in eq. (40) can change depending on the problem. The values for p and Q that we use in this work are given in Tables 1 and 2. These values satisfy the stability criteria given in eqs (42) and (43). For the Canadian model, we use $A_{\min} = 0.05$ per cent, $A_{\max} = 0.1$ per cent, $B_{\min} = 0.1$ per cent and $B_{\max} = 0.3$ per cent. For the Marmousi model, we use $A_{\min} = 0.06$ per cent, $A_{\max} = 0.2$ per cent, $B_{\min} = 0.2$ per cent and $B_{\max} = 0.4$ per cent.

The numerical experiments were performed in two well known open data sets: the Canadian (Gray & Marfurt 1995) and the Marmousi (Martin et al. 2006) velocity models that are publically available.

6.1 L-BFGS optimization method

In order to assess the achievement of AGOs when applied to FWI, we will compare their performance to the one offered by the L-BFGS method (Nocedal & Writght 1999). The L-BFGS is a QN method that approximates the direction $d_k = -H_k^{-1} G_k$ in eq. (5), without computing the full matrix H_k^{-1}. It only requires to store the changes in gradients: $\gamma_k = G_{k+1} - G_k$ and in models: $s_k = m_{k+1} - m_k$, with $r_k = 1/\gamma_k^T s_k$

for the most recent N iterations. Hence, with this information and using the well-known two-loop recursion procedure (shown below) this method can compute an approximation to $d_k = -H_k^{-1} G_k$ as follows:

(i) $q \leftarrow G_k$
(ii) **for** $(i = k-1, ..., k-N)$ **do**
(iii) $\alpha_i \leftarrow r_i s_i^T q$
(iv) $q \leftarrow q - \alpha_i \gamma_i$
(v) **end for**
(vi) $z \leftarrow \frac{s_{k-1}^T \gamma_{k-1}}{\gamma_{k-1}^T \gamma_{k-1}} q$
(vii) **for** $(i = k-N, ..., k-1)$ **do**
(viii) $\beta \leftarrow r_i \gamma_i^T z$
(ix) $z \leftarrow z + s_i(\alpha_i - \beta)$
(x) **end for**
(xi) $d_k = -z$.

When applying L-BFGS to the two velocity models, we consider a limited memory of size $N = 10$. In order to satisfy the Wolfe conditions we use the ASL method, see Pica *et al.* (1990) and Ma *et al.* (2019); this method only requires one extra forward simulation per iteration and it is given by:

$$\alpha = \alpha_t \frac{\sum_s \sum_g [d_{\text{synt}}(m_k + \alpha_t d_k) - d_{\text{synt}}(m_k)]^T [d_{\text{synt}}(m_k + \alpha_t d_k) - d_{\text{obs}}]}{\sum_s \sum_g [d_{\text{synt}}(m_k + \alpha_t d_k) - d_{\text{synt}}(m_k)]^T [d_{\text{synt}}(m_k + \alpha_t d_k) - d_{\text{synt}}(m_k)]}, \tag{51}$$

where α_t is a trial step length. Here, we use, $\alpha_t = \zeta(||m_k||/||d_k||)$ (being ζ a coefficient of proportionality and $||\cdot||$ the vectorial norm L_2). α_t needs to satisfy the following condition:

$$\max(|\alpha_t d_k|) \leq \frac{1}{100} \max(m_k). \tag{52}$$

$d_{\text{synt}}(m_k)$ =approximated data in the model m_k and d_{obs}=observed data. The coefficients s and g refer to the sources and geophones that act in each FWI iteration.

6.2 Canadian overthrust BP velocity model inversion

In our first experiment, we take a sample of the Canadian overthrust BP velocity model (Gray & Marfurt 1995), the grid size is $nz \times nx = 250 \times 556$ (not including the C-PML boundaries). The model is shown in Fig. 3 (left). The initial velocity model corresponds to a flat layer model, computed as the real velocity model's average for each grid along the x-axis as depicted in Fig. 3 (right). The acoustic wavefield is recorded by 556 receivers located 10m below the free surface, for a time $T = 3.5$ s requiring us to execute 3500 time steps. Even though we use a mesh with a dx that satisfies the Courant condition, the scales in Figs 3, 5, 6, 7 and 8 are the same as the scales in Gray & Marfurt (1995) and the model has a length of 3.3 km and a depth of 10 km.

During the multiscaling process we use frequencies: 4, 8, 14, 20, 24, 32 and 40 Hz. Our starting point is the initial velocity model, to which we apply the FWI (according to the workflow in Fig. 2). The normalized error curve, computed using eq. (48), is shown for each FWI iteration in Fig. 4. Firstly, we apply seven AGO methods: AdaGrad, RMSprop, Adadelta, Adam, Nadam, AMSGrad and RAdam to the two velocity models with FWI using L_2-norm. As a result, we obtain the final velocity models for AGOs and the L-BFGS methods, shown in Fig. 5. Secondly, we apply FWI using L_1-norm but only to the most recent AGOs (Adam, Nadam, AMSGrad and RAdam). We obtained the final velocity models, shown in Fig. 6. The differences between the true velocity model and approximated velocity models, retrieved by the FWI (based on the L_2 and L_1 norms) are shown in Figs 7 and 8, respectively . Depth profiles at $x = 820$ m and 2 km are shown in Figs 9 and 10, respectively. Fig. 11 is the comparison between the residual data with the FWI based on the L_2-norm (with L-BFGS, Adadelta, AMSGrad and RAdam) and the ones based on the L_1-norm (with Adam, Nadam, AMSGrad and RAdam); for a source located in the middle of the receivers' axis. We only show these results (for the FWI based on the L_2-norm), since the results for the final velocity models obtained with Adadelta are very similar to those obtained with AdaGrad and RMSProp; the results obtained with AMSGrad are very similar to the results obtained with Adam and Nadam.

