

Paralelización de un trazador de rayos usando el algoritmo SANEG

Parallelization of a ray tracer using the SANEG algorithm

Joali Moreno (jmoreno@kuaimare.ciens.ucv.ve)

Brígida Molina (bmolina@kuaimare.ciens.ucv.ve)

Universidad Central de Venezuela, Facultad de Ciencias
Dpto. de Computación, Centro de Cálculo Científico y Tecnológico
Caracas, Venezuela.

Resumen

La determinación de la estructura del subsuelo en regiones específicas es un proceso fundamental para la exploración petrolera. La etapa más importante y de mayor costo computacional de este proceso es el trazado de rayos. En el presente trabajo se desarrolla una versión paralela para mejorar el rendimiento computacional de un algoritmo de trazado de rayos, que usa un resolvidor no lineal de tipo espectral (SANEG). Los resultados numéricos muestran una disminución del tiempo de cómputo de un 85 %.

Palabras y frases clave: Trazado de rayos, métodos espectrales, sistemas de ecuaciones no lineales.

Abstract

The determination of subsoil structure in specific regions is a fundamental process for oil exploration. The most important and costly stage of this process is the ray tracing. We developed an alternative parallel version for a ray tracing algorithm to improve computational performance, using an spectral approach for non-linear equation (SANEG). Numerical results indicated that computational time was reduced by 85%.

Key words and phrases: Ray tracing, spectral methods, non-linear equations systems.

Recibido 2006/03/10. Revisado 2006/09/06. Aceptado 2006/09/11.

MSC (2000): Primary 86A15; Secondary 65H10, 65K05, 65K10.

Este artículo fué parcialmente financiado por el Proyecto Agenda Petróleo UCV - 97003769 y por el CDCH-UCV proyecto 03-00-5594-2004.

1 Introducción

El estudio de ondas sísmicas, permite conocer con detalle la profundidad, geometría y otras características de las interfaces que separan medios con distinta velocidad de propagación de las ondas sísmicas. Esto sirve como base para la exploración petrolera, ya que proporciona información importante para determinar regiones del subsuelo donde pueden haberse acumulado hidrocarburos a lo largo del tiempo. Por medio de la Tomografía de Reflexión Sísmica (TRS) se puede obtener una imagen sintética de la región del subsuelo que se desea explorar. Una de las etapas más importantes cuando se resuelve el problema de TRS es el problema directo y este trabajo se enfoca en resolverlo de una manera eficiente y rápida. El problema directo consiste en determinar la trayectoria de la onda al propagarse en el medio y calcular los tiempos de viaje para diferentes localizaciones de pares emisor - receptor. Este problema generalmente es resuelto mediante un algoritmo trazador de rayos sísmicos. Diferentes algoritmos de trazado de rayos, entre otros, pueden encontrarse en [3], [9], [10], [12]. Un modelo matemático que permite resolver el problema planteado, consiste en considerar el trazado de rayos sísmicos como un sistema de ecuaciones no lineales, que se construye discretizando el rayo y forzando a que se satisfaga el Principio de Fermat entre dos puntos consecutivos. El sistema no lineal se resuelve utilizando un método espectral denominado SANE Globalizado (Spectral Approach for Nonlinear Equations) propuesto por La Cruz y Raydán en [8], aunque es claro que puede usarse cualquier otro método para sistemas no lineales.

En este trabajo se desea mejorar el desempeño computacional del trazador, esto se pretende lograr haciendo uso de uno de los avances de la computación como lo es el cómputo paralelo. Para ello se plantea una técnica que permite repartir el trabajo de forma equitativa entre los procesos existentes para realizar el cálculo de la trayectoria de los rayos de forma simultánea. El motivo por el que se desea mejorar el desempeño del trazador de rayos es la gran cantidad de cómputo requerido para el cálculo de los tiempos de viaje, considerando que este aumenta en la medida que el número de rayos en el problema se incrementa.

Existen pocos algoritmos paralelos de trazado de rayos para el problema TRS, entre otros se pueden mencionar el propuesto por Jurado y Monestié en [5] y el desarrollado por Castillo, Hassam, Merentes y Molina en [1]. En ambos trabajos se resuelve el problema del trazador de rayos mediante técnicas de optimización, pero en el primero utilizan un método tipo quasi-Newton para encontrar la trayectoria del rayo [4] mientras que en el segundo usan el método del gradiente espectral [2].

2 Preliminares

En esta sección se presentan las bases teóricas para una mejor comprensión del problema planteado.

2.1 El Trazador de Rayos

El objetivo principal del trazado de rayos de un modelo emisor-receptor es obtener todas las posibles llegadas que contribuyen con una energía significativa dentro de un intervalo de tiempo. Se modela como un sistema de ecuaciones no lineales [7] que busca describir el rayo de Fermat entre el emisor y el receptor. Cabe resaltar que con este procedimiento se realiza una discretización finita del rayo.

2.1.1 El principio de Fermat

El *principio de Fermat* afirma que el rayo correcto entre dos puntos es aquel de menor tiempo de viaje. El tiempo de viaje de una onda sísmica puede calcularse como:

$$\tau^C(s) = \int_C s(x, y, z) dl ; \quad (1)$$

se denomina funcional de tiempo de viaje a lo largo de C , donde dl denota la distancia infinitesimal a lo largo de C .

De acuerdo con el *principio de Fermat* el rayo que conecta dos puntos es aquel que posee menor tiempo de viaje. Entonces el principio de Fermat afirma que

$$\tau^*(s) = \min_{C \in \mathcal{C}} \tau^C(s), \quad (2)$$

donde \mathcal{C} denota el conjunto de todos los rayos continuos que conectan al emisor con el receptor. El rayo que produce el mínimo en (2) se denota C^* .

2.1.2 Trazado de rayos entre dos puntos

Se inicia la descripción con el modelo para un rayo entre dos puntos, donde dicho rayo no cruza ninguna interfaz. Sean $\mathbf{x}^e \in \mathbb{R}^3$ un emisor y $\mathbf{x}^r \in \mathbb{R}^3$ un receptor. Sea C_e^r el rayo de Fermat que conecta a \mathbf{x}^e con \mathbf{x}^r . Se denota con $\mathcal{C}(\mathbf{x}^e, \mathbf{x}^r)$ al conjunto de rayos que conectan a \mathbf{x}^e con \mathbf{x}^r . De esta forma,

$$\tau_{er}(s) = \int_{C_e^r} s(x, y, z) dl = \min_{C \in \mathcal{C}(\mathbf{x}^e, \mathbf{x}^r)} \int_C s(x, y, z) dl, \quad (3)$$

donde $dl = [dx^2 + dy^2 + dz^2]^{1/2}$.

