


**EVALUACIÓN DEL DESEMPEÑO DEL MÉTODO DE MARCHA RÁPIDA
DESORDENADA APLICADO EN LA CONSTRUCCIÓN DE UNA IMAGEN
SÍSMICA A PARTIR DE DATOS SINTÉTICOS 1C.**

EDWIN MAURICIO VARGAS DÍAZ

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO-MECANICAS
ESCUELA DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA Y DE
TELECOMUNICACIONES**

Bucaramanga

2016



**EVALUACIÓN DEL DESEMPEÑO DEL MÉTODO DE MARCHA RÁPIDA
DESORDENADA APLICADO EN LA CONSTRUCCIÓN DE UN MODELO DE
VELOCIDADES SÍSMICAS A PARTIR DE DATOS SINTÉTICOS 1C.**

EDWIN MAURICIO VARGAS DÍAZ

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

**Director
PhD. César Antonio Duarte Gualdrón**

**Co-Director
MIE. Carlos Andrés Niño Niño**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO-MECANICAS
ESCUELA DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA Y DE
TELECOMUNICACIONES
Bucaramanga**

2016

AGRADECIMIENTOS

A Dios, porque aunque solo lo tengo en cuenta para favores, siempre me ha demostrado que está dándome una mano.

A mi madre Elsa, por su infinito ayuda y amor en cada momento de esta etapa. A mi madre y hermanas, Diana y Leidy, por ser las personas que más quiero y que por lo tanto me motivan cada día.

A Paola Riaño, por ser una gran compañía en esta última etapa y demostrarme que puedo contar con ella en cualquier momento.

Finalmente, a Felipe y Yesid, por ser los mejores compañeros y porque aportaron inmensamente a mi formación académica.

CONTENIDO

	pág.
INTRODUCCIÓN	13
1. MARCO TEÓRICO.....	17
1.1. MÉTODO DE MARCHA RÁPIDA.....	17
1.2. MÉTODO DE MARCHA RÁPIDA DESORDENADA	20
1.3. CONVERSION TIEMPO A PROFUNDIDAD	22
1.3.1. PROBLEMA DEL VALOR DE FRONTERA.....	23
2. IMPLEMENTACIÓN NÚMÉRICA	25
2.1. PILA BINARIA MÍNIMA	25
2.2. PILA DE PRIORIDAD DESORDENADA	25
3. RESULTADOS EXPERIMENTALES.....	27
3.1. MÉTODO DE MARCHA RÁPIDA.....	27
3.2. ALGORITMO DE CONVERSIÓN TIEMPO A PROFUNDIDAD.....	31
3.2.1. Experimento 1.	32
3.2.2. Experimento 2.	35
3.2.3. Experimento 3.	38
3.3. DISCUSIÓN	39

4. CONCLUSIONES.....	41
REFERENCIAS	42
BIBLIOGRAFÍA.....	46

LISTA DE FIGURAS

Figura 1. Progreso del método de marcha rápida. Los puntos negros son los conocidos, los grises el <i>frente de onda</i> y los blancos los desconocidos. Fuente: Tomado de [9].....	19
Figura 2. Estructura pila mínima. Fuente: Tomado de [9].....	20
Figura 3. Estructura de datos para el método de marcha rápida desordenada. $M = 8$ y $\delta = 1$ Fuente: Tomada de [22]	21
Figura 4. Relación entre coordenadas de tiempo y coordenadas de profundidad. Fuente: Tomado de [7].....	23
Figura 5. Pila binaria mínima y su representación en una simple lista. Fuente: Tomado de [24].....	26
Figura 6. Pila de prioridad desordenada implementada con MATLAB.....	26
Figura 7. Modelo de velocidad del medio	28
Figura 8. Tiempos de propagación calculados por el algoritmo de marcha rápida.	29
Figura 9. Tiempos de propagación calculados por el algoritmo de marcha rápida desordenada para diferentes valores de M (a) $M = 2$. (b) $M = 10$. (c) $M = 50$. (d) $M = 100$. (e) $M = 500$. (f) $M = 1000$. (g) $M = 10000$	29
Figura 10. Modelo de velocidad de entrada para el experimento 1.	32
Figura 11. Resultados analíticos de la conversión tiempo a profundidad. (a) $v(z, x)$ (km/s). (b) $t_0(z, x)$ (s). (c) $x_0(z, x)$ (km).....	33

Figura 12. Resultados experimentales. $v(z, x)$ (arriba), $t_0(z, x)$ (medio) y $x_0(z, x)$ (abajo), del algoritmo de conversión tiempo a profundidad usando (a) método de marcha rápida (b) método de marcha rápida desordenada	34
Figura 13. Error relativo velocidades de salida de los algoritmos de conversión utilizando (a) marcha rápida. (b) marcha rápida desordenada.	34
Figura 14. Modelo de velocidad de entrada para el experimento 2.	36
Figura 15. Velocidades de salida. (a) analítica. (b) marcha rápida. (c) marcha rápida desordenada.	37
Figura 16. Error relativo velocidades de salida de los algoritmos de conversión utilizando (a) marcha rápida. (b) marcha rápida desordenada.	37
Figura 17. Tiempos de ejecución del algoritmo de conversión con marcha rápida y marcha rápida desordenada en función del número de puntos del dominio N	38

LISTA DE TABLAS

Tabla 1. Complejidad computacional de las operaciones en la pila binaria. n es el número de puntos en la estructura.	20
Tabla 2. Tiempo de ejecución y error relativo del método de marcha rápida para diferentes valores del parámetro M	30
Tabla 3. Tiempo de ejecución, error relativo máximo y error relativo promedio del algoritmo de conversión utilizando los diferentes métodos de marcha rápida.	35
Tabla 4. Tiempo de ejecución, error relativo máximo y error relativo promedio del algoritmo de conversión en el medio 2.	37
Tabla 5. Tiempo de ejecución, error relativo máximo y error relativo promedio del algoritmo de conversión en el medio 1 con diferentes valores de N	38

RESUMEN

Título:

Evaluación del desempeño del método de Marcha Rápida desordenada aplicado en la construcción de un modelo de velocidades sísmicas a partir de datos sintéticos 1C.¹

Autor: Edwin Mauricio Vargas Díaz²

Palabras Claves: Sísmica, migración, Método de Marcha Rápida, pila de prioridad.

Las imágenes sísmicas son una representación del interior de la tierra que permiten identificar objetivos de exploración en la industria del petróleo, generalmente, estas imágenes se representan en dos sistemas de coordenadas: coordenadas de tiempo y profundidad. Las imágenes en tiempo son adecuadas para áreas donde la velocidad sísmica depende principalmente de la profundidad y las imágenes en profundidad para áreas estructuralmente complejas o de velocidad sísmicas arbitraria. Este trabajo se basa en un algoritmo de conversión tiempo a profundidad propuesto por Maria Kourkina Cameron que tiene como objetivo convertir imágenes sísmicas representadas en coordenadas de tiempo a coordenadas de profundidad; el principal bloque de construcción de este algoritmo de conversión es el método de marcha rápida propuesto por James Sethian. El objetivo de este trabajo es implementar y evaluar el desempeño del algoritmo de conversión tiempo a profundidad construido con una eficiente versión del método de marcha rápida basada en una estructura de datos denominada pila de prioridad desordenada. Para la evaluación, se tienen en cuenta dos criterios, tiempo de ejecución del algoritmo y precisión de la imagen obtenida. Los resultados demuestran que el algoritmo de conversión implementado con el método de marcha rápida desordenada mejora el tiempo de ejecución con una pérdida despreciable de precisión de la imagen resultante.

¹ Trabajo de Grado modalidad en investigación

² Facultad de Ingenierías Físico Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones. Director: PhD. César Antonio Duarte Gualdrón. Codirector: MIE. Carlos Andrés Niño Niño.

ABSTRACT

TITLE:

Performance evaluation of untidy fast marching method applied in the seismic image building with synthetics data 1C.³

AUTHOR: Edwin Mauricio Vargas Díaz⁴

KEYWORDS: Seismic, migration, Fast marching method, priority queue

The seismic images are representations of the interior of the earth that identifies objectives of exploration in the oil and gas industry; these images are usually represented in two coordinate systems: time coordinates and depth coordinates. The time images are adequate only for places where the seismic velocity depends primarily on the depth and the depth images are adequate for areas of structural complexity or arbitrary seismic velocity. This work is based on time to depth conversion proposed by Maria Kourkina Cameron, whose goal is to convert seismic images represented in time coordinates to seismic images in depth coordinates. The main building block of this conversion algorithm is Fast Marching Method proposed by James Sethian. The aim of this research is to implement and evaluate the performance of time to depth conversion built with an efficient fast marching method based on a data structure denominated untidy priority queue. For the evaluation, we considered two evaluation criterions, execution time and image accuracy. Results demonstrated that the time to depth conversion using untidy fast marching method improves the execution time with a small error in the final image.

³ Degree Project

⁴ Faculty of Physics Mechanics Engineering. Electrical, Electronics Engineering and Telecommunications School. Director: PhD César Antonio Duarte Gualdrón. Codirector: MIE. Carlos Andrés Niño Niño.

INTRODUCCIÓN

Las imágenes sísmicas son una representación del interior de la tierra que permiten detectar acumulaciones de hidrocarburos potenciales para la industria petrolera (petróleo y gas) [1]. Los procesos de formación de una imagen sísmica requieren datos sísmicos, estos son registros de las ondas acústicas tomados en la superficie de la tierra con un arreglo de sensores denominados geófonos [2].

Existen dos procesos básicos para producir una imagen sísmica: migración en tiempo y migración en profundidad. La migración en tiempo es un proceso rápido, común y robusto para conseguir una imagen sísmica [3], es adecuada solo para lugares donde las velocidades sísmicas dependen principalmente de la profundidad. Estas velocidades hacen referencia a las velocidades de las ondas en la tierra [4]. Además, este tipo de migración produce imágenes en coordenadas de tiempo y no tienen relación directa con las coordenadas en profundidad (coordenadas espaciales del subsuelo) cuando la velocidad varía lateralmente [5]. Un producto adicional de la migración en tiempo, son velocidades medias conocidas como velocidades de la migración en tiempo o también denominadas, velocidades migradas.