6.3 Marmousi 2 velocity model Inversion

We apply the AGO methods to a Marmousi 2 velocity model's sample (Martin *et al.* 2006), size $nz \times nx = 257 \times 522$ (not including the C-PML boundaries), depicted in Fig. 12 (left). Similar to the previous section, we start out with a horizontal flat layered model shown in Fig. 12 (right). The normalized error curve, computed using eq. (48), is presented for each FWI iteration in Fig. 13. The acoustic wavefield is recorded by 522 receivers located 10 m below the free surface, for a period of $T = 7.5$ s completed in 7500 time steps. Even though we use the given mesh with a dx that satisfies the Courant condition, the scales in Figs 12, 14, 15, 16 and 17 are the same as the ones used in Martin *et al.* (2006) and the model has a length of 10 km and a depth of 3.5 km.

Table 3. Number of forward propagations and performance in time per FWI iteration for the L-BFGS and AGOs. The running time is measure using a 2.7 GHz 12-Core Intel Xeon E5 processor machine.

	Models/methods	L-BFGS	AGOs
Forward propagations	Canadian	2	1
	Marmousi	2	1
Time	Canadian	82 s	67 s
	Marmousi	210 s	170 s

During the multiscaling process, we use frequencies: 1.5, 3, 5.25, 7.5, 9, 12 and 15 Hz. Following the same order as the previous example, Figs 14 and 15 show the final results obtained for the FWI based on the L_2- and L_1-norms, respectively. The difference between the true velocity model and some of the approximated velocity models, retrieved by the FWI (based on the L_2- and L_1-norms) are shown in Figs 16 and 17, respectively. Depth profiles at $x = 2.5$ and 5 km are displayed in Figs 18 and 19, respectively. As in the previous model we only use some AGOs to show the comparison between the residual data with the FWI based on the L_2-norm (with L-BFGS, Adadelta, AMSGrad and RAdam) and the ones based on the L_1-norm (with Adam, Nadam, AMSGrad and RAdam); for a source located in the middle of the receivers' axis, as depicted in Fig. 20.

7 RESULTS DISCUSSION

Both numerical experiments are conducted performing 100 FWI iterations per frequency within the multiscaling process with seven different frequencies. On Table 3, we show the performance comparison between the L-BFGS and AGO's methods. The running time is measure using a 2.7 GHz 12-Core Intel Xeon E5 processor machine. From these results we note that the FWI with AGOs is computationally faster than the L-BFGS. The reason is that the step length of the L-BFGS does not satisfy (or hardly satisfies) Wolfe conditions at the first trial. Consequently, one needs a line-search strategy to ensure the convergence, increasing its computational cost. In this case, we use the ASL method which offers us an optimal step length given in eq. (51). With this ASL method, we perform only one extra forward simulation in each FWI iteration. Otherwise the AGO methods (with our new step-length rule) do not perform any extra forward simulation. The line-search process is avoided assigning to each AGO an step length as indicated by eq. (41). Note that the jumps in the error curves (Figs 4 and 13) are due to changes of frequency. This produces an approximation of the velocity model's gross features with lower frequencies. As the frequency increases, the finer components appear and can be recovered.

According to the error curves' evolution for the Canadian model, shown in Fig. 4, the performance of the L-BFGS is similar to the Adam and the Nadam methods, they are able to reduce the error (between the synthetic and observed traces), but not as good as the AMSGrad method does. Since the Marmousi model is more complex than the Canadian model, from the error curves' evolution shown in Fig. 13, we observe that both, the L-BFGS and the Adadelta reduce the error but not as good as the rest of the AGO methods.

On Figs 5 and 14, we observe that the final velocity models obtained with Adam, Nadam, AMSGrad and RAdam have the highest resolution (these correspond to the FWI based on the L_2-norm). While some AGOs (particularly the methods: AdaGrad, RMSProp and Adadelta) are unable to provide good results for FWI based on the L_1-norm, the Adam, Nadam and AMSGrad methods with the L_1-norm overcome the resolution of all the methods when the L_2-norm is applied. Furthermore, a better approximation of deeper zones is shown in Figs 6 and 15, the same result can be observed in the velocity profiles (Figs 9, 10, 18 and 19). This is because the results obtained with the L_1-norm are the ones that have the closest approximation to the real model. Figs 8 and 17, show the improvement that offer some AGOs with the FWI based on the L_1-norm, by reducing the differences (between the real and approximated models) compared to the differences shown in Figs 7 and 16, respectively.

We conclude, from their fast convergence; the error curves' evolution; the final velocity models and the velocity profiles, that the best optimizers for FWI based on the L_2-norm, are: AMSGrad and RAdam. AMSGrad has a high-speed convergence and RAdam has a good stability during the FWI process. In addition, AMSGrad and RAdam results are similar to those obtained with Adam and Nadam. However the best option among all, is to apply FWI based on the L_1-norm using the Adam, Nadam and AMSGrad methods. The results of the L-BFGS algorithm are similar to some of the AGOs but can be overcome by the Adam, Nadam and AMSGrad methods, with the disadvantage that it is computationally more expensive and harder to implement than AGOs. Deeper structures are harder to recover, nevertheless AGOs work better, even so Marmousi velocity model is harder to recover than the Canadian overthrust BP velocity model. It is remarkable that the AGOs final results are very good despite that we have started the inversion process with a simple flat layered velocity model.

In Bollapragada *et al.* (2018), the authors (Nocedal the author of the L-BFGS, among them) claim that the standard L-BFGS is based on gradient approximations that are not dominated by noise. This can be a disadvantage to the use of the simultaneous sources technique, which introduces some amount of noise (crosstalk noise). Moreover, L-BFGS it is not a suitable algorithm to deal with stochastic optimization problems, see Bollapragada *et al.* (2018) and Adolphs *et al.* (2019); while AGOs are. Since the random selection of sources that comes with the dynamic simultaneous sources technique (that we use in this work) is a particular kind of stochastic optimization (Moghaddam *et al.* 2013), the L-BFGS is not considered a good option within this framework. This is the reason why some AGOs show better performance (i.e. better results and faster convergence) than the standard L-BFGS method.