Suponga que se desea aproximar N puntos del rayo C_e^r , donde

$$\begin{aligned} \mathbf{x}^0 &= \mathbf{x}^e = (x_0, y_0, z_0)^t \\ \mathbf{x}^i &= (x_i, y_i, z_i)^t, \quad \text{para } i = 1, \dots, N \\ \mathbf{x}^{N+1} &= \mathbf{x}^r = (x_{N+1}, y_{N+1}, z_{N+1})^t. \end{aligned}$$

Cabe destacar que se trabaja con un medio heterogéneo e isotrópico, por lo tanto la longitud del rayo entre dos puntos consecutivos del mismo, $(x_{i-1}, y_{i-1}, z_{i-1})$ y (x_i, y_i, z_i) , se aproxima por la longitud del segmento de recta que une a tales puntos,

$$l_i = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2}. \quad (4)$$

De esta forma, el tiempo de viaje de la ecuación integral (3) se aproxima por medio de siguiente sumatoria,

$$\mathbf{t}(X) = \sum_{i=1}^{N+1} s_i l_i, \quad (5)$$

donde $X = (x_1, \dots, x_N, y_1, \dots, y_N, z_1, \dots, z_N)^t \in \mathbb{R}^{3N}$ y $s_i = s \left(\frac{x_i + x_{i-1}}{2}, \frac{y_i + y_{i-1}}{2}, \frac{z_i + z_{i-1}}{2} \right)$.

La verdadera trayectoria del rayo tiene tiempo estacionario de \mathbf{x}^{i-1} a \mathbf{x}^i , es por ello que puede determinarse igualando a cero las derivadas parciales de \mathbf{t} de la ecuación (5) con respecto a x_i , y_i y z_i . Esto determina un sistema de $3N$ ecuaciones no lineales con $3N$ incógnitas:

$$\begin{cases} \frac{\partial \mathbf{t}}{\partial x_i} = \frac{\partial s_i}{\partial x_i} l_i + \frac{\partial l_i}{\partial x_i} s_i + \frac{\partial s_{i+1}}{\partial x_i} l_{i+1} + \frac{\partial l_{i+1}}{\partial x_i} s_{i+1} = 0 \\ \frac{\partial \mathbf{t}}{\partial y_i} = \frac{\partial s_i}{\partial y_i} l_i + \frac{\partial l_i}{\partial y_i} s_i + \frac{\partial s_{i+1}}{\partial y_i} l_{i+1} + \frac{\partial l_{i+1}}{\partial y_i} s_{i+1} = 0 \\ \frac{\partial \mathbf{t}}{\partial z_i} = \frac{\partial s_i}{\partial z_i} l_i + \frac{\partial l_i}{\partial z_i} s_i + \frac{\partial s_{i+1}}{\partial z_i} l_{i+1} + \frac{\partial l_{i+1}}{\partial z_i} s_{i+1} = 0, \end{cases} \quad (6)$$

para $i = 1, 2, \dots, N$. Además tenemos que

$$\begin{aligned} \frac{\partial l_i}{\partial x_i} &= \frac{x_i - x_{i-1}}{l_i}, & \frac{\partial l_{i+1}}{\partial x_i} &= -\frac{(x_{i+1} - x_i)}{l_{i+1}} \\ \frac{\partial l_i}{\partial y_i} &= \frac{y_i - y_{i-1}}{l_i}, & \frac{\partial l_{i+1}}{\partial y_i} &= -\frac{(y_{i+1} - y_i)}{l_{i+1}} \\ \frac{\partial l_i}{\partial z_i} &= \frac{z_i - z_{i-1}}{l_i}, & \frac{\partial l_{i+1}}{\partial z_i} &= -\frac{(z_{i+1} - z_i)}{l_{i+1}}. \end{aligned}$$

Sustituyendo las derivadas calculadas, el sistema (6) se reescribe como:

$$\begin{cases} \frac{\partial s_i}{\partial x_i} l_i + \frac{(x_i - x_{i-1})}{l_i} s_i + \frac{\partial s_{i+1}}{\partial x_i} l_{i+1} - \frac{(x_{i+1} - x_i)}{l_{i+1}} s_{i+1} = 0 \\ \frac{\partial s_i}{\partial y_i} l_i + \frac{(y_i - y_{i-1})}{l_i} s_i + \frac{\partial s_{i+1}}{\partial y_i} l_{i+1} - \frac{(y_{i+1} - y_i)}{l_{i+1}} s_{i+1} = 0 \\ \frac{\partial s_i}{\partial z_i} l_i + \frac{(z_i - z_{i-1})}{l_i} s_i + \frac{\partial s_{i+1}}{\partial z_i} l_{i+1} - \frac{(z_{i+1} - z_i)}{l_{i+1}} s_{i+1} = 0, \end{cases} \quad (7)$$

para $i = 1, 2, \dots, N$. De esta forma, el sistema de ecuaciones no lineales (7) describe la discretización en N puntos del rayo de Fermat que va del punto \mathbf{x}^e al punto \mathbf{x}^r . Otra manera de expresar el sistema (7) es la siguiente:

$$F(X) = (f_1(X), \dots, f_{3N}(X))^t = 0, \quad (8)$$

donde $X = (x_1, \dots, x_N, y_1, \dots, y_N, z_1, \dots, z_N)^t \in \mathbb{R}^{3N}$, y

$$\begin{cases} f_i(X) = \frac{\partial s_i}{\partial x_i} l_i + \frac{(x_i - x_{i-1})}{l_i} s_i + \frac{\partial s_{i+1}}{\partial x_i} l_{i+1} - \frac{(x_{i+1} - x_i)}{l_{i+1}} s_{i+1} \\ f_{i+N}(X) = \frac{\partial s_i}{\partial y_i} l_i + \frac{(y_i - y_{i-1})}{l_i} s_i + \frac{\partial s_{i+1}}{\partial y_i} l_{i+1} - \frac{(y_{i+1} - y_i)}{l_{i+1}} s_{i+1} \\ f_{i+2N}(X) = \frac{\partial s_i}{\partial z_i} l_i + \frac{(z_i - z_{i-1})}{l_i} s_i + \frac{\partial s_{i+1}}{\partial z_i} l_{i+1} - \frac{(z_{i+1} - z_i)}{l_{i+1}} s_{i+1}, \end{cases} \quad (9)$$

para $i = 1, 2, \dots, N$.

Particularmente cuando el medio posee velocidad de onda constante, el rayo de Fermat que une el emisor con el receptor es un segmento de recta.

