



UNIVERSIDAD SIMÓN BOLÍVAR

Decanato de Estudios Profesionales

Coordinación de Matemáticas

ESTIMACIÓN DE VELOCIDADES E INTERFACES USANDO DATOS SÍSMICOS Y TÉCNICAS DE OPTIMIZACIÓN NUMÉRICA

Por

Erika Patricia Gomes Goncalves

Realizado con la asesoría de

Prof. Oswaldo Jiménez

Prof. Débora Cores Carrera

Proyecto de Grado

Presentado ante la Universidad Simón Bolívar

Como requisito para optar al título de

Licenciado en Matemáticas

Opción

Estadística y Matemáticas Computacionales

Sartenejas, Noviembre 2006



UNIVERSIDAD SIMÓN BOLÍVAR
COORDINACIÓN DE MATEMÁTICA

VEREDICTO

Veredicto sobre el Proyecto de Grado "**ESTIMACIÓN DE VELOCIDADES E INTERFACES USANDO DATOS SÍSMICOS y TÉCNICAS DE OPTIMIZACIÓN NUMÉRICA**", presentado por **ERIKA PATRICIA GOMES GONCALVES**, como requisito para optar al título de LICENCIADO EN MATEMÁTICAS, OPCIÓN ESTADÍSTICA y MATEMÁTICAS COMPUTACIONALES.


Quienes suscribimos, profesores **Bernardo Feijoo, René Escalante y Oswaldo Jiménez**, miembros del jurado designado por la Coordinación de Matemáticas de la Universidad Simón Bolívar, para considerar y evaluar el Proyecto de Grado referido en el anterior párrafo, dejamos constancia de lo siguiente:

Emitimos el veredicto de **APROBADO** al Trabajo de Grado sometido a nuestra consideración. En este Proyecto se presenta un desarrollo computacional importante que resuelve el problema de la determinación de las velocidades de propagación de las ondas sísmicas en el subsuelo, así como la determinación de las ubicaciones de las interfaces de las capas que lo forman, mediante el uso de dos herramientas fundamentales, como lo son el método de optimización numérica conocido como Gradiente Espectral Proyectado y una herramienta de trazado de rayos que calcula los tiempos de viaje de las ondas sísmicas. La metodología propuesta usó desarrollos computacionales previos de ambas herramientas, las cuales fueron adaptadas y combinadas para obtener un programa funcional y eficiente, el cual arrojó resultados numéricos satisfactorios que fueron claramente expuestos por la Br. Gomes. Asimismo, la Br. Gomes demostró dominio suficiente de los temas relacionados con su trabajo, tales como la optimización no lineal numérica y el trazado de rayos o tomografía sísmica.

En fe de todo lo cual levantamos y firmamos el presente veredicto en el Valle de Sartenejas, a los nueve días del mes de noviembre de dos mil seis.



Bernardo Feijoo



René Escalante



Oswaldo Jiménez

Estimación de velocidades e interfaces usando datos sísmicos y técnicas de optimización numérica

Por

Erika Patricia Gomes Goncalves

Resumen

El problema de inversión o tomografía de tiempos de viaje es de interés en la industria petrolera, ya que su resolución permite definir las diferentes velocidades y estructuras que conforman el subsuelo, y de esa manera contribuir en forma eficiente y precisa, en la localización de posibles entrapamientos de petróleo.

Para resolver el problema es necesario hallar inicialmente, de una manera eficiente, los tiempos de viaje de los rayos que van desde un punto fuente a un punto receptor, ambos en la superficie del subsuelo, para luego obtener las velocidades y desplazamientos de las interfaces como solución de un problema de optimización no lineal. Para ello se utilizó el método del gradiente espectral proyectado por ser un método de bajo costo computacional, fácil de implementar y muy efectivo para problemas de gran escala.

En este trabajo se hacen algunas consideraciones que permiten simplificar el problema a ser resuelto, como por ejemplo que el medio existente entre cada interfaz es isótropo y homogéneo, que las trayectorias que siguen las ondas sísmicas son segmentos de recta y que las capas a considerar son planas.

La implantación utilizada para la resolución del problema requirió de la adaptación de dos códigos dados en la referencia, uno correspondiente al Trazador de Rayos, el cual se utilizó para el cálculo de los tiempos de viaje, y otro para el Método del Gradiente Espectral Proyectado, que sirvió para la estimación de las velocidades y desplazamientos verticales del modelo inicial del subsuelo dado.

Se presentan resultados numéricos que muestran que la metodología y la implantación propuestas son firmes y eficientes para resolver el problema de estimar las velocidades sísmicas y los parámetros que definen las interfaces que determinan las capas del subsuelo. Los resultados obtenidos fueron lo suficientemente satisfactorios al compararlos con los logrados en un trabajo precedente a éste.

Palabras claves: Trazado de Rayos, Optimización no lineal, Gradiente Espectral Proyectado

ÍNDICE GENERAL

INTRODUCCIÓN	1
1. CAPÍTULO 1: PLANTEAMIENTO DEL PROBLEMA	3
2. CAPÍTULO 2: GRADIENTE ESPECTRAL PROYECTADO	6
2.1. Método del Gradiente Espectral Proyectado	6
2.2. Algoritmo del Gradiente Espectral Proyectado	8
3. CAPÍTULO 3: TRAZADOR DE RAYOS 3-D	10
3.1. La función de tiempos de viaje	10
3.2. Descripción de las interfaces	11
4. CAPÍTULO 4: METODOLOGÍA	12
4.1. Matriz Jacobiana	13
4.2. Descripción de las interfaces	21
4.3. Pasos a seguir	21
5. CAPÍTULO 5: IMPLANTACIÓN	23
6. CAPÍTULO 6: RESULTADOS NUMÉRICOS	25
6.1. Modelo 1.....	26
6.2. Modelo 2.....	31
6.3. Modelo 3.....	35
6.4. Modelo 4.....	39
7. CAPÍTULO 7: CONCLUSIONES Y RECOMENDACIONES	41

8. REFERENCIAS BIBLIOGRÁFICAS	43
 A. APÉNDICE A	 45
A.1. Trazador de Rayos.....	45
A.2. Optimizador (Gradiente Espectral Proyectado).....	47
A.2.1 Evaluación del Funcional (EVALF).....	48
A.2.2 Evaluación del Gradiente (EVALG).....	49
A.2.3 Proyección (PROJ).....	49
A.3. Jacobiano (Módulo fun_jac).....	50
A.4. Herramientas (Módulo Herramientas)	51
A.4.1 Cotas.....	51
A.4.2 Errores.....	51
A.4.3 Actualizar	51

ÍNDICE DE FIGURAS

1.1 Modelo parcial de la estructura del subsuelo.....	3
4.1 Modelaje sísmico completo.....	14
4.2 Modelo de ni capas, 1 receptor y 1 fuente.....	15
4.3 Modelo de 1 capa, ns receptores y nr fuentes.....	20
4.4 Modelo de ni capas, ns receptores y nr fuentes.....	20