Stochastic descent methods come at the advantage of avoiding the cost for additional simulations in conventional line-search methods. This is a double-edged sword, because the price one needs to pay for avoiding the line-search are diminishing step sizes which could eventually result in a significantly higher number of iterations.

Another important aspect of stochastic descent methods is that conventional step-length algorithms are not directly applicable because of the dynamically changing cost functional resulting from the random combination of sources, which is why the methods presented in Section 3 become important.

8 CONCLUSIONS

We have studied the results of the AGOs applied to simultaneous-source multiscale FWI based on the L_1- and L_2-norms. This leads us to propose a new step-length formula given in eq. (41) that explicitly considers the frequencies used during the multiscaling FWI and avoids the line-search process when applied to AGOs. This allows a computational cost enhancement of the FWI and in some cases we obtain better results than the ones given by the L-BFGS method. This ameliorates the application of QN methods commonly called upon in FWI, as they have a line-searching process. Additionally, we have reduced the crosstalk noise effect applying the dynamic simultaneous-sources method by combining of techniques such as random-in-subgroup shot subsampling; random time-shifting and random polarities. Merging these strategies in a single workflow results in an acceleration of the FWI process, leading to a high-resolution physical parameter inversion with only one forward propagation per FWI iteration. We conclude that the best results are obtained when the Adam, Nadam and AMSGrad methods are applied into the FWI based on the L_1-norm. This above theory can be generalized to multiparametric 3-D FWI and with the use of GPU's, could be further accelerated computationally. We believe that it is possible to generalize the AGO methods presented here and carefully adapt them to work with search directions computed with L-BFGS, but this is a subject for further study.

ACKNOWLEDGEMENTS

We would like to thank two anonymous reviewers that greatly helped to improve the manuscript. We would like to thank comments and suggestions from Jorge O. Parra. We wish to thank Nicolás González Boileau for his kind help to improve the English. This work was partially supported by DGAPA, UNAM project IN107720 and CONACYT México under Ciencia de Frontera project number 6655.

REFERENCES

Adolphs, L., Kohler, J. & Lucchi, A., 2019. Ellipsoidal trust region methods and the marginal value of hessian information for neural network training, preprint (arXiv:1905.09201v1) [cs.LG].

Ben-Hadj-Ali, H., Operto, S. & Virieux, J., 2011. An efficient frequency-domain full waveform inversion method using simultaneous encoded sources, *Geophysics,* **76**(4), R109–R124.

Bollapragada, R., Mudigere, D., Nocedal, J., Shi, H.M. & Tang, P.T.P., 2018. A progressive batching L-BFGS method for machine learning, preprint (arXiv:1802.05374).

Boonyasiriwat, C. & Schuster, G.T., 2010. 3D multisource full-waveform inversion using dynamic random phase encoding, in *SEG Technical Program Expanded Abstracts,* pp. 1044–1049 , doi:10.1190/1.3513025.

Boonyasiriwat, C., Valasek, P., Routh, P., Cao, W., Schuster, G. & Macy, B., 2009. An efficient multiscale method for time-domain waveform tomography, *Geophysics,* **76**(6), WCC59–WCC68.

Brossier, R., Operto, S. & Virieux, J., 2009. Seismic imaging of complex structures by 2D elastic frequency-domain full-waveform inversion, *Geophysics,* **74**(6), WCC63–WCC76.

Brossier, R., Operto, S. & Virieux, J., 2010. Which data residual norm for robust elastic frequency-domain full waveform inversion?, *Geophysics,* **75**(3), R37–R46.

Chollet, F., , 2018. *Keras: The Python Deep Learning Library, Astrophysics Source Code Library, record ascl:1806.022*.

Conn, A.R., Goulf, N.I.M. & Toint, P.L., 2000, *Trust-Region methods,* SIAM Series on Optimization, https://doi.org/10.1137/1.9780898719857, 978-0-89871-460-9.

Crase, E., Pica, A., Noble, M., McDonald, J. & Tarantola, A., 1990. Robust elastic non-linear waveform inversion: Application to real data, *Geophysics,* **55**(5), 527–538.

Datta, D. & Sen, M.K., 2016. Estimating a starting model for full-waveform inversion using a global optimization method, *Geophysics,* **81**(4), R211–R223.

Díaz, E. & Guitton, A., 2011. Fast full waveform inversion with random shot decimation, *SEG Tech. Prog. Expand. Abstr.,* **30**(1), 2804–2808.

dos Santos, A.W.G. & Pestana, R.C., 2015. Time-domain multiscale full-waveform inversion using the rapid expansion method and efficient step-length estimation, *Geophysics,* **80**(4), R203–R216.

Dozat, T., 2016. Incorporating nesterov momentum into Adam, International Conference on Learning Representations, Workshop track - ICLR 2016, pp. 1–4, Puerto Rico, San Juan.

Duchi, J., Hazan, E. & Singer, Y., 2011. Adaptive subgradient methods for online learning and stochastic optimization, *J. Mach. Learn. Res.,* **12,** 2121–2159, https://www.jmlr.org/papers/volume12/duchi11a/duchi11a.pdf.

Dutta, G. & Schuster, G.T., 2014. Attenuation compensation for least-squares reverse time migration using the viscoacoustic-wave equation, *Geophysics,* **79**(6), S251–S262.

Erickson, B.J., Korfiatis, P., Akkus, Z., Kline, T. & Philbrick, K., 2017. Toolkits and libraries for deep learning, *J. Digital Imag.,* **30,** 400–405 , https://link.springer.com/article/10.1007/s10278-017-9965-6.

Gray, S.H. & Marfurt, K., 1995. Migration from topography: improving the near-surface image, *Can. J. Expl. Geophys.,* **31,** https://wiki.seg.org/wiki/1994_BP_migration_from_topography.