2.1.3 Trazado de rayos en medios con interfaces

Se puede extender la idea explicada anteriormente al trazado de rayos en medios que poseen interfaces. Por simplicidad, se realiza la descripción con un caso en el cual existe una sola discontinuidad en el campo de velocidad, esto es, cuando el rayo cruza una interfaz definida por la superficie $z = \phi(x, y)$, se obtienen dos capas, capa I y capa II, donde la velocidad de onda es distinta en cada capa, v_I y v_{II} , respectivamente (ver Figura 1).

Por medio de los subíndices se diferencian los puntos en las dos capas. El punto de cruce del rayo con la interfaz es denotado como $P_c = (x_c, y_c, z_c)^t$. El siguiente sistema de ecuaciones no lineales se obtiene de (7) y de la notación

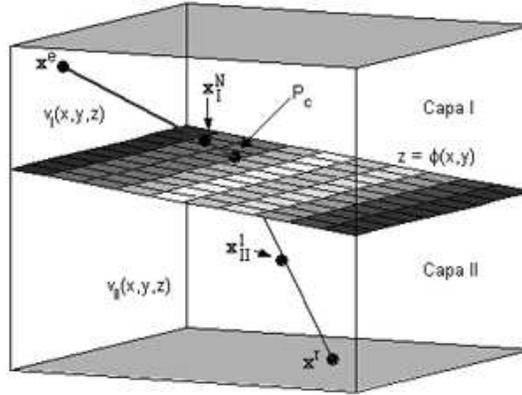


Figura 1: Medio con una interfaz.

descrita anteriormente:

$$\left\{ \begin{array}{l} \frac{\partial s_{I_i}}{\partial x_{I_i}} l_{I_i} + \frac{(x_{I_i} - x_{I_{i-1}})}{l_{I_i}} s_{I_i} + \frac{\partial s_{I_{i+1}}}{\partial x_{I_i}} l_{I_{i+1}} - \frac{(x_{I_{i+1}} - x_{I_i})}{l_{I_{i+1}}} s_{I_{i+1}} = 0 \\ \frac{\partial s_{I_i}}{\partial y_{I_i}} l_{I_i} + \frac{(y_{I_i} - y_{I_{i-1}})}{l_{I_i}} s_{I_i} + \frac{\partial s_{I_{i+1}}}{\partial y_{I_i}} l_{I_{i+1}} - \frac{(y_{I_{i+1}} - y_{I_i})}{l_{I_{i+1}}} s_{I_{i+1}} = 0 \\ \frac{\partial s_{I_i}}{\partial z_{I_i}} l_{I_i} + \frac{(z_{I_i} - z_{I_{i-1}})}{l_{I_i}} s_{I_i} + \frac{\partial s_{I_{i+1}}}{\partial z_{I_i}} l_{I_{i+1}} - \frac{(z_{I_{i+1}} - z_{I_i})}{l_{I_{i+1}}} s_{I_{i+1}} = 0 \\ \frac{\partial s_{II_i}}{\partial x_{II_i}} l_{II_i} + \frac{(x_{II_i} - x_{II_{i-1}})}{l_{II_i}} s_{II_i} + \frac{\partial s_{II_{i+1}}}{\partial x_{II_i}} l_{II_{i+1}} - \frac{(x_{II_{i+1}} - x_{II_i})}{l_{II_{i+1}}} s_{II_{i+1}} = 0 \\ \frac{\partial s_{II_i}}{\partial y_{II_i}} l_{II_i} + \frac{(y_{II_i} - y_{II_{i-1}})}{l_{II_i}} s_{II_i} + \frac{\partial s_{II_{i+1}}}{\partial y_{II_i}} l_{II_{i+1}} - \frac{(y_{II_{i+1}} - y_{II_i})}{l_{II_{i+1}}} s_{II_{i+1}} = 0 \\ \frac{\partial s_{II_i}}{\partial z_{II_i}} l_{II_i} + \frac{(z_{II_i} - z_{II_{i-1}})}{l_{II_i}} s_{II_i} + \frac{\partial s_{II_{i+1}}}{\partial z_{II_i}} l_{II_{i+1}} - \frac{(z_{II_{i+1}} - z_{II_i})}{l_{II_{i+1}}} s_{II_{i+1}} = 0, \end{array} \right. \quad (10)$$

para $i = 1, 2, \dots, N$, donde $\mathbf{x}_I^0 = \mathbf{x}^e$, $\mathbf{x}_I^{N+1} = P_c = \mathbf{x}_{II}^0$ y $\mathbf{x}_{II}^{N+1} = \mathbf{x}^r$. Las primeras $3N$ ecuaciones representan el trozo de rayo de la capa I y las restantes al trozo de rayo de la capa II. Como el punto de cruce del rayo es otra incógnita, se deben incluir tres ecuaciones auxiliares al sistema anterior. Se debe observar que \mathbf{x}_I^N y \mathbf{x}_{II}^1 son los puntos anterior y posterior al punto

P_c , respectivamente (ver Figura 1). Las ecuaciones auxiliares son:

$$\begin{cases} \frac{\partial s_{I_N}}{\partial x_c} l_{I_N} + \frac{(x_c - x_{I_N})}{l_{I_N}} s_{I_N} + \frac{\partial s_{II_1}}{\partial x_c} l_{II_1} - \frac{(x_{II_1} - x_c)}{l_{II_1}} s_{II_1} = 0 \\ \frac{\partial s_{I_N}}{\partial y_c} l_{I_N} + \frac{(y_c - y_{I_N})}{l_{I_N}} s_{I_N} + \frac{\partial s_{II_1}}{\partial y_c} l_{II_1} - \frac{(y_{II_1} - y_c)}{l_{II_1}} s_{II_1} = 0. \end{cases} \quad (11)$$

La última ecuación auxiliar viene dada por la ecuación

$$z_c = \phi(x_c, y_c), \quad (12)$$

que garantiza que el punto P_c pertenece a la interfaz. Esta última ecuación no necesita agregarse al sistema (10), ya que con las ecuaciones (11) se calculan los valores de x_c y y_c , necesarios para calcular el valor de z_c .

Al agregar al sistema (10) las ecuaciones (11) y (12), se obtiene la descripción del rayo que une a los puntos \mathbf{x}^e y \mathbf{x}^r y que atraviesa una interfaz en el punto de cruce P_c . Se puede generalizar cuando se tiene m interfaces. Para ello sólo se repite el procedimiento anterior para las interfaces $\phi_i(x, y)$, $i = 1, 2, \dots, m$, que dividen la región en $m - 1$ capas.

En el modelo del subsuelo se pueden clasificar los rayos de la siguiente manera: un rayo de tipo 1 (C_1) es aquel que atraviesa una sola capa y se refleja en la segunda interfaz, un rayo que atraviesa dos capas y se refleja en la tercera interfaz es un rayo tipo 2 (C_2).