ÍNDICE DE TABLAS

6.1 Parámetros GEP.....	25
6.2 Velocidades e interfaces reales para el modelo 1.....	26
6.3 Punto inicial de velocidades y desplazamientos verticales para el modelo 1	27
6.4 Punto solución de velocidades para el modelo 1.....	27
6.5 Punto solución de velocidades e interfaces para el modelo 1.....	29
6.6 Convergencia de los puntos solución de velocidades e interfaces para el modelo 1.....	30
6.7 Velocidades e interfaces reales para el modelo 2.....	31
6.8 Punto inicial de velocidades y desplazamientos verticales para el modelo 2	32
6.9 Punto solución de velocidades para el modelo 2.....	32
6.10 Punto solución de velocidades e interfaces para el modelo 2.....	33
6.11 Convergencia de los puntos solución de velocidades e interfaces para el modelo 2.....	34
6.12 Velocidades e interfaces reales para el modelo 3.....	35
6.13 Punto inicial de velocidades y desplazamientos verticales para el modelo 3	36
6.14 Punto solución de velocidades para el modelo 3.....	36
6.15 Punto solución de velocidades e interfaces para el modelo 3.....	38

6.16 Convergencia de los puntos solución de velocidades e interfaces para el modelo 3.....	38
6.17 Velocidades e interfaces reales para el modelo 4.....	39
6.18 Punto inicial de velocidades y desplazamientos verticales para el modelo 4.....	39
6.19 Punto solución de velocidades para el modelo 4.....	40
6.20 Convergencia para el modelo 4.....	40
A.1 Valores que puede tomar <i>inform</i>	48
A.2 Porcentajes para los valores de V y H	49

INTRODUCCIÓN

El problema considerado en este trabajo es de interés en la industria petrolera, ya que su resolución permite definir las diferentes velocidades y estructuras que conforman el subsuelo, y de esa manera contribuir en forma eficiente y precisa, en la localización de posibles entrapamientos de petróleo. Para ello es necesario obtener, de manera sencilla y rápida, los tiempos de viaje de los rayos (ondas sísmicas) que van desde un punto fuente a un punto receptor, ambos en la superficie del suelo, así como la trayectoria que siguen dichos rayos.

Una de las técnicas más aplicables a los problemas de exploración petrolera es el trazado de rayos, que consiste en tomar un modelo inicial de velocidades y de la estructura del suelo, para encontrar la trayectoria y los tiempos de viaje de los rayos que parten de ciertos puntos (fuentes) hacia otros (receptores), reflejándose en un particular modelo de capas en el subsuelo [7].

En el problema de inversión o tomografía de tiempo de viaje que se quiere resolver en este trabajo, se desea encontrar el vector de velocidades de propagación de la onda y el vector de desplazamientos verticales de las diferentes interfaces del subsuelo, a partir de los tiempos de viaje obtenidos mediante el trazado de rayos, ya mencionado. Para ello se propone utilizar un método de bajo costo computacional, fácil de implementar y muy efectivo para problemas de gran escala, conocido como gradiente espectral proyectado (GEP), el cual está basado en la dirección del gradiente negativo proyectado y la longitud del paso espectral [1].

El presente trabajo está organizado en capítulos de la siguiente manera, en el capítulo 1, se presenta el planteamiento del problema y los fundamentos del mismo. En los capítulos 2 y 3, se describe el método del gradiente espectral proyectado y su algoritmo, junto con las especificaciones del trazador de rayos. Los capítulos 4 y 5 exponen la metodología usada y la implementación propuesta para la resolución del problema. El capítulo 6, presenta ejemplos y resultados computacionales junto con la

discusión de los mismos. Finalmente, el último capítulo contiene las conclusiones y recomendaciones. Adicionalmente se dispone de un apéndice en el cual se especifican los códigos creados y modificados a partir de los datos en las referencias, los cuales hicieron posible la resolución del problema planteado.

CAPÍTULO 1

PLANTEAMIENTO DEL PROBLEMA

Dado un modelo 3D del subsuelo, el problema consiste en determinar el vector de velocidades de propagación de la onda en el subsuelo y las superficies que determinan las diferentes estructuras del mismo. Para ello, contamos con datos sísmicos provenientes del campo, los cuales se obtienen al propagarse una onda sísmica en la superficie del suelo, por medio de unos instrumentos llamados fuentes.

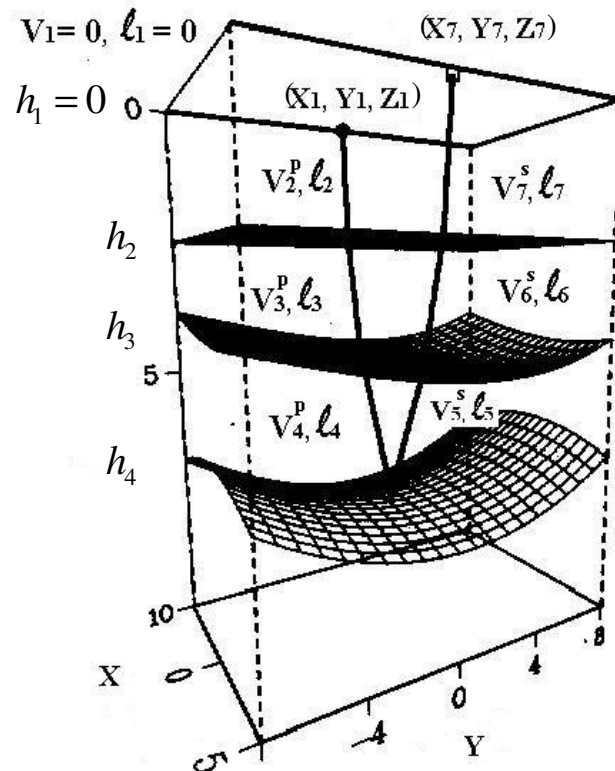


Fig. 1.1: Modelo parcial de la estructura del subsuelo.

- Fuente
- Receptor

Esta onda sísmica es percibida o recibida por diferentes puntos en la superficie llamados geófonos o receptores. La figura 1.1 describe este proceso [5]. Los segmentos de rectas entre el par fuente receptor se denomina rayo y el tiempo que tarda la onda en viajar de la fuente $x_s = (x_1, y_1, z_1)$ al receptor $x_r = (x_7, y_7, z_7)$ se conoce como tiempo de viaje.

El problema se describe matemáticamente como:

Dados los vectores de tiempo de viaje T_R , que tarda la onda sísmica en propagarse entre cada par fuente receptor de un conjunto de fuentes y receptores, se desea encontrar el vector de velocidades de propagación de la onda V y el vector de desplazamientos verticales de las diferentes interfaces del subsuelo H que minimice la diferencia entre T_R y el vector de tiempos de viaje que depende de V y H . Esto es

$$\psi(V, H) = \min \frac{1}{2} \|T_R - T(V, H)\|_2^2 \quad (1.1)$$

$$(V, H) \in \Re^{(2ni+1, ni+1)}$$

donde

$$T(V, H) = \min_{(x, y, z)} \sum_{i=2}^{2ni+1} \frac{\sqrt{[(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2]}}{v_i} = \sum_{i=2}^{2ni+1} \frac{\ell_i}{v_i} \in \Re^n \quad (1.2)$$

con:

$ni :$	El número de capas.
$n :$	El número de rayos n es el producto del número de fuentes ns , el número de receptores nr y el número de capas ni .
$T = (T_1, \dots, T_n)^T :$	Vector de tiempos de viaje, con n el número de rayos.
$(x_i, y_i, z_i) :$	Coordenadas de los puntos extremos de un segmento de rayo, en una capa k , con $k = 1, \dots, ni$

	$X = (x_1, \dots, x_{2k+1}) \in \mathfrak{R}^{2k+1}$ $Y = (y_1, \dots, y_{2k+1}) \in \mathfrak{R}^{2k+1}$ $Z = (z_1, \dots, z_{2k+1}) \in \mathfrak{R}^{2k+1} \text{ y } z_i = f_i(x_i, y_i) + h_i$
$f_i(x_i, y_i):$	la ecuación de un plano en función de las coordenadas x_i e y_i , el cual representa una interfaz.
$V = (0, v_2, \dots, v_{2ni+1}) \in \mathfrak{R}^{2ni+1}:$	Vector de velocidades.
$H = (0, h_2, \dots, h_{ni+1}) \in \mathfrak{R}^{ni+1}:$	Vector de desplazamientos verticales de las interfaces (planos).
$\ell = (0, \ell_2, \dots, \ell_{2ni+1}) \in \mathfrak{R}^{2ni+1}:$	Vector de longitudes de los rayos.