Una aproximación alternativa para obtener una imagen sísmica es llamada migración en profundidad [6]; es adecuada para áreas que también incluyen variación lateral de la velocidad y produce imágenes sísmicas en coordenadas de profundidad. El mayor desafío con esta aproximación es que requiere un modelo de velocidad sísmica [7], construirlo puede llevar mucho tiempo debido a que se utiliza una aproximación iterativa seguida de correcciones. En [3] se discuten dos estrategias para estimar un modelo inicial de velocidad; la primera, es una

conversión tiempo a profundidad basada en la interpretación de los datos sísmicos en el dominio del tiempo; y la segunda, una estrategia de inversión capa a capa basada en el dominio de profundidad.

El petróleo tiende a estar en estructuras complejas tales como intrusiones salinas y fallas [1]; en estas áreas, la migración en tiempo produce imágenes estructuralmente distorsionadas y son inadecuadas para una interpretación geológica precisa. Para remover estos errores, es necesario realizar migración en profundidad [6]; esta permite obtener una imagen de la tierra lo más real posible, con la correcta ubicación y forma del subsuelo [5] [1] [8]. Por esta razón, si es necesario realizar migración en profundidad, también es necesario disponer de un modelo de velocidad sísmica.

El presente trabajo se basa en el algoritmo de conversión tiempo a profundidad presentado en [5]. A partir de las velocidades migradas, este algoritmo construye un razonable modelo inicial de velocidad sísmica para la migración en profundidad [7]. Tiene como bloque principal de construcción el algoritmo de marcha rápida, el cual, está diseñado para resolver un problema de frontera para la ecuación eikonal, su eficiencia computacional es de orden $O(\log_2 N)$, donde N es el número de puntos en el dominio [9]. La notación $O()$ define la tasa de crecimiento de diferentes funciones en un algoritmo y sirve para comparar las velocidades de diferentes algoritmos que hacen el mismo trabajo con los mismos datos de entrada [10], en análisis de algoritmos, se considera que la eficiencia computacional de un algoritmo es más importante que la tecnología usada para ejecutarlo [11], para más detalles referirse a [11] [12].

Existen diferentes alternativas que han sido propuestas para mejorar la eficiencia computacional del Método de Marcha Rápida tradicional [13]. El presente trabajo aplica una de estas alternativas (El método de marcha rápida desordenada [14] [15]) en el algoritmo de conversión tiempo-profundidad y evalúa si se mejora la eficiencia en tiempo de ejecución sin pérdida de precisión de la imagen. Este método, a diferencia del método de marcha rápida tradicional, utiliza una estructura de datos denominada pila de prioridad desordenada con la cual se consigue una eficiencia computacional lineal $O(N)$. El posible error introducido por la simplificación se puede mantener del mismo orden de magnitud que el error introducido por la discretización espacial inherente a las implementaciones numéricas, en la práctica estos errores son virtualmente insignificantes [14].

La evaluación consiste de dos partes. La primera, es una comparación de tiempos de ejecución de los dos algoritmos conversión tiempo a profundidad; las implementaciones se realizan con el software MATLAB. En cada experimento, se mantienen las mismas condiciones: estructura de computo, procesador, RAM, memoria caché y procesos de ejecución para evitar errores en la comparación. Además, se toman varias medidas de tiempo y se realiza un promedio con el fin de reducir la influencia de los eventos aleatorios que existen en el sistema operativo. La segunda parte es una evaluación de precisión de la imagen del algoritmo de conversión modificado, para esto se calcula el error relativo de la imagen con respecto a la imagen del algoritmo de conversión normal.

El presente trabajo se organiza en cuatro capítulos principales. El primero describe los conceptos básicos: método de marcha rápida, método de marcha rápida desordenada y algoritmo de conversión tiempo a profundidad propuesto en [5]. El segundo describe solo las implementaciones numéricas de las dos estructuras de datos básicas de los algoritmos debido a que estas son la principal diferencia entre

las implementaciones. El tercer capítulo presenta, analiza y evalúa los resultados de los dos algoritmos conversión tiempo a profundidad: con marcha rápida y con marcha rápida desordenada, la sección final presenta las recomendaciones del trabajo futuro. El cuarto y último capítulo presenta las conclusiones del proyecto.

1. MARCO TEÓRICO

1.1 MÉTODO DE MARCHA RÁPIDA

Es un método desarrollado por Sethian [9] que permite resolver el problema de valor inicial: la ecuación Eikonal (1) junto con una condición de frontera inicial. Hoy en día, existen diferentes versiones de este método [13] y han sido utilizados en diferentes aplicaciones: visión por computador [16] [17], sismología [5] [18], robótica [19] [20] y tomografía [21]. La ecuación (1) describe la propagación del primer arribo de una onda en un medio, el término T es el tiempo de viaje de la onda y S es el inverso de la velocidad (F) en el medio [9].

$$|\nabla T(z, x)| = S(z, x) \quad (1)$$

Los métodos de marcha rápida computan la solución de (1) utilizando esquemas de diferencias finitas de primer y segundo orden. Una discretización de primer orden en dos dimensiones para resolver esta ecuación se presenta en [13].

$$\left[\begin{array}{l} \max(D_{ij}^{-x} T, -D_{ij}^{+x} T, 0)^2 \\ + \max(D_{ij}^{-y} T, -D_{ij}^{+y} T, 0)^2 \end{array} \right]^2 = S_{ij} \quad (2)$$

Donde D_{ij}^{+} y D_{ij}^{-} son los operadores adelante y hacia atrás de la primera derivada. Los índices del espacio discreto a lo largo del eje x y eje y son i y j , respectivamente.

La ecuación (2) se puede resolver iterativamente, sin embargo, la observación de que esta iteración contiene una relación de causalidad conduce a la optimización del método de marcha rápida. Aprovechando el hecho de que la ecuación (2)

permite propagar información en una sola dirección, de pequeños a grandes valores de T , la ecuación se puede resolver sin iteración [9].

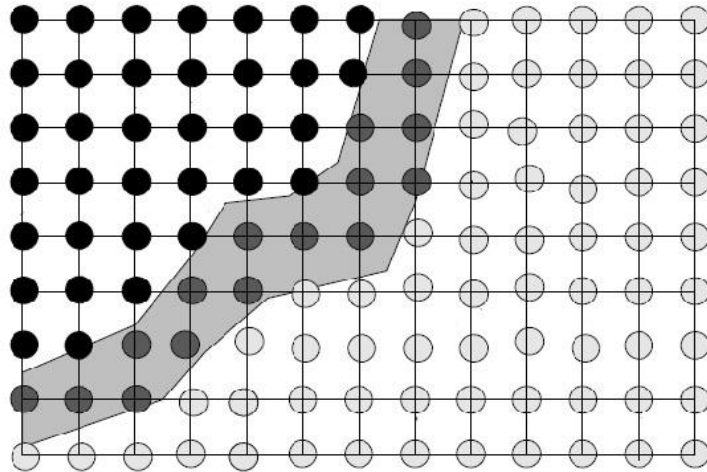
Para entender el método, considere tres conjuntos disjuntos de puntos: *conocidos*, *frente de onda* y *desconocidos*; el conjunto del *frente de onda*, también es conocido como *banda estrecha*. Estos tres conjuntos contienen todos los puntos discretos del espacio. Además, asuma como condición de frontera o fuente, una región del espacio y asigne a esta un valor de tiempo igual a cero; también considere que en todo el dominio la velocidad es mayor que cero. El procedimiento, tomado de [9] es:

Se asignan los puntos en la frontera al conjunto de conocidos, sus puntos vecinos al conjunto del frente de onda y el resto de puntos al conjunto de desconocidos. El ciclo es el siguiente:

- Sea A el punto del *frente de onda* con el valor más pequeño de T .
- Agregue el punto A al conjunto de *conocidos*.
- Agregue al conjunto del *frente de onda* todos los vecinos de A que no son *conocidos*. Si los vecinos son *desconocidos* remuévalos después de agregarlo al *frente de onda*.
- Recalcular los valores de T de los vecinos de A de acuerdo a la ecuación cuadrática (2).
- Regresar al inicio del ciclo si hay puntos en el *frente de onda*.

La Figura 1 ilustra una imagen del progreso del método en un dominio dado.

Figura 1. Progreso del método de marcha rápida. Los puntos negros son los *conocidos*, los grises el *frente de onda* y los blancos los *desconocidos*. Fuente: Tomado de [9].



La clave de una eficiente versión del método yace en encontrar una forma rápida de localizar el punto en el frente de onda con el valor más pequeño de T [9]. Para implementar el método de marcha rápida, en [13] se utiliza una estructura de datos denominada pila binaria mínima para representar el conjunto del *frente de onda*. Esta estructura es un completo árbol binario con la propiedad de que cada valor en cualquier nodo es menor o igual que el valor de sus nodos hijos. Para implementar la estructura de datos se necesitan cuatro operaciones [13]:

- Insertar un nuevo punto a la pila.
- Actualizar un punto ya existente cuyo valor ha cambiado.
- Encontrar mínimo.
- Remove mínimo.

Un ejemplo de esta estructura de datos se muestra en la Figura 2, y un resumen de las complejidades computacionales de sus operaciones se muestra en la Tabla 1. En el peor de los casos, si se consideran que todos los N puntos de la grilla se encuentran en la banda estrecha, se necesitarían N operaciones de inserción y N

de remover el mínimo, lo que conduce a una complejidad computacional de $O(N \log_2 N)$ en el peor de los casos [13].

Figura 2. Estructura pila mínima. Fuente: Tomado de [9].