2.1.4 Método espectral para sistemas de ecuaciones no lineales

Sea una función $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ continuamente diferenciable. Con la finalidad de simplificar se denota con $F_k = F(x_k)$ y con $J_k = J(x_k)$ a la matriz Jacobiana de F evaluada en $x_k \in \mathbb{R}^n$.

A continuación se presenta el Algoritmo 1 para la solución del sistema de ecuaciones no lineales el cual ha sido denominado *SANE* (Spectral Approach for Nonlinear Equations) [8].

Algoritmo 1. (*SANE*)

- Dados $x_0 \in \mathbb{R}^n$ y $\alpha_0 > 0$, asignar $k = 0$.
Paso 1: Si $\|F_k\| = 0$, parar el proceso
Paso 2: Si $F_k^t J_k F_k = 0$, parar el proceso
Paso 3: Asignar $d_k = -\text{sgn}(F_k^t J_k F_k) F_k$
Paso 4: $x_{k+1} = x_k + (1/\alpha_k) d_k$
Paso 5: $y_k = F_{k+1} - F_k$
Paso 6: $\alpha_{k+1} = \text{sgn}(F_k^t J_k F_k) \left(\frac{\alpha_k d_k^t y_k}{d_k^t d_k} \right)$.

2.1.5 SANE globalizado

El algoritmo SANE globalizado (SANEG) es el resultado de combinar el algoritmo SANE con una técnica de globalización [8]. Se utiliza como técnica de globalización una variación de la búsqueda lineal empleada por Raydán [11] en minimización irrestricta:

$$f(x_k + \lambda d_k) \leq \max_{0 \leq j \leq \min(k, M)} [f(x_{k-j})] + 2\gamma \lambda F_k^t J_k d_k. \quad (13)$$

donde $f(x) = \|F(x)\|_2^2$, $d_k = -\text{sgn}(F_k^t J_k F_k) F_k$, $\gamma \in (0, 1)$ y M es un entero no negativo.

Algoritmo 2. (SANEG)

Sean $\alpha_0 > 0$, un entero $M > 0$, $\gamma > 0$, $0 < \sigma_1 < \sigma_2 < 1$, $0 < \varepsilon < 1$, y $\delta \in [\varepsilon, 1/\varepsilon]$.

Tomar un punto inicial $x_0 \in \mathbb{R}^n$ y asignar $k = 0$.

Paso 1: Si $\|F_k\| = 0$, parar el proceso

Paso 2: Si $|F_k^t J_k F_k| / F_k^t F_k < \varepsilon$, parar el proceso

Paso 3: Si $\alpha_k \leq \varepsilon$ o $\alpha_k \geq 1/\varepsilon$, entonces asignar $\alpha_k = \delta$

Paso 4: Tomar $d_k = -\text{sgn}(F_k^t J_k F_k) F_k$

Paso 5: Asignar $\lambda = 1/\alpha_k$

Paso 6: Si $f(x_k + \lambda d_k) \leq \max_{0 \leq j \leq \min(k, M)} [f(x_{k-j})] + 2\gamma \lambda F_k^t J_k d_k$, ir al

Paso 8

Paso 7: Escoger $\sigma \in [\sigma_1, \sigma_2]$, asignar $\lambda = \sigma \lambda$ e ir al Paso 6

Paso 8: Asignar $\lambda_k = \lambda$, $x_{k+1} = x_k + \lambda_k d_k$, $y_k = F_{k+1} - F_k$

Paso 9: Asignar $\alpha_{k+1} = \text{sgn}(F_k^t J_k F_k) \left(\frac{d_k^t y_k}{\lambda_k d_k^t d_k} \right)$, hacer $k = k + 1$ e ir al Paso 1.

3 Idea y algoritmo propuesto

En esta sección se describe el algoritmo secuencial y su respectiva versión paralela para resolver el problema de trazado de rayos sísmicos.

3.1 Algoritmo secuencial para el trazado de rayos sísmicos

Para el algoritmo secuencial se indican los parámetros de entrada, de salida y un esquema detallado de su funcionalidad.

La versión secuencial consiste en calcular todos los rayos para cada par emisor-receptor (ver figura 2) de la siguiente forma: se encuentra una trayectoria inicial del rayo y se utiliza como iterado inicial para resolver el sistema de ecuaciones no lineales; una vez hallada la trayectoria final del rayo, se calcula el tiempo de viaje.

Parámetros de entrada:

- Número de receptores ($nrec$) y sus coordenadas.
- Número de emisores ($nemi$) y sus coordenadas.
- Número de capas ($ncap$).
- Función de velocidad por capa y su derivada.
- Funciones que definen a las interfaces.
- Número de puntos para aproximar el rayo.

Parámetros de salida:

- Vector que contiene los puntos que determinan la trayectoria de cada rayo.
- Tiempos de viaje.

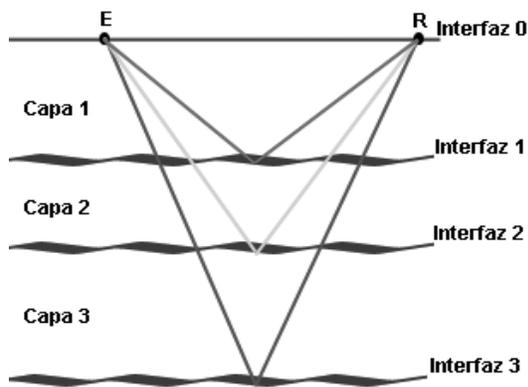


Figura 2: Rayos para un par emisor-receptor.

Algoritmo 3. (Versión secuencial)

Para $e = 1$ **hasta** $nemi$ **hacer**
(se calculan los rayos asociados al emisor e)
Para $r = 1$ **hasta** $nrec$ **hacer**
(se calculan los rayos asociados al par emisor e y receptor r)
Para $c = 1$ **hasta** $ncap$ **hacer**
(se calcula el rayo correspondiente al emisor e y receptor r que atraviesan c capas)

- Obtener la trayectoria inicial del rayo
- Obtener la trayectoria aproximada del rayo resolviendo el sistema de ecuaciones no lineales con el algoritmo SANEG
- Calcular el tiempo de viaje

Fpara
Fpara
Fpara

3.2 Algoritmo paralelo para el trazado de rayos sísmicos

Existen muchas razones que justifican la implementación de una versión paralela que permita mejorar el desempeño del algoritmo secuencial ya mencionado. La primera razón, es la complejidad del modelo sintético del subsuelo a utilizar. Este modelo está conformado por las interfaces que determinan las capas y las funciones que representan a la velocidad del medio en cada capa, al igual que las derivadas de estas funciones. En segundo lugar, la cantidad de pares emisores-receptores, que junto con el número de capas del modelo, determinan de manera directa la cantidad de rayos a calcular. Por último, se tiene que por cada trayectoria del rayo a determinar, se debe resolver un sistema de ecuaciones no lineales cuya dimensión varía en función de las capas que atravesase el mismo.