Ha, W. & Shin, C., 2013. Efficient Laplace-domain full waveform inversion using a cyclic shot subsampling method, *Geophysics,* **78**(2), R37–R46.

Herrmann, F.J., Erlangga, Y.A. & Lin, T.T., 2009. Compressive simultaneous full-waveform simulation, *Geophysics,* **74**(4), A35–A40.

Jeong, W., Pyun, S., Son, W. & Min, D., 2013. A numerical study of simultaneous-source full waveform inversion with *l*1-norm, *Geophys. J. Int.,* **194**(3), 1727–1737.

Jin, S. & Madariaga, R., 1993. Background velocity inversion with a genetic algorithm, *Geophys. Res. Lett.,* **20**(2), 93–96.

Jing, X., Finn, C.J., Dickens, T.A. & Willen, D.E., 2000. Encoding multiple shot gathers in prestack migration, in *SEG Expanded Abstracts,* pp. 786–790, Tulsa Oklahoma USA , doi:10.1190/1.1816188.

Kaelin, B. & Guitton, A., 2006. Imaging condition for reverse time migration, in *SEG Technical Program Expanded Abstracts,* pp. 2594–2598 , doi:10.1190/1.2370059.

Kingma, D.P. & Ba, J.L., 2015. Adam: a method for stochastic optimization, preprint (arXiv:1412.6980) [cs.LG], 1–13 , https://arxiv.org/abs/1412.6980.

Komatitsch, D. & Martin, R., 2007. An unsplit convolutional perfectly matched layer improved at grazing incidence for the seismic wave equation, *Geophysics,* **72**(5), 155–167.

Krebs, J.R., Anderson, J.E., Hinkley, D., Neelamani, R., Lee, S., Baumstein, A. & Lacasse, M., 2009. Fast full-wavefield seismic inversion using encoded sources, *Geophysics,* **74**(6), WCC177–WCC188.

Levander, A., 1988. Fourth-order finite-differences P-SV seismograms, *Geophysics,* **53,** 1425–1436.

Liyuan, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J. & Han, J., 2019. On the variance of the adaptive learning rate and beyond, preprint (arXiv:1908.03265), https://arxiv.org/abs/1908.03265.

Ma, X., Li, Z., Ke, P., Xu, S., Liang, G. & Wu, X., 2019. Research of step-length estimation methods for full waveform inversion in time domain, *Explor. Geophys.,* **50**(6), 583–599.

Ma, Y. & Hale, D., 2012. Quasi-newton full-waveform inversion with a projected hessian matrix, *Geophysics,* **77**(5), R207–R216.

Martin, G.S., Wiley, R. & Marfurt, K.J., 2006. Marmousi2: an elastic upgrade for marmousi, *Leading Edge,* **25**(2), 156–166.

Métivier, L., Brossier, R., Operto, S. & Virieux, J., 2013. Full waveform inversion and the truncated Newton method, *J. Sci. Computi. Soc. Indus. Appl. Math.,* **59**(1), 153–195.

Métivier, L., Brossier, R., Operto, S. & Virieux, J., 2017. Full waveform inversion and the truncated Newton method, *SIAM Rev.,* **59**(1), 153–195.

Moghaddam, P.P., Keers, H., Herrmann, F.J. & Mulder, W.A., 2013. A new optimization approach for source-encoding full-waveform inversion, *Geophysics,* **73**(3), R125–R132.

Mulder, W., Steenweg, R. & Roos, C., 2006. Nesterov's method and L-BFGS minimisation applied to acoustic migration, in *European Association of Geoscientists & Engineers (EAGE) 68th Conference and Exhibition* , doi:10.3997/2214-4609.201402426.

Nesterov, Y.E., 1983. A method for unconstrained convex minimization problem with the rate of convergence $\mathcal{O}(1/k^2)$, *Dokl. ANSSSR (translated as Soviet. Math. Docl.),* **269,** 543–547.

Nocedal, J. & Writght, S., 1999. *Numerical Optimization,* Springer Verlag, New York.

Pica, A., Diet, J.P. & Tarantola, A., 1990. Nonlinear inversion of seismic reflection data in a laterally invariant medium, *Geophysics,* **55**(3), 284–292.

Plessix, R.E., 2006. A review of the adjoint-state method for computing the gradient of a functional with geophysical applications, *Geophys. J. Int.,* **167**(2), 495–503.

Polyak, B., 1964. Some methods of speeding up the convergence of iteration methods, *USSR Comput. Math. Math. Phys.,* **4**(5), 1–17.

Reddi, S.J., Kale, S. & Kumar, S., 2018. On the convergence of Adam and beyond, preprint (arXiv:1904.09237), https://arxiv.org/abs/1904.09237.

Richardson, A., 2018. Seismic full-waveform inversion using deep learning tools and techniques, pp. 1–17, preprint (arXiv:1801.07232), https://arxiv.org/abs/1801.07232.

Romero, L.A., Ghiglia, D.C., Ober, C.C. & Morton, S.A., 2000. Phase encoding of shot records in prestack migration, *Geophysics,* **65**(2), 426–436.

Ruder, S., 2016. An overview of gradient descent optimization algorithms, pp. 1–14, preprint (arXiv:1609.04747) https://arxiv.org/abs/1609.04747.

Sambridge, M. & Mosegaard, K., 2002. Monte Carlo methods in geophysical inverse problems, *Rev. Geophys.,* **40**(3), 3–10-3-29.

Schuster, G., 2017. *Seismic Inversion,* Society of Exploration Geophysicists, Tulsa OK, USA.

Schuster, G.T., Wang, X., Huang, Y., Dai, W. & Boonyasiriwat, C., 2011. Theory of multisource crosstalk reduction by phase-encoded statics, *Geophys. J. Int.,* **184**(3), 1298–1303.

Shi, C.W. & He, B.S., 2018. Multiscale full-waveform inversion based on shot subsampling, *Appl. Geophys.,* **15,** 261–270.