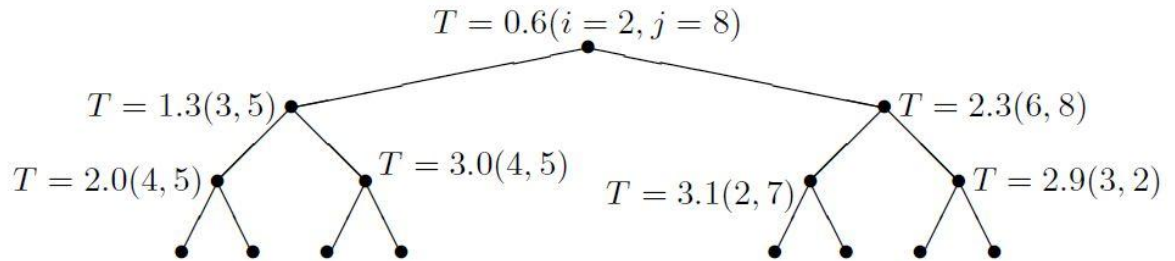


Tabla 1. Complejidad computacional de las operaciones en la pila binaria. n es el número de puntos en la estructura.

Operación	Complejidad computacional
Insertar	$O(\log_2 n)$
Actualizar	$O(n)$
Encontrar mínimo	$O(1)$
Remover mínimo	$O(\log_2 n)$

1.2 MÉTODO DE MARCHA RÁPIDA DESORDENADA

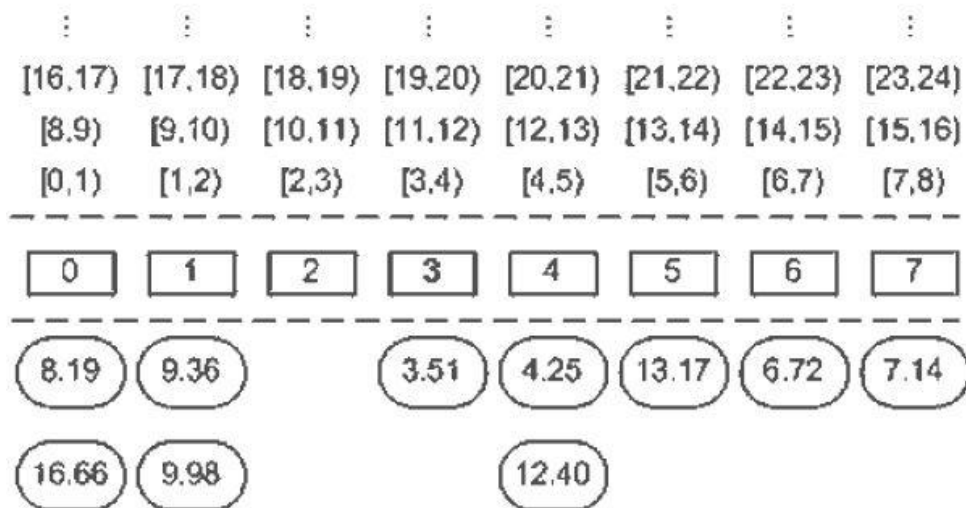
El método de marcha rápida desordenada sigue el mismo procedimiento que el método de marcha rápida; la diferencia con este último es que utiliza una estructura de datos especial que reduce la complejidad computacional del método a $O(N)$ [14]. Esta estructura es denominada pila de prioridad desordenada.

La pila desordenada es un conjunto de M diferentes listas que contienen los diferentes tiempos de propagación en el frente de onda. Esta pila se implementa

como un arreglo circular de M listas FIFO y se caracteriza por que cada uno de ellas contiene valores muy cercanos de tiempo pero de manera desordenada, esto permite conseguir complejidades computacionales constantes $O(1)$ para las operaciones de insertar y remover mencionadas en la sección anterior y reducir la complejidad computacional total del algoritmo [13]. Sin embargo, como los elementos en cada lista están desordenados se introducen errores en el resultado final. El error introducido está en el orden del error introducido por la discretización [14].

Los dos parámetros importantes de esta estructura de datos son: el número de listas M y el rango de tiempos que contiene cada lista (δ). A medida que aumenta el número de listas se reduce el error relativo del método de marcha rápida desordenada comparado con el método de marcha rápida normal. En la Figura 3 se muestra un ejemplo de esta estructura.

Figura 3. Estructura de datos para el método de marcha rápida desordenada. $M = 8$ y $\delta = 1$. Fuente: Tomada de [22].



En [15] se demuestra que un valor apropiado para δ es:

$$\delta = \frac{h}{MF_{min}} \quad (3)$$

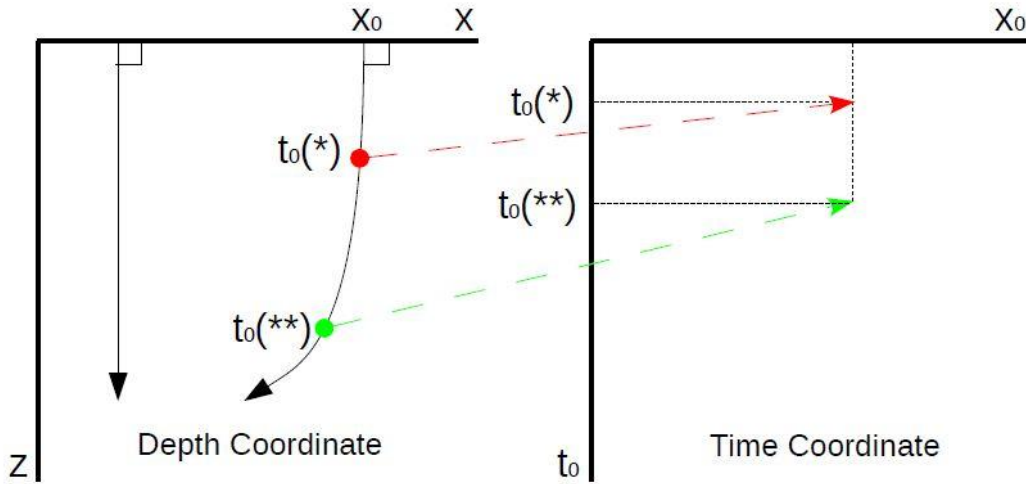
Donde h es el espaciamiento de la grilla y F_{min} la velocidad mínima del medio. Escogiendo este valor de δ , se tiene un único parámetro que define la pila de prioridad desordenada: M

1.3 CONVERSION TIEMPO A PROFUNDIDAD

Un punto en el interior de la tierra puede ser representado en dos coordenadas: coordenadas de profundidad y coordenadas de tiempo. Considere un punto α ubicado en el subsuelo. Las coordenadas de profundidad dan la posición espacial o real del punto $\alpha(z, x)$, donde x es la posición lateral y z la profundidad del punto. Las coordenadas de tiempo corresponden al par ordenado $\alpha(t_0, x_0)$, donde la coordenada x_0 es el primer arribo en la superficie de una onda que parte desde el punto α y t_0 es el tiempo que demora en hacerlo.

La conexión entre estos dos tipos de coordenadas está dado por el concepto de rayo imagen introducido por Hubral en el año 1977 [23]. Para entender este concepto, considere la aproximación de alta frecuencia de las ondas en la tierra que resulta en la ecuación eikonal (1). Las características de esta ecuación pueden ser vistas como rayos. De todos los rayos que parten del punto $\alpha(z, x)$, el rayo que llega con menor tiempo y perpendicular a la superficie es denominado el rayo imagen. La Figura 4 muestra un ejemplo del mapeo de dos puntos del dominio de tiempo a profundidad utilizando este concepto.

Figura 4. Relación entre coordenadas de tiempo y coordenadas de profundidad. Fuente: Tomado de [7].



El algoritmo de conversión tiempo a profundidad propuesto en [5], convierte las velocidades de la migración en tiempo $v(t_0, x_0)$ a velocidades sísmicas $v(z, x)$, esta salida es utilizada como modelo de velocidad para la migración en profundidad.

1.3.1 PROBLEMA DEL VALOR DE FRONTERA. La entrada del algoritmo está en el dominio del tiempo (t_0, x_0) y la salida en el dominio de profundidad (z, x) . Sea $S(z, x)$ el inverso de la velocidad en el punto (z, x) . El inverso de la velocidad lo denominaremos suavidad. Considere un rayo imagen que parte de un punto (z, x) a un punto de la superficie x_0 con un tiempo de viaje t_0 . Sea $x_{min} < x_0 < x_{max}$, $0 < t_0 < t_{max}$, $x_{min} < x < x_{max}$, $z_{min} < z < z_{max}$. A partir de $S(t_0, x_0)$ el algoritmo encuentra $S(z, x)$, $x_0(z, x)$ y $t_0(z, x)$, esto es, la suavidad en el punto (z, x) , la localización en la superficie del rayo imagen que parte desde (z, x) y el tiempo de viaje a lo largo del rayo. Estas últimas tres variables están relacionadas de acuerdo al siguiente sistema de ecuaciones:

$$|\nabla T|^2 = S^2(x_0(z, x), t_0(z, x)) \quad (4)$$

$$\nabla t_0 \cdot \nabla x_0 = 0 \quad (5)$$

La ecuación (4) es la ecuación eikonal y la ecuación (5) indica que los rayos imágenes son ortogonales a las curvas $T = \text{constantes}$. Las condiciones de frontera para el sistema son:

$$x_0(0, x) = x, \quad t_0(0, x) = 0, \quad S(0, x) = S(x_0 = x, t_0 = 0) \quad (6)$$

2. IMPLEMENTACIÓN NÚMERICA

Para la implementación de los algoritmos se utilizaron los siguientes recursos: procesador Intel Core i7 3.60GHz x8, 15.7 GB de RAM, NVIDIA GTX 750 y sistema operativo Debian Jessie. La realización del programa de marcha rápida, marcha rápida desordenada y conversión tiempo a profundidad se realizó con el software MATLAB siguiendo los algoritmos numéricos descritos en [9], [15] y [5], respectivamente. Como se describió en el capítulo anterior, el procedimiento de los dos métodos de marcha rápida es el mismo, a diferencia que cada uno de ellos implementa una estructura de datos diferentes para poder almacenar los puntos del conjunto del *frente de onda*. En este capítulo se presenta los esquemas del árbol binario para el método de marcha rápida y la pila de prioridad desordenada para el método de marcha rápida desordenada que se utilizaron para la implementación.