En este caso el paralelismo a implementar es el que sugiere de forma natural el problema de trazado de rayos sísmicos. La trayectoria de cada rayo del modelo puede ser calculada de forma independiente; esto permite que cada proceso determine la trayectoria de un conjunto de rayos, sin que sea necesaria comunicación alguna con otros procesos. En consecuencia se pueden obtener más rayos en menos tiempo de CPU.

Para ello se debe tomar en cuenta un aspecto muy importante al momento de distribuir los rayos entre los procesos involucrados, de forma que la carga de trabajo esté lo mejor balanceada posible. Este aspecto es la dimensión del sistema de ecuaciones no lineales que se debe resolver al calcular la trayectoria

final del rayo, que viene dada por $2 * ncap * (3N + 2) - 2$ donde N es el número de puntos utilizados para aproximar el rayo entre dos interfaces y $ncap$ es el número de capas presente en el modelo.

El mecanismo para distribuir los datos entre los procesos es el siguiente:

- Se establecen unidades de datos conformadas por un par emisor-receptor; por cada una de estas unidades se calculan $ncap$ rayos donde $ncap$ es el número de capas del modelo.
- Se divide la cantidad de unidades de datos entre el número de procesos involucrados, a fin de obtener el número de unidades que le corresponde a cada proceso.
- Las unidades de datos restantes se subdividen en rayos, a estos se le asigna un peso dependiendo del número de capas que atraviese el rayo.
- Los rayos son distribuidos entre los procesos de manera equitativa en función del peso asignado.

Para una mejor comprensión de los pasos de este mecanismo de distribución, se detalla el siguiente ejemplo simplificado. Suponga un modelo sintético del subsuelo que posee tres capas, un punto emisor y tres receptores; esto implica que se tienen tres unidades de datos (ver figura 3) y por cada una de ellas se deben calcular tres rayos. Además, suponga que se cuenta con dos procesos. El número de unidades de datos que le corresponden a cada proceso es simplemente el número total de unidades de datos entre el número de procesos. Para el ejemplo dado, el resultado es uno, es decir, a cada proceso se le asigna una unidad de datos. Ahora bien, se puede observar que falta una unidad de datos por distribuir la cual se divide en rayos y se le asigna un peso para luego ser repartido. Al primer rayo se le asigna peso uno, al segundo peso dos y así sucesivamente; esta designación obedece a que la dimensión del sistema para calcular la trayectoria del segundo rayo es prácticamente el doble que la dimensión del sistema para el primer rayo.

Parámetros de entrada:

- Número de procesos.
- Cantidad de pares emisores receptores que corresponden a cada proceso.

Parámetros de salida:

- Coordenadas de los emisores y receptores correspondientes al proceso p .

Se debe resaltar que estos parámetros son adicionales a los presentados en la sección anterior del algoritmo secuencial. Además, se tienen n procesos donde el proceso 0 es el proceso coordinador.

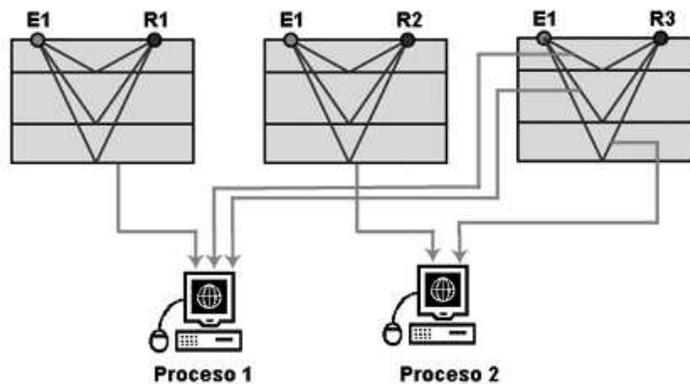


Figura 3: Forma de distribución del algoritmo 4

Algoritmo 4. (Versión Paralela)

Si ($ident = 0$) entonces

- Calcular la cantidad de pares emisor-receptor que le corresponde a cada proceso ($cant_e_r$)

Para $p = 1$ hasta $nproc$ hacer

- Enviar a cada proceso la información requerida para calcular la trayectoria de los rayos

Fpara

(*el proceso coordinador obtiene sus rayos*)

Para $er = 1$ hasta $cant_e_r$ hacer

(*se calcula el rayo correspondiente al par emisor e y receptor r que atraviesan c capas*)

- Obtener el par emisor e y receptor r
- Obtener la trayectoria inicial del rayo
- Obtener la trayectoria aproximada del rayo resolviendo el sistema de ecuaciones no lineales con el algoritmo SANEG

Fpara

Para $p = 1$ hasta $nproc$ hacer

- Recibir los emisores y receptores y el vector con la trayectoria de los rayos provenientes del proceso p

Fpara

Sino (*el resto de los procesos*)

- Recibir del proceso coordinador la información

```

    para calcular los rayos (el proceso obtiene sus rayos)
Para  $er = 1$  hasta  $cant\_e\_r$  hacer
    (se calcula el rayo correspondiente al par emisor  $e$  y receptor  $r$  que
    atraviesan  $c$  capas)
    - Obtener el par emisor  $e$  y receptor  $r$ 
    - Obtener la trayectoria inicial del rayo
    - Obtener la trayectoria aproximada del rayo
      resolviendo el sistema de ecuaciones no lineales
      con el algoritmo SANEG
Fpara
    - Enviar al proceso coordinador los tiempos de viaje
      calculados
Fsi

```

4 Resultados Numéricos

En esta sección se presentan los resultados numéricos obtenidos de los experimentos realizados por medio de tablas y gráficas, las cuales permiten comparar el costo de CPU de las versión paralela presentada en este trabajo con las versión secuencial. Los datos de prueba fueron suministrados por Z. Castillo, W. La Cruz y A. Reyes.

4.1 Ambiente de desarrollo

El hardware utilizado para la implementación fue un cluster de 12 nodos y un nodo maestro, perteneciente al Departamento de Computación de la Facultad de Ciencias y Tecnología (FACyT), Universidad de Carabobo. Cada nodo posee un procesador Pentium 4, 1.8 GHz, memoria de 512 Mb y un disco duro de 30 Gb. El software utilizado en la implementación de los algoritmos es el siguiente: lenguaje Fortran 77, sistema operativo Red Hat Linux y Estándar MPI (Message Passing Interface). Es importante destacar que tanto la versión secuencial como la versión paralela se compilaron y ejecutaron en el cluster ya mencionado, para así poder comparar los tiempos de CPU.