A fin de simplificar el problema es necesario hacer algunas suposiciones sobre las características del subsuelo. Considerando que tenemos un modelo del subsuelo con ni capas, en el cual cada capa está delimitada por dos interfaces, una superior y una inferior, siendo la superficie la primera interfaz, se tienen $ni + 1$ interfaces en total. Entonces suponemos que el medio entre cada par de interfaces sucesivas es homogéneo, es decir, las propiedades físicas no cambian con la posición, y es isótropo, esto es, la velocidad con la que la onda atraviesa el medio no varía con la dirección.

Es importante señalar que los valores de V y H no permanecen constantes, sino, por el contrario cambian, siempre permaneciendo dentro de un intervalo, el cual es obtenido a través de información previa, proveniente del campo, que es usada para regularizar la solución; esto será descrito en los sucesivos capítulos.

CAPÍTULO 2

GRADIENTE ESPECTRAL PROYECTADO

Para hallar una solución de (1.1) se propone utilizar un método de bajo costo computacional, fácil de implementar y muy efectivo para problemas de gran escala, el gradiente espectral proyectado (GEP), el cual está basado en la dirección del gradiente negativo proyectado y la longitud del paso espectral [1].

2.1. Método del Gradiente Espectral Proyectado

El método del gradiente espectral proyectado es un algoritmo para resolver el siguiente problema de optimización:

$$\min_{x \in \Omega} F(x) \quad (2.1)$$
$$F : \Re^n \rightarrow \Re$$

donde Ω es un conjunto cerrado convexo en \Re^n , y F es la función objetivo, la cual es continua y diferenciable en un conjunto abierto que contiene a Ω .

El método de GEP, aplicado al problema (1.1), consiste en generar una sucesión de iterados de la forma:

$$x_{k+1} = x_k + \lambda_k (P_{\Omega}(x_k - \alpha_K \nabla F(x_k)) - x_k) \quad (2.2)$$

donde:

$g(x) = \nabla F(x) :$	es el gradiente de la función F evaluado en x .
$P_{\Omega} :$	es la proyección ortogonal en Ω .
$P_{\Omega}(x_k - \alpha_K \nabla F(x_k)) - x_k :$	dirección del gradiente espectral proyectado.
$\alpha_k :$	el cociente inverso de Rayleigh $\frac{\langle s_{k-1}, s_{k-1} \rangle}{\langle s_{k-1}, y_{k-1} \rangle} = \frac{(s_{k-1})^t s_{k-1}}{(s_{k-1})^t y_{k-1}}$ [1]. con $s_k = x_{k-1} - x_k$ $y_k = g(x_{k-1}) - g(x_k)$ con $g(x) = \nabla F(x)$.
$\lambda_k :$	longitud del paso espectral.

Defínase

$$d_k = P_{\Omega}(x_k - \alpha_K \nabla F(x_k)) - x_k$$

entonces se tendrá una expresión simplificada de (2.2)

$$x_{k+1} = x_k + \lambda_k d_k \quad (2.3)$$

donde d_k es la dirección del gradiente espectral proyectado y λ_k es la longitud del paso espectral obtenido mediante la búsqueda lineal no monótona de Grippo, Lampariello y Lucidi [9].

La búsqueda lineal consiste en realizar un recortamiento en la longitud de paso propuesta:

$$\lambda = \frac{\lambda_{k-1} \|d_{k-1}\|^2}{d_{k-1}^t (d_k - d_{k-1})}$$

hasta que satisfaga la siguiente relación:

$$F(x_{k+1}) \leq \max_{0 \leq j \leq \min(k, M-1)} F(x_{k-j}) + \gamma \lambda (g(x_k))^t d_k \quad (2.4)$$

donde M es un natural y $\gamma \in (0,1)$.

Es importante resaltar que la condición (2.4) obliga a que exista un decrecimiento en la función objetivo (2.1) al cabo de algunas iteraciones, lo que garantiza la convergencia global del método.

2.2. Algoritmo del Gradiente Espectral Proyectado

A continuación se presenta el algoritmo GEP [1]:

El algoritmo inicia con x_o como una proyección ortogonal en Ω , se denota $g(x) = \nabla F(x)$ y se usa un entero $M \geq 1$, un parámetro pequeño $\alpha_{\min} > 0$, un parámetro grande $\alpha_{\max} > \alpha_{\min}$, un parámetro de ajuste $\gamma \in (0,1)$ y $0 < \sigma_1 < \sigma_2 < 1$ parámetros de seguridad. Inicialmente, $\alpha_o \in [\alpha_{\min}, \alpha_{\max}]$ es arbitrario.

Dado $x_k \in \Omega$ y $\alpha_k \in [\alpha_{\min}, \alpha_{\max}]$, el siguiente algoritmo describe como obtener x_{k+1} y α_{k+1} :

Paso 1. Si se satisface el criterio de parada, parar.

Si $\|P_{\Omega}(x_k - g(x_k)) - x_k\| = 0$, parar y declarar que x_k es estacionario.

Paso 2. Búsqueda o *backtracking*

Paso 2.1. Calcular $d_k = P_{\Omega}(x_k - \alpha_k g(x_k)) - x_k$

Hacer $\lambda \leftarrow 1$

Paso 2.2. Hacer $x_+ = x_k + \lambda d_k$

Paso 2.3. Se ajusta el valor de λ y de x_+ hasta que

$$F(x_+) \leq \max_{0 \leq j \leq \min(k, M-1)} F(x_{k-j}) + \gamma \lambda (g(x_k))^t d_k$$

El ajuste es

$$\lambda = \frac{\lambda_{k-1} \|d_{k-1}\|^2}{d_{k-1}^t (d_k - d_{k-1})},$$

$$x_+ = x_k + \lambda d_k$$

Paso 2.4. Se definen

$$x_{k+1} = x_+$$

$$s_k = x_{k-1} - x_k,$$

$$y_k = g(x_{k-1}) - g(x_k)$$

Paso 3.

Calcule $b_k = (s_k)^t y_k$

Si $b_k \leq 0$, hacer $\alpha_{k+1} = \alpha_{\max}$, si no calcule $\alpha_k = (s_k)^t y_k$ y

$$\alpha_{k+1} = \min \left\{ \alpha_{\max}, \max \left\{ \alpha_{\min}, \frac{a_k}{b_k} \right\} \right\}$$

La prueba de la convergencia del método GEP se demuestra en [1] asumiendo ciertas condiciones. El criterio de parada mencionado en el paso 1 del algoritmo anterior, se refiere fundamentalmente a la norma de la proyección del gradiente sobre Ω , la cual debe ser pequeña según una tolerancia establecida. Asimismo, el algoritmo verifica que no se ejecute una cantidad muy grande de evaluaciones de la función, del gradiente y de iteraciones en general.