2.1 PILA BINARIA MÍNIMA

Es un completo árbol binario con la propiedad de que cada nodo tiene un valor menor o igual que sus nodos hijos. Este árbol puede ser representado por una simple lista. Si p es la ubicación de un nodo del árbol en la lista, la ubicación de los nodos hijos son simplemente $2p$ y $2p + 1$. La Figura 5 muestra un árbol binario mínimo y su representación en una lista.

2.2 PILA DE PRIORIDAD DESORDENADA

La pila de prioridad desordenada es un conjunto de M diferentes listas. Esta se implementa como un arreglo circular de M listas FIFO y se caracteriza por que cada uno de ellas contiene valores muy cercanos de tiempo pero de manera desordenada. Debido a que el número de puntos en cada lista FIFO no es la

misma, un simple arreglo matricial tendría espacios de memoria innecesarios, por la tanto, para nuestra implementación se utiliza una estructura matricial de datos en MATLAB creada utilizando el comando struct. Cada campo de la estructura es una de las M listas de la pila. En la Figura 6 se ejemplifica la implementación de esta estructura matricial llamada por brevedad Pila y cuyo campo se llama lista.

Figura 5. Pila binaria mínima y su representación en una simple lista. Fuente: Tomado de [24]

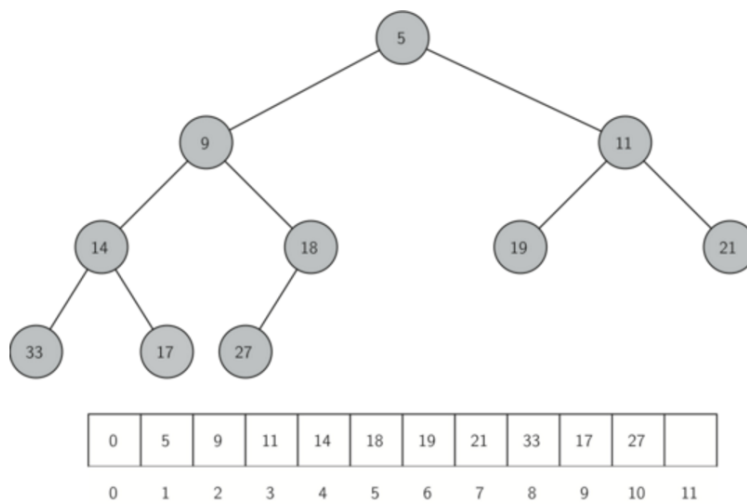
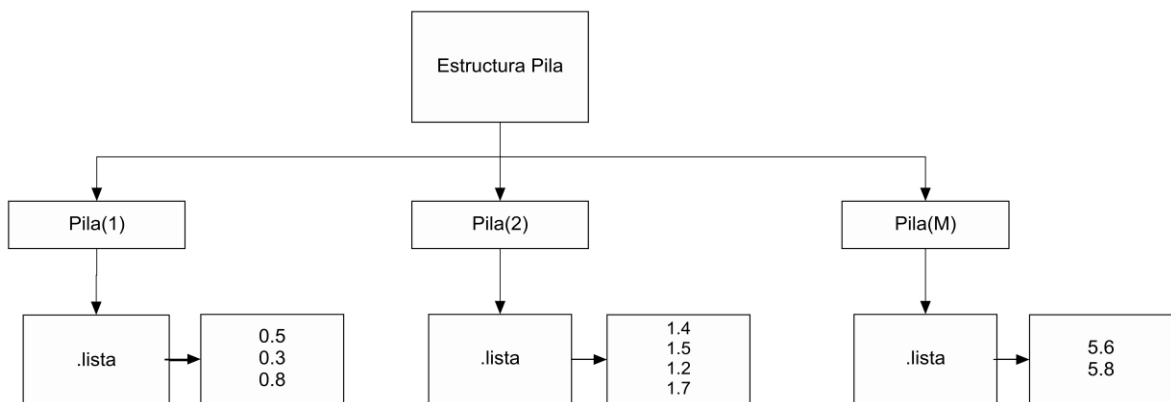


Figura 6. Pila de prioridad desordenada implementada con MATLAB.



3. RESULTADOS EXPERIMENTALES

El presente capítulo muestra los resultados experimentales del trabajo. La primera sección compara la implementación del método de marcha rápida normal y el método de marcha rápida desordenada. La segunda sección compara el método de conversión tiempo a profundidad [5] implementado con los dos diferentes métodos de marcha rápida en tres diferentes experimentos; para cada experimento se evalúa el tiempo de ejecución y la precisión de las imágenes resultantes. La última sección presenta una discusión acerca del trabajo futuro.

Para la implementación de estos métodos se utilizó como software de desarrollo el paquete básico de MATLAB. Para una buena comparación, en cada experimento, se utilizan los mismos recursos: procesador Intel Core i7 3.60GHz x8, 15.7 GB de RAM, NVIDIA GTX 750 y sistema operativo Debian Jessie; además, se mantienen los mismos procesos de ejecución para evitar errores en la comparación. Con el fin de reducir la influencia de los eventos aleatorios que existen en el sistema operativo sobre las medidas de tiempo, se calcula el promedio de 100 de ellas y se toma como medida para realizar la comparación.

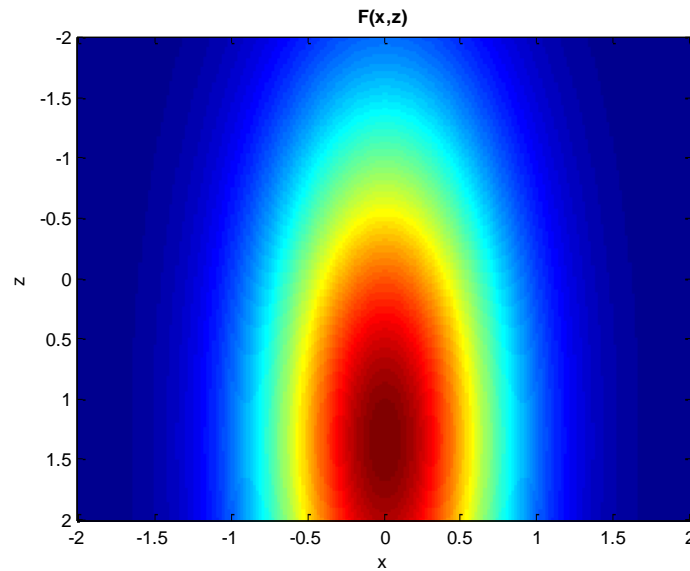
3.1 MÉTODO DE MARCHA RÁPIDA

Se compara el método de marcha rápida y marcha rápida desordenada calculando con cada uno de ellos los tiempos de propagación de una onda que viaja en el subsuelo. El campo de velocidades del medio es una anomalía gaussiana de la forma:

$$F(x, z) = 1 + \exp(-c(x^2 + (z - 1)^2)) \quad (7)$$

El parámetro c indica que tan aguda es la anomalía gaussiana, mayores valores de c indican una anomalía más aguda. El dominio discreto es una grilla 200×200 $x \times z$ donde $x, z \in [-2, 2]$. Tomamos un valor de $c = 1$ con el cual resulta el modelo de velocidades de la Figura 7.

Figura 7. Modelo de velocidad del medio



Se aplica los dos métodos de marcha rápida teniendo como condición de frontera la superficie del medio y el modelo de velocidades de la Figura 7. Debido a que el método de marcha rápida desordenada tiene diferentes estructuras dependiendo del parámetro M , en cada comparación se probará con valores de $M = 2, 10, 50, 100, 500, 1000, 10000$. El resultado del método de marcha rápida se muestra en la Figura 8 y los resultados del método de marcha rápida desordenada para diferentes valores de M se muestran en la Figura 9. El tiempo de ejecución del método de marcha rápida fue de 13.385 segundos. En la Tabla 2 se muestra los tiempos de ejecución y el error relativo máximo del algoritmo de marcha rápida desordenada comparado con el método de marcha rápida.

Figura 8. Tiempos de propagación calculados por el algoritmo de marcha rápida.

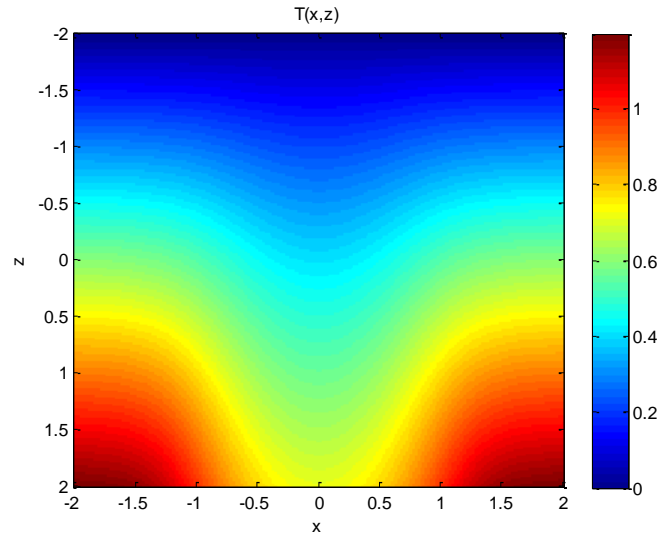


Figura 9. Tiempos de propagación calculados por el algoritmo de marcha rápida desordenada para diferentes valores de M (a) $M = 2$. (b) $M = 10$. (c) $M = 50$. (d) $M = 100$. (e) $M = 500$. (f) $M = 1000$. (g) $M = 10000$.