4.2 Modelos de prueba

Para la evaluación de los algoritmos presentados en este trabajo se crearon 4 modelos. En cada uno de los modelos se ubicaron $nemi$ emisores con coordenadas $(100 * i, 200 * i, 0)$, $i = 0, \dots, nemi - 1$, y $nrec$ receptores con

coordenadas $(1000 * i, 2000 * i, 0)$, $i = 0, \dots, nrec - 1$. El número de puntos N que se empleó para describir el rayo por capas es $N = 20$.

Los parámetros utilizados para el método SANEG son los siguientes: $\gamma = 10^{-4}$, $\varepsilon = 10^{-10}$, $\sigma_1 = 0,1$, $\sigma_2 = 0,5$, $\alpha_0 = 1$, $M = 10$. El escalar $F_k^t J_k F_k$ se aproxima empleando la siguiente fórmula [7] :

$$F_k^t J_k F_k = F_k^t \left[\frac{F(x_k + hF_k) - F_k}{h} \right], \text{ con } h = 10^{-7}.$$

Para la escogencia de λ , se usa un modelo parabólico de dos puntos (Kelley [6, pp. 142-143]).

La notación utilizada en las tablas de resultados es la siguiente:

- **E-R:** Cantidad de pares Emisor - Receptor.
- **p:** Número de procesos.

Un aspecto importante que se debe tener en cuenta para los experimentos realizados es que estos se llevaron a cabo en un cluster que no es de uso exclusivo, por lo que se realizaron 10 ejecuciones para cada modelo y se calculó un promedio con los resultados obtenidos, esto con la finalidad de darle credibilidad a los resultados.

Primer modelo de prueba: Consta de diez capas con velocidad constante en cada capa e interfaces planas paralelas a la superficie (ver figura 4).

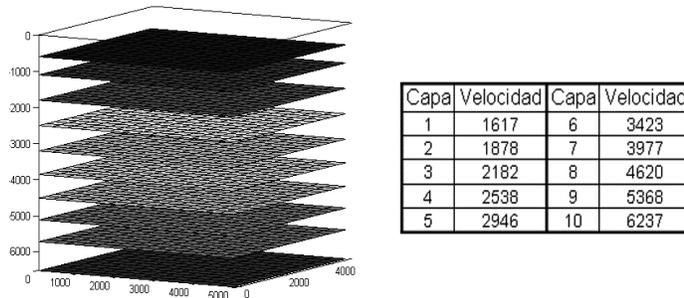


Figura 4: Modelo interfaces planas (10 capas).

Segundo modelo de prueba: Formado por cinco capas con velocidad variable en cada capa e interfaces planas con diferente inclinación (ver figura 5).

Tercer modelo de prueba: Posee cinco capas con velocidad variable en cada capa e interfaces con relieve (ver figura 6).

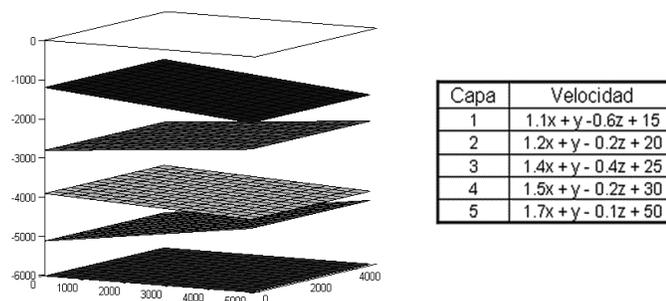


Figura 5: Modelo de interfaces inclinadas.

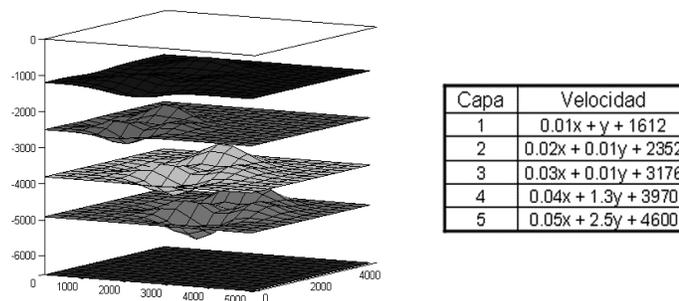


Figura 6: Modelo de interfaces con relieve.

Cuarto modelo de prueba: Consta de cinco capas con velocidad variable en cada capa y una combinación de interfaces planas e interfaces con relieve (ver figura 7).

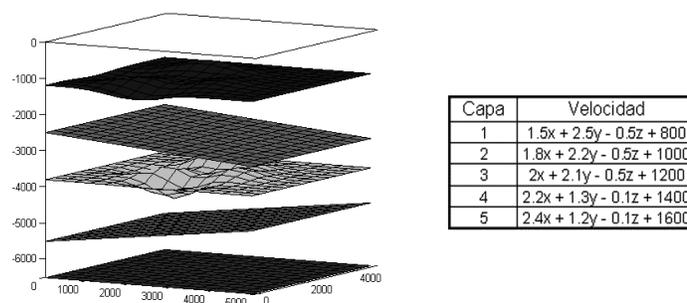


Figura 7: Combinación de interfaces.

4.3 Resultados de los experimentos

Los siguientes resultados corresponden a la versión paralela. Los datos mostrados en las tablas corresponden con el tiempo de CPU tanto secuencial como paralelo para 4, 8 y 12 procesos. En las gráficas se observa el comportamiento del factor de aceleración con respecto al factor de aceleración ideal.

Primer modelo: interfaces planas (10 capas).

E-R	Tiempo secuencial	Tiempo paralelo		
		p=4	p=8	p=12
100	12,64	6,02	3,19	1,71
200	39,99	17,06	9,48	5,94
400	204,56	76,99	36,89	22,59
800	533,03	246,59	123,68	63,12
1600	1758,77	952,79	326,24	180,32
3200	4946,95	2353,06	1210,53	525,67
6400	10820,13	5651,68	2311,40	1166,92

Cuadro 1: Tiempo de ejecución para el primer modelo.