Sun, B. & Alkhalifah, T., 2019. ML-descent: an optimization algorithm for FWI using machine learning, in *SEG Technical Program Expanded Abstracts,* pp. 2288–2291 , doi:10.1190/segam2019-3215304.1.

Sun, J., Niu, Z., Innanen, K.A., Li, J. & Trad, D.O., 2020. A theory-guided deep-learning formulation and optimization of seismic waveform inversion, *Geophysics,* **85**(2), R87–R99.

Tarantola, A., 1984. Linearized inversion of seismic reflection data, *Geophys. Prospect.,* **32**(6), 998–1015.

Tarantola, A., 1987. *Inverse Problem Theory: Methods for Data Fitting and Model Parameter Estimation,* Elsevier, New York.

Tarantola, A., 2005. *Inverse Problem Theory and Methods for Model Parameter Estimation,* SIAM, Philadelphia, 1st edn..

Tieleman, T. & Hinton, G., 2012, Lecture 6.5-rmsprop: Divide the Gradient by a Running Average of its Recent Magnitude, *COURSERA: Neural networks for machine learning,* **4,** 2, 26–31. Available at: https://www.cs.toronto.edu/tijmen/csc321/slides/lecture_slides_lec6.pdf.

van Herwaarden, D.P., Boehm, C., Afanasiev, M., Thrastarson, S., Krischer, L., Trampert, T. & Fichtner, A., 2020. Accelerated full-waveform inversion using dynamic mini-batches, *Geophys. J. Int.,* **221**(2), 1427–1438.

Virieux, J. & Operto, S., 2009. An overview of full waveform inversion in exploration geophysics, *Geophysics,* **76**(4), WCC1–WCC26.

Virieux, J., Asnaashari, A., Brossier, R., Métivier, L., Ribodetti, A. & Zhou, W., 2014. An introduction to full waveform inversion, *Geophys. Ref. Ser.,* R1–1-R1-40 , doi:10.1190/1.9781560803027.entry6.

Wolfe, P., 1969. Convergence conditions for ascent methods, *SIAM Rev.,* **11**(2).

Wolfe, P., 1971. Convergence conditions for ascent methods. II: some corrections, *SIAM Rev.,* **13**(2), 226–235, https://doi.org/10.1137/1013035.

Yang, F. & Ma, J., 2019. Deep-learning inversion: a next generation seismic velocity-model building method, *Geophysics,* **84**(4), R583–R599.

Yuan, Y.-X., 1999. A Review of trust region algorithms for optimization, in *ICIAM 99: Proceedings of the Fourth International Congress on Industrial and Applied Mathematics,* Vol. **99**(1), pp. 271–282.

Zeiler , Matthew D., 2012. ADADELTA: An Adaptive Learning Rate Method, *arXiv:1212.5701v1,* https://arxiv.org/abs/1212.5701.

Zhan, G., Dai, W. & Schuster, G.T., 2013. Acoustic multisource waveform inversion with deblurring, *J. Seism. Explor.,* **22**(5), 477–488 , doi:10.3997/2214-4609.201400800.

APPENDIX A: WHAT ARE THE ADAPTIVE GRADIENT OPTIMIZATION METHODS

The bedrock of successful ML approaches, are the optimization techniques used during training. Currently, the optimizers for ML problems are the AGO methods. AGOs are variants of SGD methods. AGOs are also known as Adaptive Learning Rate (ALR) methods since in the ML literature, the step-length parameter is also called the learning rate. AGOs warrantee the convergence to global minimum when minimizing convex misfit functionals (Ruder 2016). However, empirically it has been found that some AGOs (e.g. the Adam method) often outperform problems with non-convex objective functionals, see Kingma & Ba (2015). Adolphs *et al.* (2019) proved that some AGOs (particularly the AdaGrad, RMSProp and Adam methods) can be seen as a first-order trust region methods with ellipsoidal constraints which outperforms its spherical counterpart.

In general, AGO methods use gradient updates scaled by square roots of exponential moving average of squared past gradients, what allows them to automatically adapt the step length according to evolution of the optimization process. The fact that the step length is able to adapt in different ways, gives rise to a family of different AGOs, see Reddi *et al.* (2018).

AGOs are the optimizers at the core of powerful ML packages such as TensorFlow, Keras and Caffe (Erickson *et al.* 2017). Some of these methods became popular within the Deep Learning community where their strengths and weaknesses have been largely discussed, see Ruder (2016).

APPENDIX B: DERIVATION OF THE GRADIENT USING THE ADJOINT-STATE METHOD

Let us consider the velocity-stress formulation given in eq. (45), this can be written in matrix form as follows:

$$\underbrace{\begin{bmatrix} \frac{\partial}{\partial t} & -\nu^2\frac{\partial}{\partial x} & -\nu^2\frac{\partial}{\partial z} \\ -\frac{\partial}{\partial x} & \frac{\partial}{\partial t} & 0 \\ -\frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial t} \end{bmatrix}}_{A(\vec{m})} \underbrace{\begin{bmatrix} V \\ \sigma_{xx} \\ \sigma_{zz} \end{bmatrix}}_{W(\vec{m})} = \underbrace{\begin{bmatrix} f \\ 0 \\ 0 \end{bmatrix}}_{\text{source-term}}, \tag{53}$$

where W are the approximated (or synthetic) seismograms recorded at the free surface with $\vec{m} = [m_1, ..., m_n]^T$, being m_i a physical parameter (e.g. velocity and density). The observed (or acquired) seismic data is d_{obs}, then the residual is given by $\Delta d = W(\vec{m}) - d_{\text{obs}}$. For simplicity, we write the cost function (based on the L_2-norm) as follows:

$$\begin{aligned} \epsilon(\vec{m}) &= \frac{1}{2}\int_0^T \parallel \Delta d \parallel^2 dt, \\ &= \frac{1}{2}\int_0^T (W(\vec{m}) - d_{\text{obs}}, W(\vec{m}) - d_{\text{obs}})dt, \end{aligned} \tag{54}$$

where $(\cdot, \cdot)$ is the interior dot product. Then, the gradient component of the misfit function with respect to m_i is given by:

$$\frac{\partial \epsilon(\vec{m})}{\partial m_i} = \int_0^T \left(\frac{\partial W(\vec{m})}{\partial m_i}, W(\vec{m}) - d_{\text{obs}} \right) dt, \tag{55}$$

where $\frac{\partial W(\vec{m})}{\partial m_i}$ is Fréchet derivative which we approximate with the *adjoint-state method* (Plessix 2006).