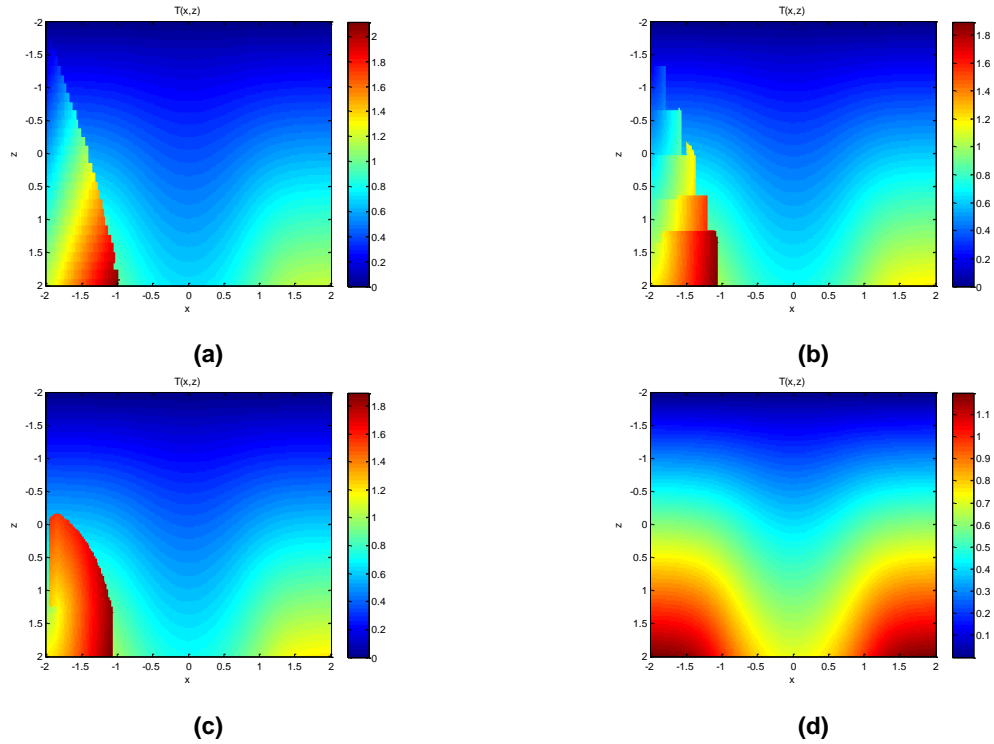


Figura 9. (Continuación)

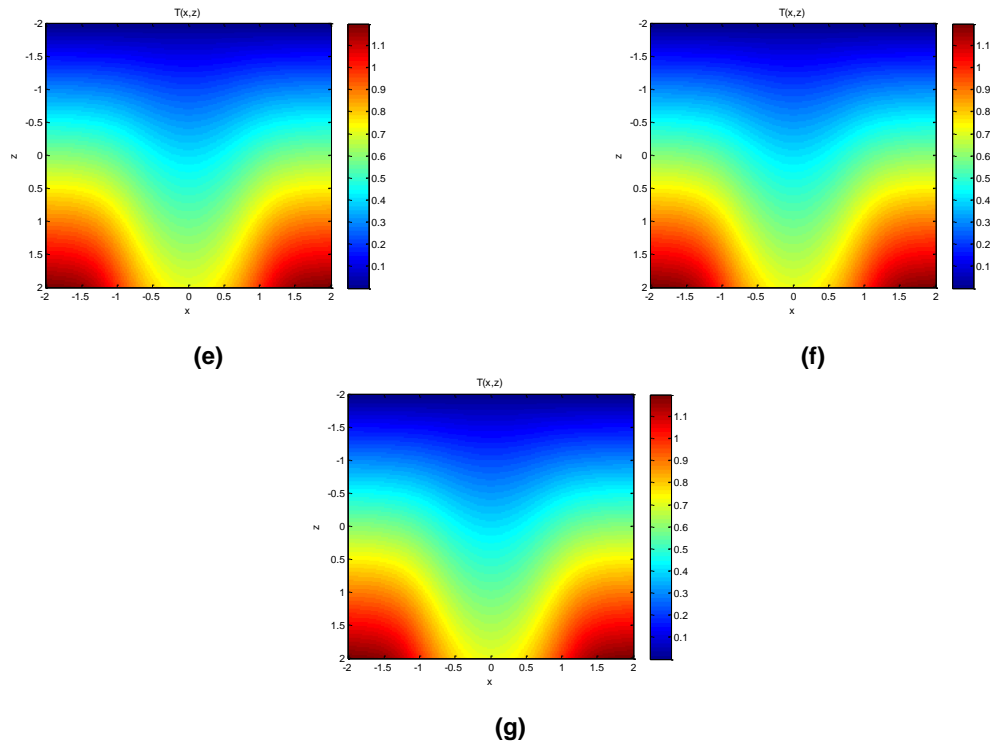


Tabla 2. Tiempo de ejecución y error relativo del método de marcha rápida para diferentes valores del parámetro M

Número de listas M	Tiempo de ejecución [s]	Error relativo máximo [%]
2	6.59	120
10	6.51	117
50	6.49	181
100	6.50	0.23
500	6.44	0.23
1000	6.47	0.23
10000	6.44	0.23

De la Tabla 2, se puede observar que a mayor cantidad de listas, el tiempo de ejecución no cambia significativamente pero el error relativo máximo disminuye y se establece en 0.23 %. En [14], se muestra que efectivamente el error disminuye

al aumentar el valor de M ; además, basado en rigurosos estudios empíricos, en [22] se toma como un valor admisible para el número de listas la cantidad $M = 2N_d$, donde N_d es el número de puntos en la estructura de datos. Para este experimento, el máximo valor de N_d que se podría tener es el número de puntos en la frontera del dominio, por tanto, el valor indicado sería $M = 2 \times 800 = 1600$. Como este valor se encuentra en el rango de M con el menor error relativo, aceptamos este valor empírico para M y lo usaremos en los experimentos restantes.

3.2 ALGORITMO DE CONVERSIÓN TIEMPO A PROFUNDIDAD

Para evaluar la mejora en tiempo del algoritmo de conversión tiempo a profundidad usando el algoritmo de marcha rápida desordenada se realizaron tres experimentos en dos diferentes medios analíticos tomados de [7], en cada experimento se prueban el algoritmo de conversión tiempo a profundidad con marcha rápida y con marcha rápida desordenada y se comparan sus tiempos de ejecución. Los dos primeros experimentos se realizan manteniendo el mismo número de datos $N = 200 \times 700$ y en dos diferentes medios. El tercer experimento se realiza variando el número de puntos del dominio N y con el mismo medio del experimento 1. Además, para mostrar que esta mejora en tiempo no produce errores considerables se han comparado las imágenes resultantes con la imagen esperada teóricamente y se ha calculado el error relativo máximo y promedio para su evaluación.

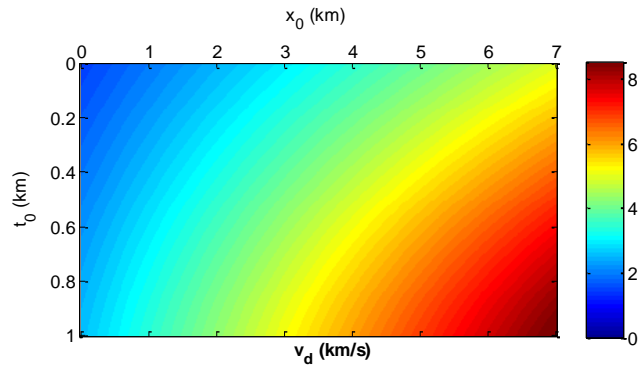
Cabe resaltar que las velocidades analíticas en los experimentos incluyen variación lateral de la velocidad para comprobar que el algoritmo de conversión tiempo a profundidad produce un modelo de velocidad adecuado para la migración en profundidad.

3.2.1 Experimento 1. El primer medio es un modelo de velocidad con gradiente constante cuya velocidad en coordenadas de tiempo y entrada al algoritmo es:

$$v_d(t_0, x_0) = \frac{(v_0 + g_x x_0)^2}{g \cosh(gt_0) - g_z \sinh(gt_0)} \quad (8)$$

Donde v_0 , g_x y g_z son constantes y $g = \sqrt{g_x^2 + g_z^2}$. Para el experimento se tomaron valores de $v_0 = 1.75$, $g_x = 0.5$ y $g_z = 0.75$, la entrada resultante se muestra en la Figura 10. El dominio de entrada discreto es una grilla 200×700 $t_0 \times x_0$ donde $t_0 \in [0, 1]$ y $x_0 \in [0, 7]$.

Figura 10. Modelo de velocidad de entrada para el experimento 1.



Las salidas en profundidad del algoritmo se describen por las siguientes ecuaciones:

$$v(z, x) = v_0 + g_x x + g_z z \quad (8)$$

$$t_0(z, x) = \frac{1}{g} \operatorname{arccosh} \left[\frac{g^2 (\sqrt{(v_0 + g_x x)^2 + g_x^2 z^2} + g_z z) - v g_z^2}{v g_x^2} \right] \quad (9)$$

$$x_0(z, x) = x + \frac{\sqrt{(v_0 + g_x x)^2 + g_x^2 z^2} - (v_0 + g_x x)}{g_x} \quad (10)$$

El dominio de salida discreto es una grilla 200×700 $z \times x$ donde $z \in [0, 2]$ y $x \in [0, 7]$. La Figura 11, Figura 12 y Figura 13 muestran los resultados analíticos, los resultados de la implementación y el error relativo de las velocidades de salida experimentales, respectivamente. En la Tabla 3 se resumen los tiempos de ejecución, error relativo máximo y error relativo promedio de la velocidad para el algoritmo de conversión tiempo a profundidad utilizando las dos versiones de marcha rápida. El dominio de salida indica un valor máximo de $N_d = 1800$, por lo tanto, para el algoritmo de marcha rápida desordenada se toma un valor de $M = 3600$.

Figura 11. Resultados analíticos de la conversión tiempo a profundidad. (a) $v(z, x)$ (km/s). (b) $t_0(z, x)$ (s). (c) $x_0(z, x)$ (km).