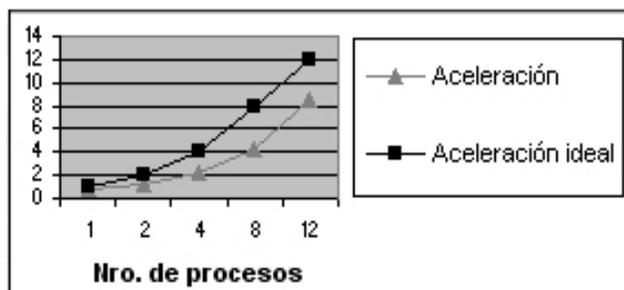


Figura 8: Resultados para el primer modelo con 800 pares emisor-receptor (8000 rayos).

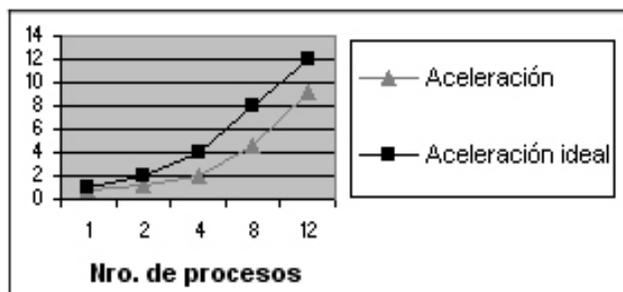


Figura 9: Resultados para el primer modelo con 6400 pares emisor-receptor (64000 rayos).

Segundo modelo: interfaces inclinadas

E-R	Tiempo secuencial	Tiempo paralelo		
		p=4	p=8	p=12
100	11,52	5,97	2,95	1,56
200	22,46	11,01	5,57	3,01
400	40,97	19,19	9,85	5,34
800	80,86	36,86	18,89	10,44
1600	238,31	106,39	54,38	30,49
3200	546,78	233,04	119,53	67,22
6400	1345,23	534,02	291,14	163,59

Cuadro 2: Tiempo de ejecución para el segundo modelo.

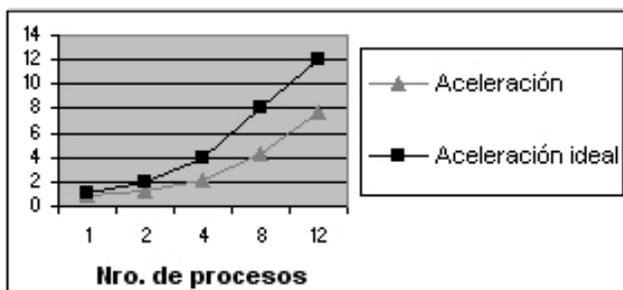


Figura 10: Resultados para el segundo modelo con 800 pares emisor-receptor (4000 rayos).

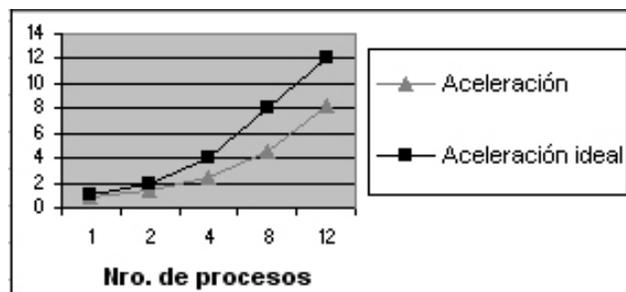


Figura 11: Resultados para el segundo modelo con 6400 pares emisor-receptor (32000 rayos).

Tercer modelo: interfaces con relieve.

E-R	Tiempo secuencial	Tiempo paralelo		
		p=4	p=8	p=12
100	8,37	4,19	2,12	1,05
200	26,01	11,98	6,34	3,24
400	98,26	43,57	23,38	11,94
800	331,84	141,94	76,65	39,36
1600	710,54	286,87	159,01	83,13
3200	1698,25	645,21	366,21	191,96
6400	3175,02	1128,21	673,21	345,25

Cuadro 3: Tiempo de ejecución para el tercer modelo.

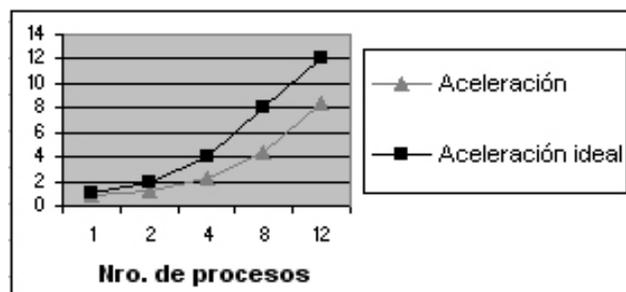


Figura 12: Resultados para el tercer modelo con 800 pares emisor-receptor (4000 rayos).

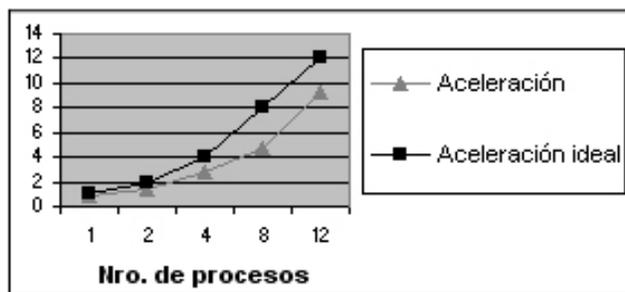


Figura 13: Resultados para el tercer modelo con 6400 pares emisor-receptor (32000 rayos).

Cuarto modelo: combinación de interfaces.

E-R	Tiempo secuencial	Tiempo paralelo		
		p=4	p=8	p=12
100	5,01	2,56	1,28	0,64
200	19,01	9,01	4,81	2,42
400	58,159	25,89	10,19	7,31
800	207,25	86,86	39,95	25,74
1600	530,39	215,50	99,57	64,58
3200	1188,52	445,07	271,24	141,53
6400	2180,70	785,36	468,47	252,82

Cuadro 4: Tiempo de ejecución para el cuarto modelo.

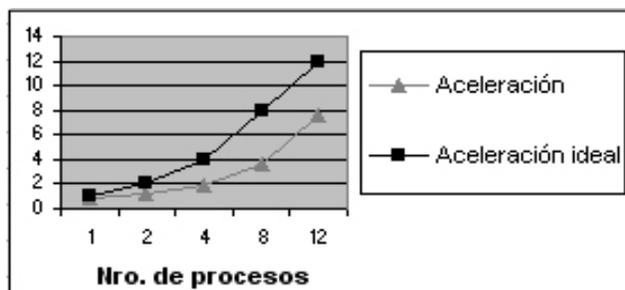


Figura 14: Resultados para el cuarto modelo con 800 pares emisor-receptor (4000 rayos).