According to Dutta & Schuster (2014), we can acquire the Fréchet derivative, by taking the derivative of eq. (53) with respect to the model m_i to obtain:

$$A(\vec{m})\frac{\partial W(\vec{m})}{\partial m_i} + \frac{\partial A(\vec{m})}{\partial m_i} W(\vec{m}) = 0, \tag{56}$$

hence

$$\frac{\partial W(\vec{m})}{\partial m_i} = -A(\vec{m})^{-1}\frac{\partial A(\vec{m})}{\partial m_i} W(\vec{m}). \tag{57}$$

Substituting eq. (57) into eq. (55), we obtain:

$$\frac{\partial \epsilon(\vec{m})}{\partial m_i} = -\int_0^T \left(A(\vec{m})^{-1}\frac{\partial A(\vec{m})}{\partial m_i} W(\vec{m}), \underbrace{W(\vec{m}) - d_{\text{obs}}}_{\Delta d} \right) dt, \tag{58}$$

$$= -\int_0^T \left(\frac{\partial A(\vec{m})}{\partial m_i} W(\vec{m}), \{A(\vec{m})^{-1}\}^\dagger \Delta d \right) dt. \tag{59}$$

According to Schuster (2017): $\{A(\vec{m})^{-1}\}^\dagger \Delta d = [\check{V}, \check{\sigma}_{xx}, \check{\sigma}_{zz}]^T$ corresponds to the backward-propagated residual wavefields that satisfy eq. (47).

Since the only model to update in eq. (53) is the velocity $\vec{m} = [\nu(X)]$, then

$$\frac{\partial A(\nu)}{\partial \nu} = \begin{bmatrix} 0 & -2\nu\frac{\partial}{\partial x} & -2\nu\frac{\partial}{\partial z} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{60}$$

therefore

$$\frac{\partial A(\nu)}{\partial \nu} W(\nu) = \begin{bmatrix} -2\nu\left(\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{zz}}{\partial z}\right) \\ 0 \\ 0 \end{bmatrix}. \tag{61}$$

Substituting eq. (61) into eq. (59), we obtain:

$$\frac{\partial \epsilon(\nu)}{\partial \nu} = 2\nu \int_0^T (\nabla \cdot \vec{\sigma})\, \check{V} dt. \tag{62}$$

To eliminate spatial derivatives in $\nabla \cdot \vec{\sigma}$, we use its definition:

$$\nabla \cdot \vec{\sigma} = \frac{1}{v^2} \frac{\partial V}{\partial t}, \tag{63}$$

then the gradient with respect to the velocities $v(X)$ is given by:

$$\frac{\partial \epsilon(v)}{\partial v} = \frac{2}{v(X)} \int_0^T \frac{\partial V}{\partial t} \check{V} dt. \tag{64}$$

where $V(X, t)$ satisfies the forward model given in eq. (53) and $\check{V}(X, t)$ satisfies the backward model given in eq. (47) with $\Psi = \Delta d$. The computation of the gradient for the misfit function based on the L_1-norm is similar but with the difference that $\check{V}(X, t)$ must satisfy eq. (47) with $\Psi = \frac{\Delta d}{|\Delta d|}$.

Bibliografía

Adolphs, L., Kohler, J., and Lucchi, A. (2019). Ellipsoidal Trust Region Methods and the Marginal Value of Hessian Information for Neural Network Training. *arXiv:1905.09201v1 [cs.LG]*. https://arxiv.org/abs/1905.09201v1.

Aki, K. and Richards, P. (2009). Quantitative Seismology, Second Edition. *University Science Book*.

Baysal, E., Kosloff, D., and Sherwood, J. (1983). Reverse Time Migration. *Geophysics*, 48(4):1514–1524.

Ben-Hadj-Ali, H., Operto, S., and Virieux, J. (2011). An efficient frequency-domain full waveform inversion method using simultaneous encoded sources. *Geophysics*, 76(4):R109–R124. https://doi.org/10.1190/1.3581357.

Berenger, J.-P. (1994). A Perfectly Matched Layer for the Absorption of Electromagnetic Waves. *Journal of Computational Physics*, 114(2):185–200. https://doi.org/10.1006/jcph.1994.1159.

Bernal-Romero, M. and Iturrarán-Viveros, U. (2020). Accelerating full-waveform inversion through adaptive gradient optimization methods and dynamic simultaneous sources. *Geophysical Journal International*, 225(1):97–126. https://doi.org/10.1093/gji/ggaa583.

Bollapragada, R., Mudigere, D., Nocedal, J., Shi, H. M., and Tang, P. T. P. (2018). A Progressive Batching L-BFGS Method for Machine Learning. *arXiv:1802.05374*. https://arxiv.org/abs/1802.05374.

Boonyasiriwat, C. and Schuster, G. T. (2010). 3D Multisource Full-Waveform Inversion using Dynamic Random Phase Encoding. *SEG Technical Program Expanded Abstracts 2010*, pages 1044–1049. doi: 10.1190/1.3513025.

Boonyasiriwat, C., Valasek, P., Routh, P., Cao, W., Schuster, G., and Macy, B. (2009). An efficient multiscale method for time-domain waveform tomography. *Geophysics*, 76(6):WCC59–WCC68. doi: 10.1190/1.3151869.