CAPÍTULO 3

TRAZADOR DE RAYOS

3-D

Ésta es una de las técnicas más aplicables en los problemas de exploración petrolera. Consiste en tomar un modelo inicial de velocidades y de la estructura del suelo, para encontrar la trayectoria y los tiempos de viaje de los rayos que parten de ciertos puntos llamados fuentes hacia otros llamados receptores, de tal manera que dichas trayectorias tenga mínimo tiempo de viaje.

3.1 La función de tiempos de viaje

En resumen, el trazado de rayos es un problema directo que consiste en determinar el tiempo de viaje de una onda que atraviesa un número de interfaces [3]. Así, el tiempo de viaje discretizado T_j a lo largo del rayo j , está dado por:

$$T_j = \sum_{i=2}^{2ni+1} \frac{\ell_i}{v_i}$$

donde el valor de ℓ_i viene dado por la fórmula:

$$\ell_i = \sqrt{[(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2]} \quad \text{y} \quad z_i = f_i(x_i, y_i) + h_i$$

con $f_i(x_i, y_i)$ la ecuación del plano de la interfaz y (x_i, y_i) las coordenadas del rayo j .

El teorema de Fermat afirma que una onda que viaja de un punto a otro sigue la trayectoria de menor tiempo de viaje [6]. Entonces el problema queda como se indic6 anteriormente en (1.2)

$$T(X, Y, Z) = \min_{(x,y,z)} \sum_{i=2}^{2ni+1} \frac{\sqrt{[(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2]}}{v_i}$$

el cual tambi6n depender6 del vector de velocidades de propagaci6n de la onda V y el vector de desplazamientos verticales de las diferentes interfaces del subsuelo H .

El n6mero de rayos depende del n6mero de receptores, fuentes y capas del medio. Se considera que el medio es homog6neo y por lo tanto, el segmento de rayo en la capa i , comprendido entre dos interfaces consecutivas $i-1$ y i , es una recta.

3.2 Descripci6n de las interfaces

Siendo la interface f_i un plano, la ecuaci6n que lo describe ser6 determinada por tres puntos no colineales situados sobre dicho plano [8]. Sean $A, B, C \in \Re^3$ dichos puntos, donde $A = (a_1, b_1, c_1)$, $B = (a_2, b_2, c_2)$, $C = (a_3, b_3, c_3)$, y siendo la ecuaci6n del plano

$$\alpha_1(x - a_1) + \alpha_2(y - b_1) + \alpha_3(z - c_1) = 0$$

la interfaz f_i que pasa por dichos puntos viene dada por:

$$f_i(x, y) = \frac{-\alpha_1(x - a_1) - \alpha_2(y - b_1)}{\alpha_3} + c_1 \quad (3.1)$$

con

$$\alpha_1 = (b_2 - b_1)(c_3 - c_1) - (c_2 - c_1)(b_3 - b_1) \quad (3.2)$$

$$\alpha_2 = (c_3 - c_1)(a_3 - a_1) - (a_2 - a_1)(c_3 - c_1)$$

$$\alpha_3 = (a_2 - a_1)(b_3 - b_1) - (a_3 - a_1)(b_2 - b_1)$$

CAPÍTULO 4

METODOLOGÍA

Resolver el problema de optimización (1.1) utilizando datos provenientes de un trazador de rayos, es el objetivo principal de este trabajo.

La resolución del problema supone un modelo inicial de la estructura del subsuelo: número de capas, receptores y fuentes, velocidades en cada capa, posición de cada uno de los receptores y fuentes, y las coordenadas de cada una de las $ni + 1$ interfaces anteriormente señaladas.

En base a experimentos realizados en campo, se conoce a priori el tiempo T_R de recorrido de cada rayo, pero dado que no se posee, se usará data sintética.

La función objetivo que se quiere minimizar por medio del método del gradiente espectral presentado en (2.1), es la siguiente:

$$F(V, H) = \frac{1}{2} \|T_R - T(V, H)\|_2^2 \quad (4.1)$$

donde $T_R \in \mathfrak{R}^n$ es un vector conocido a través de datos de exploración sísmica, y contiene los tiempos de viaje de los rayos que van desde cada fuente a cada receptor en la superficie y $T(V_k, H_k)$ son los tiempos de viaje estimados y perturbados bruscamente o débilmente vía modelo inicial de la estructura del subsuelo dada.

Dicho método generará una sucesión de iterados como en (2.3) de la forma:

$$x_{k+1} = x_k + \lambda_k [P_\Omega(x_k - \alpha_k \nabla F(x_k)) - x_k]$$
$$x_k = \{V_k, H_k\}$$

donde λ_k es la longitud del paso espectral y $P_\Omega(x_k - \alpha_k \nabla F(x_k)) - x_k$ es la dirección del gradiente espectral proyectado [1], y en este problema en particular

$$\nabla F(V_k, H_k) = -J_T(V_k, H_k)^T (T_R - T(V_k, H_k)) \quad (4.2)$$

Observe que (V_k, H_k) no permanecen fijos, por el contrario cambian constantemente, permaneciendo dentro de un intervalo el cual es definido a través de información proveniente del campo, que permite reducir el espacio solución.

El intervalo antes mencionado estará definido por cotas superiores e inferiores correspondientes a la velocidad V y a la profundidad H , las cuales serán definidas en base a porcentajes. En vista de que no se posee información del campo, dichas cotas están basadas en pruebas hechas con datos sintéticos (Ver Apéndice A).

Nótese que la evaluación del gradiente (4.2), $g(V_k, H_k) = \nabla F(V_k, H_k)$ en el modelo de (V_k, H_k) requiere conocer los tiempos de viaje estimados por medio del trazador de rayos $T(V_k, H_k) = (\tau_1, \dots, \tau_n)^t$, los valores $T_R = (t_1, \dots, t_n)^t$ de los tiempos de viaje reales (debido al alcance de este estudio, datos sintéticos) y la matriz jacobiana evaluada en (V_k, H_k) , denotada por $J_T \in \Re^{n \times 3ni}$ (vea figura 4.1).

Así, para $n = ni \cdot ns \cdot nr$ rayos, la forma general de la matriz jacobiana para los tiempos de viaje $T(V_k, H_k)$, es la siguiente:

$$J_T = \left(\begin{array}{ccc|ccc} \frac{\partial T_1}{\partial v_2} & \dots & \frac{\partial T_1}{\partial v_{2ni+1}} & \frac{\partial T_1}{\partial h_2} & \dots & \frac{\partial T_1}{\partial h_{ni+1}} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial T_n}{\partial v_2} & \dots & \frac{\partial T_n}{\partial v_{2ni+1}} & \frac{\partial T_n}{\partial h_2} & \dots & \frac{\partial T_n}{\partial h_{ni+1}} \\ \hline \frac{\partial T_1}{\partial v_2} & \dots & \frac{\partial T_1}{\partial v_{2ni+1}} & \frac{\partial T_1}{\partial h_2} & \dots & \frac{\partial T_1}{\partial h_{ni+1}} \\ \vdots & & \vdots & \vdots & & \vdots \\ \frac{\partial T_n}{\partial v_2} & \dots & \frac{\partial T_n}{\partial v_{2ni+1}} & \frac{\partial T_n}{\partial h_2} & \dots & \frac{\partial T_n}{\partial h_{ni+1}} \end{array} \right)_{(n \times 3ni)}$$

4.1 Matriz Jacobiana

Para la evaluación del gradiente (4.2), es necesario calcular la Matriz Jacobiana, es por ello que se crea el módulo `fun_jac.f90`.