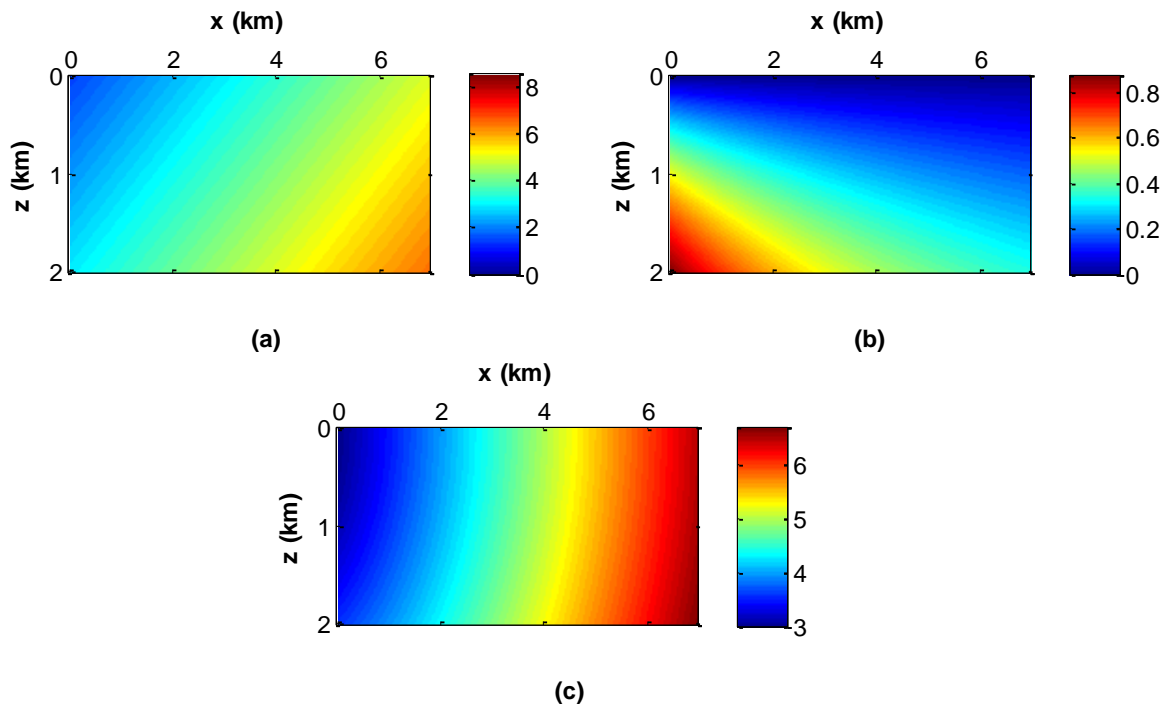


Figura 12. Resultados experimentales. $v(z, x)$ (arriba), $t_0(z, x)$ (medio) y $t_0(z, x)$ (abajo), del algoritmo de conversión tiempo a profundidad usando (a) método de marcha rápida (b) método de marcha rápida desordenada

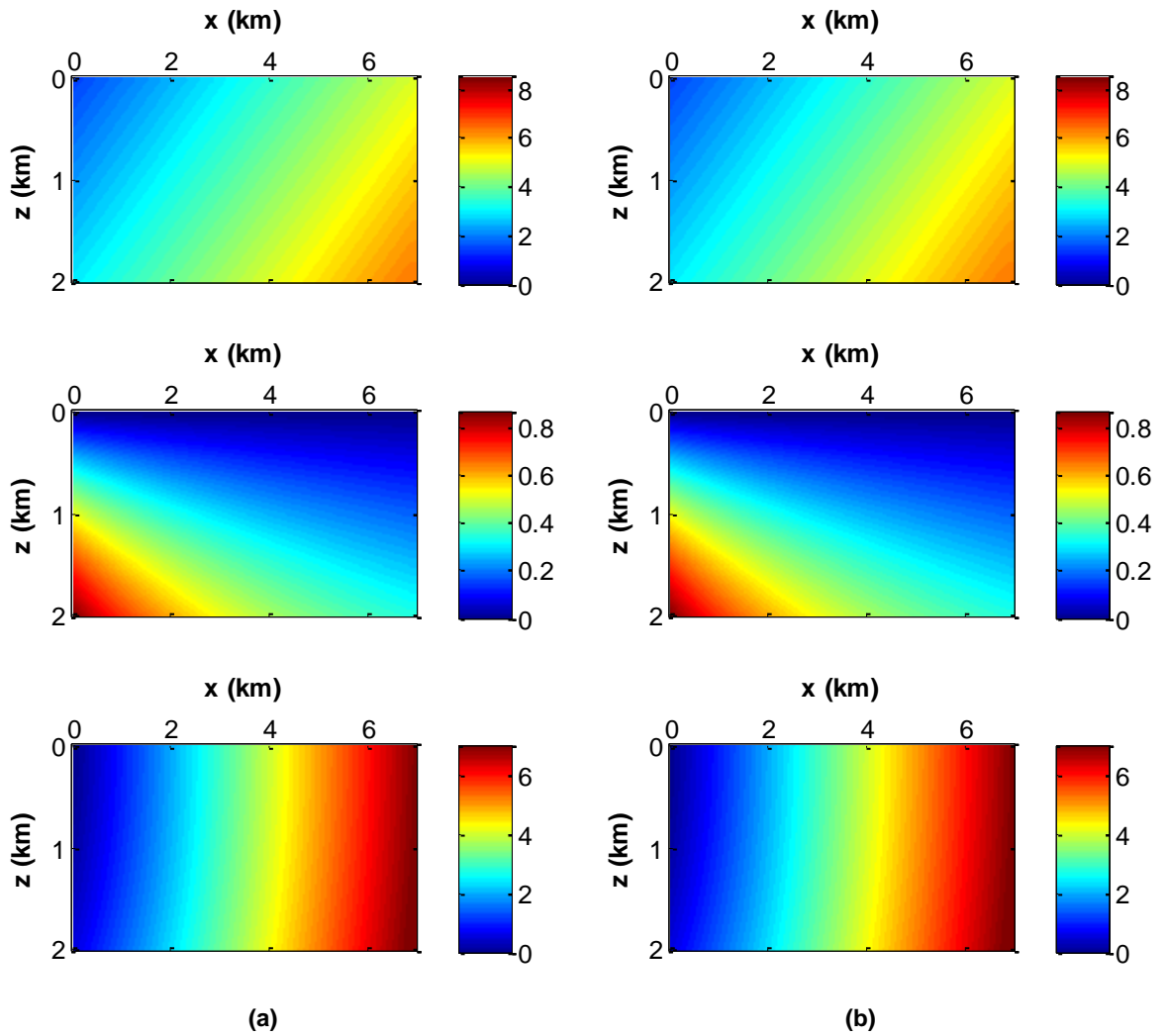


Figura 13. Error relativo velocidades de salida de los algoritmos de conversión utilizando (a) marcha rápida. (b) marcha rápida desordenada.

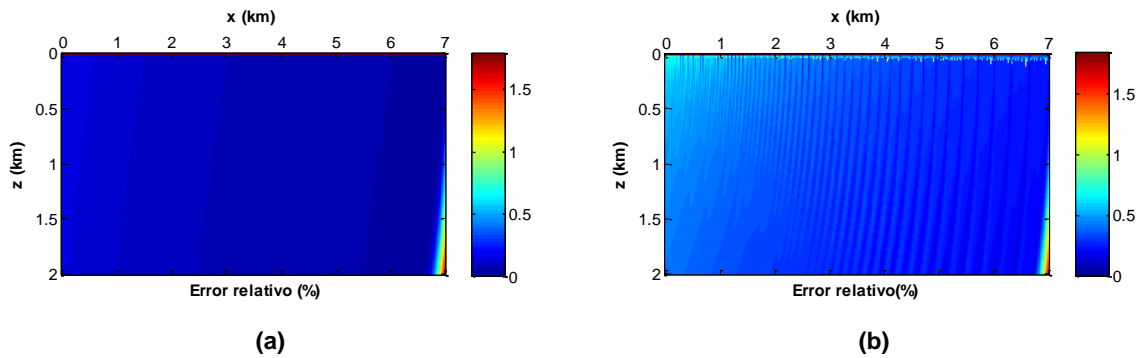


Tabla 3. Tiempo de ejecución, error relativo máximo y error relativo promedio del algoritmo de conversión utilizando los diferentes métodos de marcha rápida.

	Tiempo de ejecución [s]	E_r [%]	\bar{E}_r [%]
Marcha rápida	221.32	1.7974	0.0915
Marcha rápida desordenada	119.28	1.8369	0.3351

De la Tabla 3 se puede deducir que el aumento de velocidad al utilizar el método de marcha rápida desordenada es de 1.85 veces mayor que utilizar el método de marcha rápida con una diferencia de precisión de alrededor del 2%.

3.2.2 Experimento 2. El segundo medio es un modelo de velocidad de entrada descrito por la siguiente ecuación:

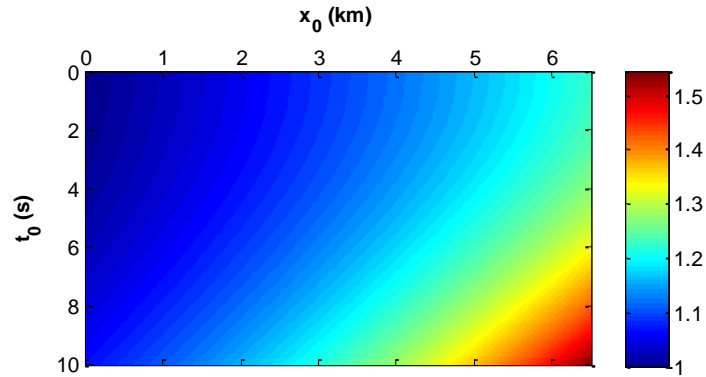
$$v_d(t_0, x_0) = \frac{\sqrt{\omega_0 - 2q_x x_0}}{\omega_0 - 2q_x x_0 - q_x^2 \sigma^2} \quad (11)$$

Donde ω_0 y q_x son constantes y σ está definido por:

$$\sigma(t_0, x_0) = \left[\frac{3t_0 + \sqrt{9t_0^2 + 4(\omega_0 - 2q_x x_0)^3 / q_x^2}}{2q_x^2} \right]^{\frac{1}{3}} - \frac{\omega_0 - 2q_x x_0}{q_x} \left[\frac{2}{3q_x t_0 + \sqrt{9q_x^2 t_0^2 + 4(\omega_0 - 2q_x x_0)^3}} \right]^{\frac{1}{3}} \quad (12)$$

El dominio de entrada discreto es una grilla de igual tamaño que el experimento 1, $200 \times 700 t_0 \times x_0$ donde $t_0 \in [0, 10]$ y $x_0 \in [0, 6.5]$. Escogiendo valores de $\omega_0 = 1$ y $q_x = 0.0260$ se obtiene el modelo de velocidad de la Figura 14.