Brossier, R., Operto, S., and Virieux, J. (2010). Which data residual norm for robust elastic frequency-domain full waveform inversion? *Geophysics*, 75(3):R37–R46. https://doi.org/10.1190/1.3379323.

Bunks, C., Saleck, F., Zaleski, S., and Chavent, G. (1995). Multiscale seismic waveform inversion. *Geophysics*, 60(5):1942–2156. http://dx.doi.org/10.1190/1.1443880.

Chollet, F. (2017). *Keras: The Python Deep Learning library*.

Courant, R., Friedrichs, K., and Lewy, H. (1928). Über die partiellen differenzengleichungen der mathemathischen physik. *Mathematische Annalen*, 100:32–74.

Courant, R., Friedrichs, K., and Lewy, H. (1967). On the partial difference equations of mathematical physics. *IBM journal*, pages 215–234.

Crase, E., Pica, A., Noble, M., McDonald, J., and Tarantola, A. (1990). Robust elastic non-linear waveform inversion: Application to real data. *Geophysics*, 55(5):527–538.

Díaz, E. and Guitton, A. (2011). Fast full waveform inversion with random shot decimation. *SEG Technical Program Expanded Abstracts*, 30(1):2804–2808. https://doi.org/10.1190/1.3627777.

dos Santos, A. W. G. and Pestana, R. C. (2015). Time-domain multiscale full-waveform inversion using the rapid expansion method and efficient step-length estimation. *Geophysics*, 80(4):R203–R216. https://doi.org/10.1190/geo2014-0338.1.

Dozat, T. (2016). Incorporating Nesterov Momentum into Adam. *Fourth International Conference on Learning Representations*, pages 1–4. Puerto Rico, San Juan. May 2-4.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, 12:2121–2159. https://www.jmlr.org/papers/volume12/duchi11a/duchi11a.pdf.

Dutta, G. and Schuster, G. T. (2014). Attenuation compensation for least-squares reverse time migration using the viscoacoustic-wave equation. *Geophysics*, 79(6):S251–S262. https://doi.org/10.1190/geo2013-0414.1.

Erickson, B. J., Korfiatis, P., Akkus, Z., Kline, T., and Philbrick, K. (2017). Toolkits and Libraries for Deep Learning. *Journal of Digital Imaging*, 30:400–405. https://link.springer.com/article/10.1007/s10278-017-9965-6.

Gray, S. H. and Marfurt, K. (1995). Migration from topography: Improving the near-surface image. *Canadian Journal of Exploration*, 30(1).

Ha, W. and Shin, C. (2013). Efficient Laplace-domain full waveform inversion using a cyclic shot subsampling method. *Geophysics*, 78(2):R37–R46. https://doi.org/10.1190/geo2012-0161.1.

Herrmann, F. J., Erlangga, Y. A., and Lin, T. T. (2009). Compressive simultaneous full-waveform simulation. *Geophysics*, 74(4):A35–A40. https://doi.org/10.1190/1.3115122.

Jeong, W., Pyun, S., Son, W., and Min, D. (2013). A numerical study of simultaneous-source full waveform inversion with $l1$-norm. *Geophys. J. Int.*, 194(3):1727–1737. https://doi.org/10.1093/gji/ggt182.

Jing, X., Finn, C. J., Dickens, T. A., and Willen, D. E. (2000). Encoding multiple shot gathers in prestack migration. *SEG Expanded Abstracts*, pages 786–790. Tulsa Oklahoma USA. https://doi.org/10.1190/1.1816188.

Kaelin, B. and Guitton, A. (2006). Imaging condition for reverse time migration. *SEG Technical Program Expanded Abstracts*, pages 2594–2598. https://doi.org/10.1190/1.2370059.

Kausel, E. (2006). Fundamental Solutions in Elastodynamics. A Compendium. *Cambridge University Press*.

Kelley, C. T. (1999). Iterative Methods for Optimization: Frontiers in Applied Mathemathics. *SIAM Press, Philadelphia*.

Kingma, D. P. and Ba, J. L. (2015). Adam: a Method for Stochastic Optimization. *arXiv:1412.6980 [cs.LG]*, pages 1–13. https://arxiv.org/abs/1412.6980.

Komatitsch, D. and Martin, R. (2007). An unsplit convolutional perfectly matched layer improved at grazing incidence for the seismic wave equation. *Geophysics*, 72(5):155–167. https://doi.org/10.1190/1.2757586.

Krebs, J. R., Anderson, J. E., Hinkley, D., Neelamani, R., Lee, S., Baumstein, A., and Lacasse, M. (2009). Fast full-wavefield seismic inversion using encoded sources. *Geophysics*, 74(6):WCC177–WCC188. https://doi.org/10.1190/1.3230502.

Levander, A. (1988). Fourth-order finite-differences P-SV seismograms. *Geophysics*, 53:1425–1436. https://doi.org/10.1190/1.1442422.

Liyuan, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., and Han, J. (2019). On the Variance of the Adaptive Learning Rate and Beyond. *arXiv preprint arXiv:1908.03265*. https://arxiv.org/abs/1908.03265.

Ma, X., Li, Z., Ke, P., Xu, S., Liang, G., and Wu, X. (2019). Research of step-length estimation methods for full waveform inversion in time domain. *Exploration Geophysics*, 50(6):583–599. https://doi.org/10.1080/08123985.2019.1641266.

Martin, G. S., Wiley, R., and Marfurt, K. J. (2006). Marmousi2: An elastic upgrade for Marmousi. *The Leading Edge*, 25(2):156–166. https://doi.org/10.1190/1.2172306.

McMechan, G. A. (1983). Migration by extrapolation of time-dependent boundary values. *Geophys. Prosp.*, 31:413–420. https://doi.org/10.1111/j.1365-2478.1983.tb01060.x.

Moczo, P. (1998). Introduction to modeling seismic wave propagation by the finite-difference methods. Disaster Prevention Research Institute, Kyoto University.