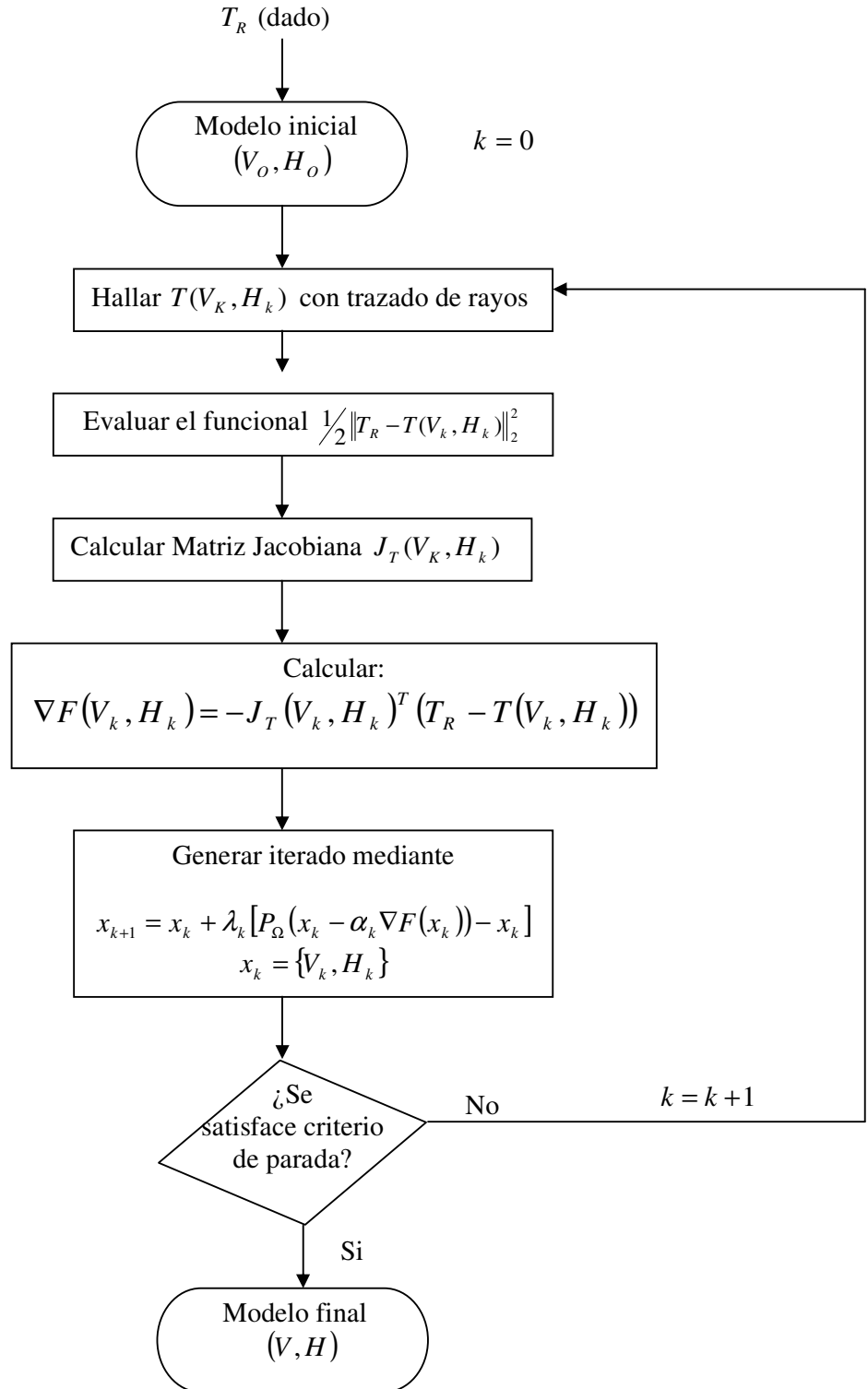


Fig. 4.1: Modelaje sísmico completo.

Este módulo efectúa el cálculo de la matriz jacobiana de los tiempos de viaje de los rayos que viajan de un punto fuente x_s a un punto receptor x_r , reflejándose en las diferentes capas del subsuelo, como se aprecia en la figura 4.2.

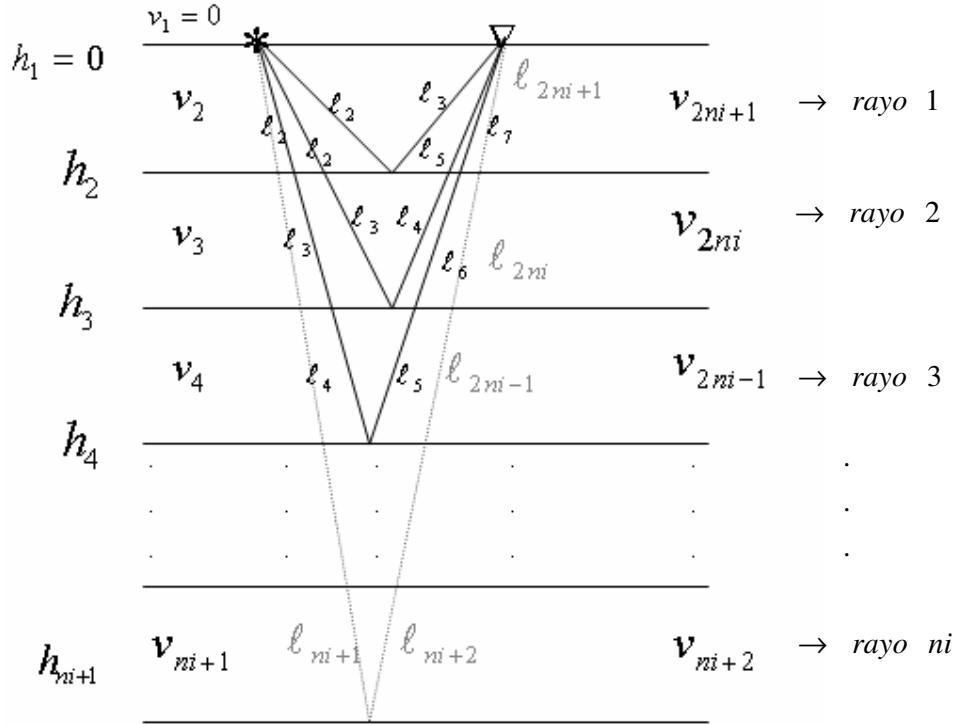


Fig. 4.2: Modelo de ni capas, 1 receptor y 1 fuente

* Fuente
▽ Receptor

Observe la figura 4.2 y note que se trata de un modelo de ni capas que posee sólo una fuente y un receptor, este es el modelo más sencillo posible, puesto que cada capa contendrá únicamente un rayo, lo que da un total de ni rayos.