Figura 14. Modelo de velocidad de entrada para el experimento 2.



Analíticamente, la velocidad de salida en profundidad depende sólo de la posición lateral:

$$v(x, z) = \frac{1}{\sqrt{\omega_0 - 2q_x x}} \quad (13)$$

El dominio de salida discreto es una grilla 200×700 $z \times x$ donde $z \in [0, 2]$ y $x \in [0, 6.5]$. La Figura 15 muestra la velocidad de salida analítica y los dos resultados de la implementación. La Figura 16 muestra el error relativo de las velocidades de salida comparadas con la salida analítica. En la Tabla 4 se resumen los resultados de tiempo de ejecución, error relativo máximo y error relativo promedio de la conversión de velocidad utilizando las dos versiones de marcha rápida. Como el dominio de puntos es el mismo que en el experimento 1, para el algoritmo de marcha rápida desordenada también se toma un valor de $M = 3600$.

La Tabla 4 indica un aumento de aproximadamente el doble de velocidad (1.95) usando el método de marcha rápida desordenada y con diferencia de errores de alrededor del 7%.

Figura 15. Velocidades de salida. (a) analítica. (b) marcha rápida. (c) marcha rápida desordenada.

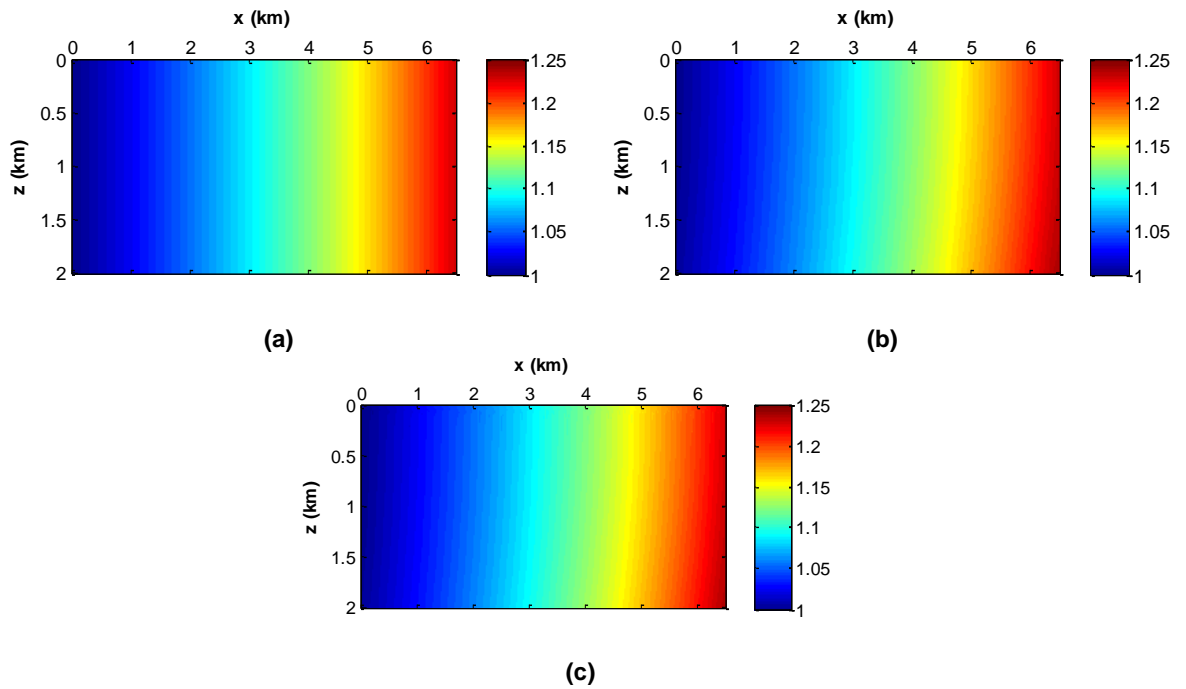


Figura 16. Error relativo velocidades de salida de los algoritmos de conversión utilizando (a) marcha rápida. (b) marcha rápida desordenada.

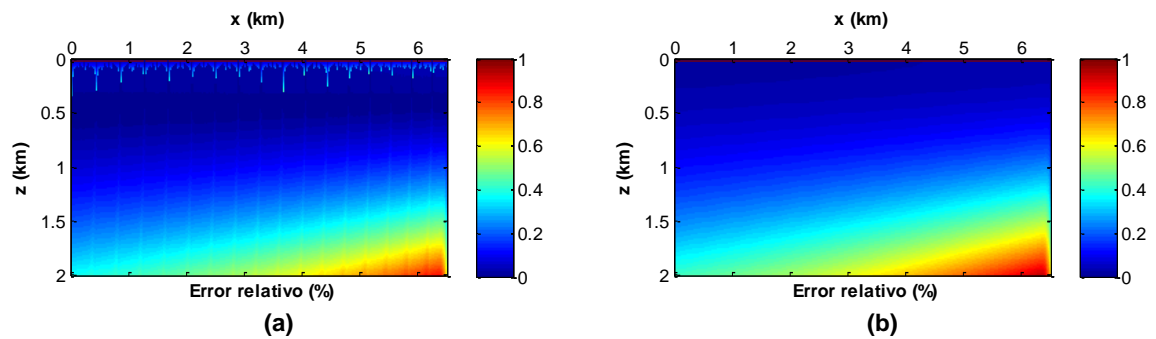


Tabla 4. Tiempo de ejecución, error relativo máximo y error relativo promedio del algoritmo de conversión en el medio 2.

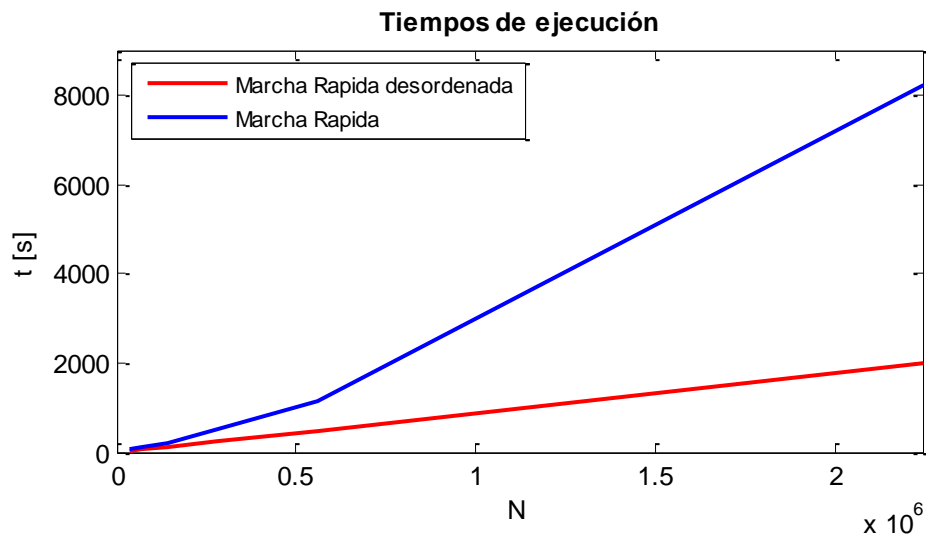
	Tiempo de ejecución [s]	E_r [%]	\bar{E}_r [%]
Marcha rápida	205.6425	0.9416	0.2346
Marcha rápida desordenada	107.1315	0.8698	0.1882

3.2.3 Experimento 3. En este experimento se realiza el mismo experimento 1 pero con diferentes valores de N . La siguiente tabla muestra los resultados de tiempos de ejecución, el error relativo máximo y promedio de las dos implementaciones del algoritmo de conversión. En la Figura 17 se muestra la comparación de los tiempos de ejecución de los dos algoritmos en función de N .

Tabla 5. Tiempo de ejecución, error relativo máximo y error relativo promedio del algoritmo de conversión en el medio 1 con diferentes valores de N

N	Marcha Rápida			Marcha rápida desordenada		
	t [s]	E_r [%]	\bar{E}_r [%]	t [s]	E_r [%]	\bar{E}_r [%]
100×350	53.8103	1.9096	0.1774	30.6924	1.9886	0.6219
100×700	111.6103	1.9096	0.3052	59.8918	1.9786	0.5588
200×700	221.32	1.7974	0.0915	119.28	1.8369	0.3351
200×1400	533.0658	1.7974	0.1555	245.2232	1.8319	0.2909
400×1400	1139.11	1.7419	0.0483	490.19	1.7616	0.1785
800×2800	8214.7	1.7144	0.0268	2007.3	1.7242	0.0934

Figura 17. Tiempos de ejecución del algoritmo de conversión con marcha rápida y marcha rápida desordenada en función del número de puntos del dominio N .



A partir de la Tabla 5, se puede concluir que la velocidad del algoritmo de conversión con marcha rápida desordenada es siempre más alta que el algoritmo de conversión de marcha rápida normal y a una razón mayor a medida que aumenta el número de puntos N . A pesar de que no se sabe cuál es la complejidad computacional resultante de los algoritmos de conversión, de este experimento podemos comprobar que se tiene mejor eficiencia computacional utilizando el método de marcha rápida desordenada.