Moghaddam, P. P., Keers, H., Herrmann, F. J., and Mulder, W. A. (2013). A new optimization approach for source-encoding full-waveform inversion. *Geophysics*, 73(3):R125–R132. https://doi.org/10.1190/geo2012-0090.1.

Nemeth, T., Chengjun, W., and Schuster, G. T. (1999). Least Squares Migration of incomplete data. *Geophysics*, 64:208–221.

Nesterov, Y. E. (1983). A method for unconstrained convex minimization problem with the rate of convergence $\mathcal{O}(1/k^2)$. *Doklady ANSSSR (translated as Soviet. Math. Docl.)*, 269:543–547.

Nocedal, J. and Nash, S. G. (1991). A numerical study of the limited memory bfgs method and truncated newton method for large scale optimization. *SIAM Journal on Optimization*, 1:358–372.

Nocedal, J. and Writght, S. (1999). *Numerical Optmization*. Springer Verlag, New York.

Operto, S., Gholami, Y., Prieux, V., Ribodetti, A., Métivier, L., Brossier, R., and Virieux, J. (2013). A guided tour of multiparameter full-waveform inversion with multicomponent data: From theory to practice. *The leading edge*, 32:p.1040–1054.

Pica, A., Diet, J. P., and Tarantola, A. (1990). Nonlinear inversion of seismic reflection data in a laterally invariant medium. *Geophysics*, 55(3):284–292. https://doi.org/10.1190/1.1442836.

Plessix, R. E. (2006). A review of the adjoint-state method for computing the gradient of a functional with geophysical applications. *Geophys. J. Int.*, 167(2):495–503. https://doi.org/10.1111/j.1365-246X.2006.02978.x.

Polyak, B. (1964). Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17. https://doi.org/10.1016/0041-5553(64)90137-5.

Reddi, S. J., Kale, S., and Kumar, S. (2018). On the convergence of Adam and beyond. *arXiv:1904.09237*. https://arxiv.org/abs/1904.09237.

Richardson, A. (2018). Seismic full-waveform inversion using deep learning tools and techniques. *arXiv preprint arXiv:1801.07232*, pages 1–17. https://arxiv.org/abs/1801.07232.

Romero, L. A., Ghiglia, D. C., Ober, C. C., and Morton, S. A. (2000). Phase encoding of shot records in prestack migration. *Geophysics*, 65(2):426–436. https://doi.org/10.1190/1.1444737.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv:1609.04747*, pages 1–14. https://arxiv.org/abs/1609.04747.

Schuster, G. (2017). *Seismic Inversion*. Society of Exploration Geophysicists, Tulsa OK, USA.

Schuster, G. T., Wang, X., Huang, Y., Dai, W., and Boonyasiriwat, C. (2011). Theory of multisource crosstalk reduction by phase-encoded statics. *Geophys. J. Int.*, 184(3):1298–1303. https://doi.org/10.1111/j.1365-246X.2010.04906.x.

Shi, C. W. and He, B. S. (2018). Multiscale full-waveform inversion based on shot subsampling. *Applied Geophysics*, 15:261–270. https://doi.org/10.1007/s11770-018-0669-6.

Stein, S. and Wysession, M. (2003). An Introduction to Seismology, Earthquakes, and Earth Structure. *Blackwell Publishing Ltd*.

Sun, J., Niu, Z., Innanen, K. A., Li, J., and Trad, D. O. (2020). A theory-guided deep-learning formulation and optimization of seismic waveform inversion. *Geophysics*, 85(2):R87–R99. https://doi.org/10.1190/geo2019-0138.1.

Tarantola, A. (1984). Linearized inversion of seismic reflection data. *Geophysical Prospecting*, 32(6):998–1015. https://doi.org/10.1111/j.1365-2478.1984.tb00751.x.

Tarantola, A. (1986). A strategy for nonlinear elastic inversion of seismic reflection data. *Geophysics*, 51(10):1893–1903. https://doi.org/10.1190/1.1442046.

Tarantola, A. (1987). *Inverse Problem Theory: Methods for Data Fittingand Model Parameter Estimation*. Elsevier, New York.

Tarantola, A. (2005). *Inverse Problem Theory and Methods for Model Parameter Estimation*. SIAM, 1st edn., Philadelphia. https://doi.org/10.1137/1.9780898717921.

Tieleman, T. and Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf.

Treitel, S. and Lines, L. (2001). Past, present, and future of geophysical inversion - A new millennium analysis. *Geophysics*, 66(1):21–24. https://doi.org/10.1190/1.1444898.

Virieux, J. (1986). P-SV wave propagation in heterogeneous media: velocity-stress finite-difference method. *Geophysics*, 51(4):889–901. https://doi.org/10.1190/1.1442147.

Virieux, J. and Operto, S. (2009). An overview of full waveform inversion in exploration geophysics. *Geophysics*, 76(4):WCC1–WCC26. https://doi.org/10.1190/1.3238367.

Whitmore, N. D. (1983). Iterative depth migration by backward time propagation: 53th annual international meeting. *SEG, Expanded Abstracts*, pages 382–385. https://doi.org/10.1190/1.1893867.

Wolfe, P. (1969). Convergence conditions for ascent methods. *SIAM Review*, 11(2).

Wolfe, P. (1971). Convergence conditions for ascent methods. ii: Some corrections. *SIAM Review*, 13(2).

Yang, F. and Ma, J. (2019). Deep-learning inversion: a next generation seismic velocity-model building method. *Geophysics*, 84(4):R583–R599. https://doi.org/10.1190/geo2018-0249.1.

Zeiler, M. D. (2012). Adadelta: An adaptive learning rate method. *arXiv:1212.5701v1*. https://arxiv.org/abs/1212.5701.

Zhan, G., Dai, W., and Schuster, G. T. (2013). Acoustic multisource waveform inversion with deblurring. *Journal of Seismic Exploration*, 22(5):477–488. DOI: 10.3997/2214-4609.201400800.