Como ya se mencionó, el número total de rayos depende del número de pares (x_s, x_r) y el número de capas ni , en el medio. Se considera que el medio es homogéneo y por lo tanto, el segmento de rayo en la capa i , comprendida entre dos interfaces consecutivas $i-1$ e i , es una recta.

Así, para $n = ni \cdot ns \cdot nr$ rayos, la forma general de la matriz jacobiana de los tiempos de viaje es la siguiente:

$$J_T = \left(\begin{array}{ccc|ccc} \frac{\partial T_1}{\partial v_2} & \dots & \frac{\partial T_1}{\partial v_{2ni+1}} & \frac{\partial T_1}{\partial h_2} & \dots & \frac{\partial T_1}{\partial h_{ni+1}} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial T_n}{\partial v_2} & \dots & \frac{\partial T_n}{\partial v_{2ni+1}} & \frac{\partial T_n}{\partial h_2} & \dots & \frac{\partial T_n}{\partial h_{ni+1}} \end{array} \right) \begin{array}{l} \longrightarrow \text{rayo } 1 \\ \vdots \\ \longrightarrow \text{rayo } n \end{array}$$

(n \times 3 \cdot ni)

Note que cada fila j corresponde a la derivada de los tiempos de viaje para el rayo j con respecto a V y H , donde:

$$J_T \in \mathfrak{R}^{n \times 3ni}$$

$n = ni \cdot ns \cdot nr$: número total de rayos

ns : número de fuentes

nr : número de receptores

ni : número de capas

v : vector de velocidades, $v = (0, v_2, \dots, v_{2ni+1}) \in \mathfrak{R}^{2ni+1}$

h : vector de traslación (en dirección vertical), $h = (0, h_2, \dots, h_{ni+1}) \in \mathfrak{R}^{ni+1}$

ℓ : vector longitud de los rayos, $\ell = (0, \ell_2, \dots, \ell_{2ni+1}) \in \mathfrak{R}^{2ni+1}$

T_j : tiempo de viaje para el rayo j

Es ya sabido del capítulo 3 que el tiempo de viaje T_j a lo largo del rayo j , está dado por

$$T_j = \sum_{i=2}^{2ni+1} \frac{\ell_i}{v_i}$$

donde el valor de ℓ_i viene dado por la fórmula:

$$\ell_i = \sqrt{[(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2]} \quad (4.3)$$

y

$z_i = f(x_i, y_i) + h_i$ con $i = 1, \dots, k$ donde $k = ni + 1$, para los rayos que bajan

$z_i = f(x_i, y_i) + h_{2k+1-i}$ con $i = k, \dots, 2ni$, para los rayos que suben

con $f(x_i, y_i)$ la ecuación del plano de la interfaz vista en 3.1.

Así, los tiempos de viaje para cada uno de los rayos de la figura 4.2, correspondiente al modelo de ni capas, con un par fuente – receptor (caso sencillo), son los siguientes:

$$\begin{aligned}
T_1 &= \frac{\ell_2}{v_2} + \frac{\ell_3}{v_{2ni+1}} \\
T_2 &= \frac{\ell_2}{v_2} + \frac{\ell_3}{v_3} + \frac{\ell_4}{v_{2ni}} + \frac{\ell_5}{v_{2ni+1}} \\
T_3 &= \frac{\ell_2}{v_2} + \frac{\ell_3}{v_3} + \frac{\ell_4}{v_4} + \frac{\ell_5}{v_{2ni-1}} + \frac{\ell_6}{v_{2ni}} + \frac{\ell_7}{v_{2ni+1}} \\
&\vdots \\
T_{ni} &= \frac{\ell_2}{v_2} + \frac{\ell_3}{v_3} + \frac{\ell_4}{v_4} + \frac{\ell_5}{v_5} + \frac{\ell_6}{v_6} + \frac{\ell_7}{v_7} + \dots + \frac{\ell_{2ni+1}}{v_{2ni+1}}
\end{aligned}$$

Es importante señalar que cada rayo tiene valores de ℓ_i distintos entre sí, debido a que cada rayo tiene distintas coordenadas en el plano.

Finalmente la matriz jacobiana general para el modelo de ni capas, con un par fuente – receptor (caso sencillo), es la siguiente:

$$J_T = \langle A | B \rangle$$

donde:

$$A = \begin{bmatrix}
-\frac{\ell_2}{v_2^2} & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & -\frac{\ell_3}{v_{2ni+1}^2} \\
-\frac{\ell_2}{v_2^2} & -\frac{\ell_3}{v_3^2} & 0 & 0 & 0 & 0 & \dots & 0 & -\frac{\ell_4}{v_{2ni}^2} & -\frac{\ell_5}{v_{2ni+1}^2} \\
-\frac{\ell_2}{v_2^2} & -\frac{\ell_3}{v_3^2} & -\frac{\ell_4}{v_4^2} & 0 & 0 & 0 & \dots & -\frac{\ell_5}{v_{2ni-1}^2} & -\frac{\ell_6}{v_{2ni}^2} & -\frac{\ell_7}{v_{2ni+1}^2} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
-\frac{\ell_2}{v_2^2} & -\frac{\ell_3}{v_3^2} & -\frac{\ell_4}{v_4^2} & -\frac{\ell_5}{v_5^2} & -\frac{\ell_6}{v_6^2} & -\frac{\ell_7}{v_7^2} & \dots & -\frac{\ell_{2ni-1}}{v_{2ni-1}^2} & -\frac{\ell_{2ni}}{v_{2ni}^2} & -\frac{\ell_{2ni+1}}{v_{2ni+1}^2}
\end{bmatrix}$$

$$B = \begin{bmatrix} \frac{z_2 - z_1}{\ell_2 \gamma_2} - \frac{z_3 - z_2}{\ell_3 \gamma_{2n+1}} & 0 & \dots & 0 \\ \frac{z_2 - z_1}{\ell_2 \gamma_2} - \frac{z_3 - z_2}{\ell_3 \gamma_{2n+1}} + \frac{z_4 - z_3}{\ell_4 \gamma_{2ni}} & \frac{z_3 - z_2}{\ell_3 \gamma_3} - \frac{z_4 - z_3}{\ell_4 \gamma_{2ni}} & \dots & 0 \\ \frac{z_2 - z_1}{\ell_2 \gamma_2} - \frac{z_3 - z_2}{\ell_3 \gamma_3} + \frac{z_6 - z_5}{\ell_6 \gamma_{2ni}} & \frac{z_3 - z_2}{\ell_3 \gamma_3} - \frac{z_4 - z_3}{\ell_4 \gamma_{2ni}} + \frac{z_6 - z_5}{\ell_6 \gamma_{2ni}} & \dots & 0 \\ \frac{z_2 - z_1}{\ell_2 \gamma_2} - \frac{z_3 - z_2}{\ell_3 \gamma_3} + \frac{z_6 - z_5}{\ell_6 \gamma_{2ni}} + \frac{z_7 - z_6}{\ell_7 \gamma_{2n+1}} & \frac{z_3 - z_2}{\ell_3 \gamma_3} - \frac{z_4 - z_3}{\ell_4 \gamma_{2ni}} + \frac{z_6 - z_5}{\ell_6 \gamma_{2ni}} + \frac{z_7 - z_6}{\ell_7 \gamma_{2n+1}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \frac{z_2 - z_1}{\ell_2 \gamma_2} - \frac{z_3 - z_2}{\ell_3 \gamma_3} + \frac{z_{2ni} - z_{2ni-1}}{\ell_{2ni} \gamma_{2ni}} & \frac{z_3 - z_2}{\ell_3 \gamma_3} - \frac{z_4 - z_3}{\ell_4 \gamma_{2ni}} + \frac{z_{2ni} - z_{2ni-1}}{\ell_{2ni} \gamma_{2ni}} & \dots & \frac{z_{2ni-2} - z_{2ni-3}}{\ell_{2ni-2} \gamma_{2ni-2}} - \frac{z_{2ni-1} - z_{2ni-2}}{\ell_{2ni-1} \gamma_{2ni-1}} \\ \frac{z_2 - z_1}{\ell_2 \gamma_2} - \frac{z_3 - z_2}{\ell_3 \gamma_3} + \frac{z_{2ni} - z_{2ni-1}}{\ell_{2ni} \gamma_{2ni}} & \frac{z_3 - z_2}{\ell_3 \gamma_3} - \frac{z_4 - z_3}{\ell_4 \gamma_{2ni}} + \frac{z_{2ni} - z_{2ni-1}}{\ell_{2ni} \gamma_{2ni}} & \dots & \frac{z_{2ni-2} - z_{2ni-3}}{\ell_{2ni-2} \gamma_{2ni-2}} - \frac{z_{2ni-1} - z_{2ni-2}}{\ell_{2ni-1} \gamma_{2ni-1}} \end{bmatrix}$$