3.3 DISCUSIÓN

El método de marcha rápida desordenada ha sido utilizado en diferentes medios [25], y en algunas aplicaciones con rendimiento en tiempo real [26] [27]. El mejor rendimiento del método en estos trabajos al igual que en el presente aplicado en sísmica, se ha obtenido realizando la configuración de sus parámetros de manera experimental. Estos trabajos tienen en común que si hay una variación de los datos de entrada hay que reajustar los parámetros del método; de esta observación, se pudo identificar dos trabajos a futuro, el primero es realizar un estudio para elegir de manera automática el valor óptimo de M fijando el parámetro δ según [15], y el segundo es realizar un estudio para configurar automáticamente tanto el parámetro M como el parámetro δ a partir de los datos de entrada para obtener el método más eficiente.

Por otra parte, durante el desarrollo del proyecto se encontraron diferentes implementaciones del método de marcha rápida que mejoran su eficiencia computacional. Entre estas mejoras se encuentra el método de marcha rápida iterativo que podría mejorar considerablemente el tiempo de ejecución del algoritmo a costa de precisión; para aplicaciones de tomografía, en el trabajo [28] se implementa este método iterativo en una GPU y se presenta un aumento de velocidad de hasta 80 veces comparada con el método de marcha rápida normal.

Para aplicaciones sísmicas es importante contar tanto con precisión como rapidez, por tanto se espera implementar el algoritmo de conversión tiempo a profundidad con el método de marcha iterativo en una GPU y realizar una evaluación de la calidad de la imagen resultante.

Finalmente, como sólo se realizaron experimentos con datos sintéticos, se tiene como trabajo posterior realizar experimentos con datos reales y aplicar el algoritmo de conversión de tiempo a profundidad en el proceso de obtener una imagen real de la tierra.

4. CONCLUSIONES

Este trabajo implementa y evalúa el algoritmo de conversión tiempo a profundidad utilizando el método de marcha rápida con una pila binaria y el método de marcha rápida desordenada. Como resultado del trabajo se tiene que en un medio sintético común, el algoritmo de conversión tiempo a profundidad con marcha rápida desordenada se ejecuta en menor tiempo que su contraparte y sin una pérdida considerable de precisión, el error relativo máximo en los experimentos fue menor del 2% y el error relativo promedio menor del 0.7 %. Además, a medida que se aumenta el número de puntos del dominio de salida N , la velocidad del algoritmo de conversión con marcha rápida desordenada comparado con el algoritmo de conversión con marcha rápida clásico es cada vez mayor, para un valor de N alrededor de 2000000 el aumento de velocidad fue de 4 veces mayor.

REFERENCIAS

- [1] ALBERTIN, Uwe, et al. La era de las imágenes en escala de profundidad. *Oilfield Review*. 2002. vol. 14, no. 1, p. 2-17.
- [2] BAETEN, Guido, et al. Acquisition and processing of point receiver measurements in land seismic. *SEG Technical Program Expanded Abstracts 2000*. January 2000, p. 41-44.
- [3] YILMAZ, Özdogan. Seismic data analysis: processing, inversion, and interpretation of seismic data. Tulsa: Society of exploration geophysicists, vol. 1, 2001. 998 p. ISBN 978-1-56080-098-9.
- [4] SHEARER, Peter. Introduction to seismology. Edinburgh: Cambridge University Press, 2009. 391 p. ISBN 978-0-521-88210-1.
- [5] CAMERON, Maria; FOMEL, Sergey y SETHIAN, James. Seismic velocity estimation from time migration. *Inversion problems*. 2007. vol. 23, no. 4, p. 1329.
- [6] CAMERON, Maria; FOMEL, Sergey y SETHIAN, James. Time-to-depth conversion and seismic velocity estimation using time-migration velocity. *Geophysics*. 2008. vol. 73, no 5, p. 205.
- [7] LI, Siwei and FOMEL, Sergey. A robust approach to time-to-depth conversion and interval velocity estimation from time migration in the presence of lateral velocity variations. *Geophysical Prospecting*. 2015 vol. 63, nº 2, pp. 315-337.
- [8] HERNÁNDEZ, Gloria. Métodos para obtener modelos de velocidad para la conversión tiempo-profundidad. Tesis de ingeniería geofísica. Ciudad de México: Instituto Politécnico Nacional. Escuela superior de ingeniería y

arquitectura. Ciencias de la tierra, 2015. 95 p.

- [9] SETHIAN, James. Fast marching methods. *SIAM Review*. 1999. vol. 41, no 2, pp. 199-235.
- [10] PETRUSEVA, Silvana. Comparison of the efficiency of two algorithms which solve the shortest path problem with an emotional agent. *Yugoslav Journal of Operations Research*. 2006. vol. 16, no 2.
- [11] ARORA, Sanjeev and BARAK, Boaz. Computational complexity: a modern approach. Edinburgh: Cambridge University Press, 2009. 583 p.
- [12] MCCONNELL, Jeffrey. Analysis of algorithms. Massachusetts: Jones & Bartlett Publishers, 2009. 435 p. ISBN 978-0-7637-0782-8.
- [13] GÓMEZ, Javier; LUMBIER, Alejandro; GARRIDO, Santiago y MORENO, Luis. Fast methods for eikonal equations: an experimental survey. *Robotics and Autonomous Systems* 2015. vol 61. no 2. pp 137-152.
- [14] YATZIV, Liron; BARTESAGHI, Alberto y SAPIRO, Guillermo. $O(N)$ implementation of the fast marching algorithm. *Journal of computational physics*. 2006, vol. 212, no 2, pp. 393-399.
- [15] RASCH, Christian and SATZGER, Thomas. Remarks on the implementation of the fast marching method. *IMA Journal of Numerical Analysis*. 2009, vol. 29, nº 3, pp. 806-813.
- [16] FORCADEL, Nicolas; LE GUYADER, Carole y GOUT, Christian. Generalized fast marching method: applications to image segmentation. *Numerical Algorithms*. 2008. vol. 48, nº 1-3, pp. 189-211.
- [17] BASU, Sreetama y RACOCEANU, Daniel. Reconstructing neuronal morphology from microscopy stacks using fast marching. *IEEE international*

conference on image processing (ICIP), 2014 pp. 3597-3601.

- [18] ZHANG, Xiaoning y BORDING, R. Fast marching method seismic traveltimes with reconfigurable field programmable gate arrays. *Canadian Journal of Exploration Geophysics*. 2011 vol. 36, nº 1, pp. 60-68.
- [19] GÓMEZ, Javier. Advanced applications of the fast marching square planning method. Master's thesis. Madrid: Carlos III University. Department Of Systems Engineering And Automation. 2013. 137 p.
- [20] LIU, Yuanchang y BUCKNALL, Richard. Path planning algorithm for unmanned surface vehicle formations in a practical maritime environment. *Ocean Engineering*. 2015. vol. 97, pp. 126-144.
- [21] QU, Xin-Xin; LIU, Si-Xin y WANG, Fei. A new ray tracing technique for crosshole radar traveltime tomography based on multistencils fast marching method and the steepest descend method. *International conference on Ground penetrating Radar (15th : 2014: Brussels)*. p. 503-508.
- [22] WUTTISITTIKULKIJ, Chulalongkorn; CHAODIT, Aswakul y SIANGSUKONE, T. Study of optimised bucket widths in calendar queue for discrete event simulator. *Proceedings of Thailand's Electrical Engineering Conference (26th : 2003: Bangkok)*, pp. 6-7.
- [23] HUBRAL, Peter. Time Migration - Some Ray Theoretical Aspects. *Geophysical Prospecting*. 1977. vol. 25, nº 4, pp. 738-745.
- [24] MILLER, Bradley y RANUM, David L. Problem Solving with Algorithms and Data Structures Using Python. Portland: Franklin, Beedle & Associates Inc, 2011.
- [25] GÓMEZ, Javier .Fast marching methods in path and motion planing:

improvements and high-level applications. TESIS DOCTORAL. Madrid: UNIVERSIDAD CARLOS III DE MADRID. Departamento de Ingeniería de Sistemas Y Automática. 2015. 159 p.

- [26] WHITTY, Mark; COSSELL, Stephen; DANG, Kim Son; GUIVANT, Jose y KATUPITIYA, Jayantha. Autonomous navigation using a real-time 3d point cloud. *Australasian Conference on Robotics and Automation* (2010: Brisbane). p. 404-413.
- [27] ROBLED0, Alicia; GUIVANT, Jose y COSSELL, Stephen. Pseudo priority queues for Real-Time performance on dynamic programming processes applied to path planning. *Australasian Conference on Robotics and Automation*, (2010: Brisbane). p. 414-422.
- [28] LI, Shengying; MUELLER, Klaus; JACKOWSKI, Marcel; DIONE, Donald y STAIB, Lawrence. Physical-space refraction-corrected transmission ultrasound computed tomography made computationally practical. *International Conference on Medical Image Computing and Computer-Assisted Intervention* (11th: 2008: New York). p. 280-288.

BIBLIOGRAFÍA

CAMERON, Maria; FOMEL, Sergey y SETHIAN, James. Time-to-depth conversion and seismic velocity estimation using time-migration velocity. *Geophysics*. 2008. vol. 73, no 5, p. 205.

RASCH, Christian and SATZGER, Thomas. Remarks on the implementation of the fast marching method. *IMA Journal of Numerical Analysis*. 2009, vol. 29, n° 3, pp. 806-813.

SETHIAN, James. Fast marching methods. *SIAM Review*. 1999. vol. 41, no 2, pp. 199-235.

YATZIV, Liron; BARTESAGHI, Alberto y SAPIRO, Guillermo. $O(N)$ implementation of the fast marching algorithm. *Journal of computational physics*. 2006, vol. 212, no 2, pp. 393-399.