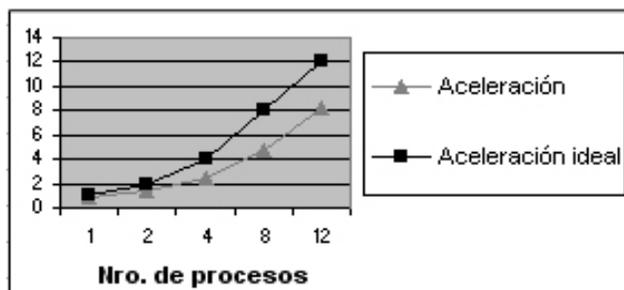


Figura 15: Resultados para el cuarto modelo con 6400 pares emisor-receptor (32000 rayos).

Al comparar la versión secuencial con la versión paralela, se puede observar que el tiempo de CPU varía dependiendo del modelo que se use como dato de entrada para el cálculo de los rayos; este modelo viene determinado por el número y tipo de interfaces y las funciones que describen la velocidad de onda en cada capa. En el caso del primer modelo con interfaces planas, donde las velocidades se incrementan en la medida que la capa se encuentre a una mayor profundidad (ver figura 4), trae como consecuencia que el método SANEG realice más iteraciones para encontrar la trayectoria del rayo final. Estos resultados se muestran en la tabla 1 y las figuras 8 y 9, donde pueden observarse que los tiempos de cómputo son mayores, que los obtenidos con los otros modelos de prueba. Cuando las interfaces del modelo poseen relieves también se produce un aumento en el tiempo de cómputo, esto se puede apreciar en el tercer modelo (ver figura 6), cuyos resultados son presentados en la tabla 3 y las figuras 12 y 13.

Los mejores resultados se observan, cuando el modelo del subsuelo utilizado es más complicado, en cuanto a las interfaces y las funciones de velocidad por capa, que lo definen. El primer y tercer modelo son los modelos más complejos de los cuatro modelos presentados; el primer modelo contempla velocidades que crecen de forma exponencial, mientras que el tercer modelo está conformado por interfaces con relieve. En estos casos se nota que el número de iteraciones requerido para encontrar la solución del sistema de ecuaciones no lineales, con el método SANEG, es mucho mayor en comparación con los otros modelos, incrementándose el tiempo de CPU. Esta es la razón por la que en este modelo se obtuvieron mejores resultados con la versión paralela, ya que se aprovecha mejor el trabajo de los procesos en el cálculo de la trayectoria de cada uno de los rayos, requiriéndose el mismo tiempo de comunicación, que

consumieron al calcular los rayos en los otros modelos.

En general, la versión paralela se ejecuta en aproximadamente el 15 % del tiempo de cómputo requerido por el algoritmo secuencial utilizando 12 procesos y el máximo factor de aceleración obtenido está cercano a 10. Aunque, es claro ver en las gráficas donde se muestra el factor de aceleración, que para ninguno de los modelos se logró obtener la aceleración ideal que debe ser igual al número de procesos utilizados en la ejecución del algoritmo. Esto debido al tiempo empleado por los procesos para recibir los datos provenientes del proceso coordinador y el tiempo utilizado por éstos, para enviar los resultados obtenidos una vez culminado el procesamiento de los datos. A pesar de esto, los resultados obtenidos son bastante satisfactorios, tomando en cuenta que el algoritmo de trazado de rayos es ejecutado tantas veces como sea necesario, para obtener un modelo aproximado del subsuelo.

5 Conclusiones

Se desarrolló una versión paralela para el algoritmo de trazado de rayos, que consistió en distribuir los pares emisor-receptor entre los procesos de forma equitativa y cada proceso calcula los rayos correspondientes con sus datos y los respectivos tiempos de viaje.

En los resultados de los experimentos numéricos realizados para la versión paralela, se observó en promedio una disminución del tiempo de cómputo de un 85 %, conservándose la misma calidad en la solución que la versión secuencial. El máximo factor de aceleración logrado está cercano a 10, empleando 12 procesos. A pesar, que no es la aceleración ideal, los resultados son excelentes.

Es importante observar, que cualquier reducción en el tiempo de cómputo del algoritmo de trazado de rayos, es significativa, ya que el trazador es una de las etapas que consume mayor tiempo en el problema de Tomografía sísmica; además de que este algoritmo es invocado tanta veces como sea necesario para construir el modelo sintético deseado.

Referencias

- [1] Z. Castillo, H. Laham, N. Merentes, B. Molina. *Un Algoritmo Paralelo del Problema de Tomografía de Reflexión Sísmica*. XIV Jornadas de Matemáticas, Barquisimeto, Venezuela, 2001.
- [2] Z. Castillo, D. Cores, M. Raydán. *Low Cost Optimization Techniques for Solving the Nonlinear Seismic Reflection Tomography Problem*. *Optimization and Engineering*, 1:155–169, 2000.

- [3] D. Cores, G. Fung, J. Maguregui. *A new 3D ray tracing using nonlinear optimization techniques*. IV International Conference on Approximation and Optimization in the Caribbean, Caracas, Venezuela, 1997.
- [4] F. Jurado, P. Lailly, A. Ehinger. Fast 3D two-point ray tracing for travel-time tomography. *SPIE proceedings Mathematical Methods in Geophysical Imaging V*, 3453:70–81, 1998.
- [5] F. Jurado, C. Monestié. New functionalities in the bending ray tracing jerry. *KIM 1998 Annual Report, Institut Franais du Pétrole, Pau, France*.
- [6] C. T. Kelley. *Iterative Methods for Linear and Nonlinear Equations*. SIAM, Philadelphia, 1995.
- [7] W. La Cruz. *Enfoque Espectral para Ecuaciones No Lineales de Gran Escala, y una Aplicación en Geofísica*. Tesis Doctoral, Universidad Central de Venezuela, 2004.
- [8] W. La Cruz, M. Raydán. Nonmonotone spectral methods for large-scale nonlinear systems. *Optimization Meth & Soft*, 18:583–599, 2003.
- [9] V. Pereyra, W. H. Lee, H. B. Keller. Solving two-point seismic-ray tracing problems in a heterogeneous medium. *Bulletin of the Seismological of America*, 70(1):79–99, 1980.
- [10] W. Prothero, W. J. Taylor, J. A. Eickemeyer. A fast two-point three-dimensional ray tracing algorithm using a simple search method. *Bull. Seismol. Soc. Am.*, 78:1190–1198, 1988.
- [11] M. Raydán. The Barzilai and Borwein Gradient Method for the large Scale Unconstrained Minimization Problems. *SIAM*, 7(1):26–33, 1997.
- [12] Jr. W. A. Scheneider, K. A. Razinger, A. H. Blach, C. Kruse. A dynamic programming approach to first arrival computation in media with arbitrarily distributed velocities. *Geophysics*, 57:39–50, 1992.