El caso más complicado sucede cuando tenemos más de una fuente y más de un receptor; para este caso los tiempos de viaje para cada uno de los rayos son los siguientes:

Primera capa (ver figura 4.4):

$$T_1 = \frac{\ell_2}{v_2} + \frac{\ell_3}{v_{2ni+1}}$$

$$T_2 = \frac{\ell_2}{v_2} + \frac{\ell_3}{v_{2ni+1}}$$

\vdots

$$T_{nsnr} = \frac{\ell_2}{v_2} + \frac{\ell_3}{v_{2ni+1}}$$

Segunda capa:

$$T_{nsnr+1} = \frac{\ell_2}{v_2} + \frac{\ell_3}{v_3} + \frac{\ell_4}{v_{2ni}} + \frac{\ell_5}{v_{2ni+1}}$$

\vdots

$$T_{2nsnr} = \frac{\ell_2}{v_2} + \frac{\ell_3}{v_3} + \frac{\ell_4}{v_{2ni}} + \frac{\ell_5}{v_{2ni+1}}$$

Tercera capa:

$$T_{2nsnr+1} = \frac{\ell_2}{v_2} + \frac{\ell_3}{v_3} + \frac{\ell_4}{v_4} + \frac{\ell_5}{v_{2ni-1}} + \frac{\ell_6}{v_{2ni}} + \frac{\ell_7}{v_{2ni+1}}$$

\vdots

$$T_{3nsnr} = \frac{\ell_2}{v_2} + \frac{\ell_3}{v_3} + \frac{\ell_4}{v_4} + \frac{\ell_5}{v_{2ni-1}} + \frac{\ell_6}{v_{2ni}} + \frac{\ell_7}{v_{2ni+1}}$$

\vdots

ni-ésima capa (ver figura 4.4):

$$T_{(ni-1)nsnr+1} = \frac{\ell_2}{v_2} + \frac{\ell_3}{v_3} + \frac{\ell_4}{v_4} + \frac{\ell_5}{v_5} + \frac{\ell_6}{v_6} + \frac{\ell_7}{v_7} + \dots + \frac{\ell_{2ni-1}}{v_{2ni+1}}$$

\vdots

$$T_{ninsnr} = \frac{\ell_2}{v_2} + \frac{\ell_3}{v_3} + \frac{\ell_4}{v_4} + \frac{\ell_5}{v_5} + \frac{\ell_6}{v_6} + \frac{\ell_7}{v_7} + \dots + \frac{\ell_{2ni+1}}{v_{2ni+1}}$$

Igual que para el caso sencillo ya descrito, cada rayo tiene valores de ℓ_i distintos entre sí, debido a que cada rayo tiene distintas coordenadas en el plano y distintas combinaciones fuente – receptor, como se observa en las figuras 4.3 y 4.4.

Observe que para la figura 4.3, los únicos tiempos de viaje a calcular corresponden a los de la primera capa ya mostrados, con $v_{2ni+1} = v_3$

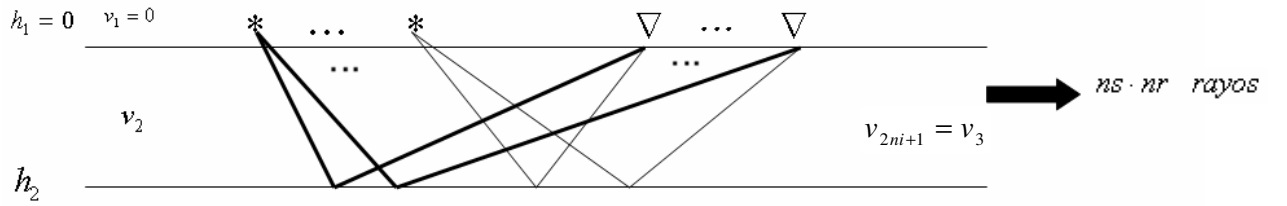


Fig. 4.3: Modelo de 1 capa, ns receptores y nr fuentes

* Fuente
 ∇ Receptor

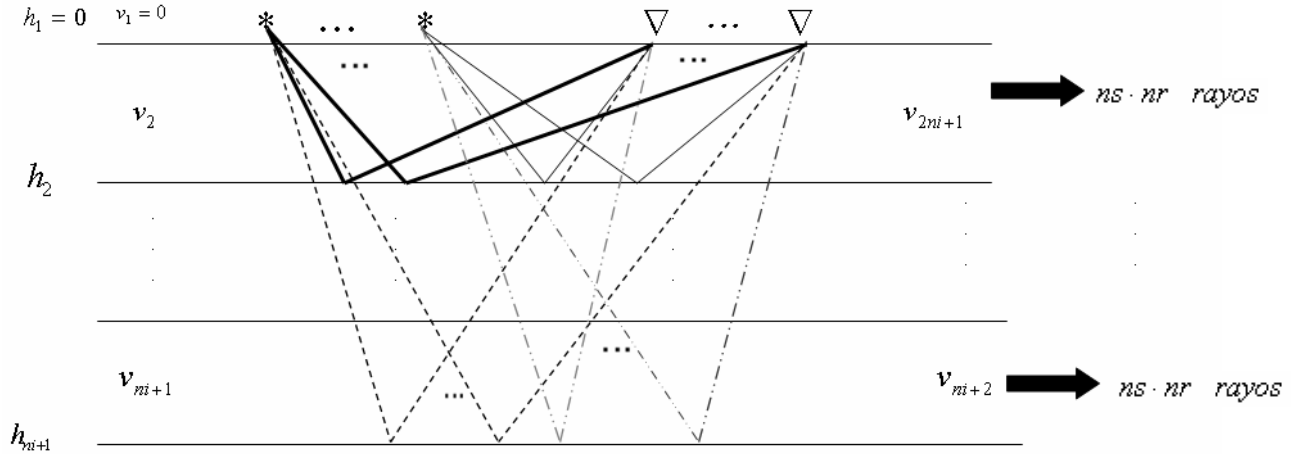


Fig. 4.4: Modelo de ni capas, ns receptores y nr fuentes

* Fuente
 ∇ Receptor

4.2 Descripción de las interfaces

En la sección 3.2 se hizo una breve descripción de las interfaces f_i . Para definir las se requiere de tres puntos no colineales $A = (a_1, b_1, c_1)$, $B = (a_2, b_2, c_2)$, $C = (a_3, b_3, c_3) \in \mathfrak{R}^3$ y de f_i definida como en (3.1).

Sea h_i el valor de traslación vertical para la capa i , entonces los tres puntos no colineales dados, quedan determinados de la siguiente forma:

$$A = (a_1, b_1, c_1 + h_i), \quad B = (a_2, b_2, c_2 + h_i), \quad C = (a_3, b_3, c_3 + h_i)$$

donde la interfaz f_i que pasa por dichos puntos viene dada por:

$$f_i(x, y) = \frac{-\alpha_1(x - a_1) - \alpha_2(y - b_1)}{\alpha_3} + c_1 + h_i$$

con α_1 , α_2 y α_3 definidas como en (3.2).

4.3 Pasos a seguir

A partir de la descripción anterior, los pasos a seguir para resolver el problema, sobre una determinada región del subsuelo, son los siguientes:

1. Colocar en el área de estudio ns fuentes y nr receptores, y obtener un vector de tiempos de viaje reales $T_R = (t_1, \dots, t_n)^t$.
2. Establecer un modelo inicial de la estructura de las capas del subsuelo y valores para las profundidades y velocidades en cada capa.
3. Utilizar el procedimiento de trazado de rayos para determinar los tiempos de viaje perturbados, $T(V_k, H_k)$

4. Establecer cotas sobre las variables para reducir el espacio solución (determinación del conjunto P_Ω)
5. Resolver el problema de optimización (1.1), mediante el método del gradiente espectral:
 - i. Evaluar el funcional (4.1)
 - ii. Calcular la matriz jacobiana $J_T \in \Re^{nx3ni}$
 - iii. Evaluar el gradiente (4.2)
 - iv. Proyectar sobre el conjunto P_Ω
6. Obtener el modelo final de profundidades y velocidades.

CAPÍTULO 5

IMPLANTACIÓN

En este capítulo se presenta el esquema general del algoritmo final (ver figura 4.1) utilizado para la resolución del problema planteado.

El esquema utilizado para adaptar los algoritmos al problema, contiene básicamente 7 pasos, que se especifican a continuación:

1. Obtener datos del modelo geológico del área de estudio:
 - a. Número de capas, ni .
 - b. Número de fuentes, ns .
 - c. Número de receptores, nr .
 - d. Velocidades y profundidades iniciales, (V_o, H_o) .
 - e. Posición de las fuentes, x_s .
 - f. Posición de los receptores, x_r .
 - g. Coordenadas de los puntos que definen las interfaces.
 - h. Tiempos de viaje reales T_R .
2. Determinar el iterado inicial.
 - a. Obtener iterado inicial de velocidades y traslaciones verticales (perturbados con respecto a los datos reales o para este estudio datos sintéticos).

3. Calcular los tiempos de viaje $T(V, H)$
4. Determinar las cotas para los iterados (V_k, H_k) , el conjunto P_Ω estará definido en base a las cotas inferiores y superiores correspondientes a las velocidades V y V_o , y a la profundidad H y H_o , este último siempre será cero.
5. Aplicar el método de gradiente espectral proyectado.
 - a. Evaluación del funcional $\frac{1}{2} \|T_R - T(V_k, H_k)\|_2^2$
 - b. Cálculo de la matriz jacobiana $J_T(V_k, H_k) \in \Re^{nx3ni}$
 - c. Evaluación del gradiente $-J_T(V_k, H_k)^T (T_R - T(V_k, H_k))$
 - d. Proyección del iterado en el conjunto P_Ω .
6. Generar velocidades y profundidades finales en cada capa.

Para esto se adaptó el código trazador de rayos [6] (utilizado en los pasos 1 y 3), el cual fue escrito en FORTRAN 77 y el código Gradiente Espectral Proyectado (GEP) [2], escrito en FORTRAN 90. Las modificaciones realizadas se explican brevemente en el apéndice A.

La subrutina GEP (utilizada en el paso 5) implementa el Método del Gradiente Espectral Proyectado para encontrar los mínimos locales de una función dada con restricciones convexas descritas en [1] mediante el algoritmo [2].

El usuario debe incluir subrutinas externas EVALF, para la evaluación de la función (4.1), EVALG para la evaluación del gradiente (4.2), y PROJ para proyectar un punto arbitrario en una región factible determinada en el paso 4. La forma de estas subrutinas (EVALF, EVALG y PROJ) serán descritas en el apéndice A.

La máquina usada para esta implantación es una PC *Intel Pentium IV, 1.7 GHz*, 256 Mb. de memoria RAM y sistema operativo Windows XP Profesional.

CAPÍTULO 6

RESULTADOS NUMÉRICOS

A continuación se presentan resultados numéricos provenientes de modelos sintéticos, los cuáles son expuestos a través de tablas y gráficas.

Básicamente se realizaron pruebas con varios modelos, para los cuales se analizó la relación entre el tamaño del problema y los parámetros: número de iteraciones, número de evaluaciones de la función y número de evaluaciones del gradiente. En todas las pruebas realizadas se consideran capas planas.

Se hicieron numerosas pruebas variando los parámetros antes mencionados, y a continuación se presenta las que permiten una mejor verificación de la correcta metodología e implantación.

En algunos casos se detuvo el programa porque se alcanzó un número máximo de iteraciones o evaluaciones de la función.

Los parámetros para el GEP se muestran en la tabla 6.1:

M:	21
Número máximo de iteraciones:	200.000,00
Número máximo de evaluaciones del funcional y el gradiente:	200.000,00
Criterio de parada (eps):	1×10^{-11}

Tabla 6.1: Parámetros GEP

Es importante señalar que cada componente del vector de velocidades difiere del punto inicial en un 30% de la componente correspondiente a la data real, dado que los valores correspondientes a los desplazamientos verticales (H) son inicialmente cero en todos los casos, los valores perturbados de H serán el 30% de las coordenadas z de las interfaces.

Se consideraron cuatro modelos, en los tres primeros se utilizan interfaces planas horizontales con variaciones en el número de capas, fuentes, y receptores junto con variaciones en los valores de las velocidades y desplazamientos verticales. El cuarto modelo considera interfaces planas tanto horizontales como buzantes (diferentes valores de la coordenada z para una misma interfaz).

6.1 Modelo 1

El siguiente modelo es bastante sencillo, consta de 3 fuentes, 8 receptores y 3 capas, lo que produce 72 rayos. La distribución de las fuentes y los receptores se estableció de manera aleatoria. En la tabla 6.2 se expone el modelo real de velocidades e interfaces, en donde cada columna de la izquierda representa un vector de velocidades y cada columna de la derecha un vector correspondiente a la coordenada z de cada interfaz.

Velocidades (V)						Interfaz (coordenada z)		
1		2		3		1	2	3
Bajada	Subida	Bajada	Subida	Bajada	Subida			
2.0000	2.0000	2.30000	2.30000	12.000	12.000	1.0700	0.9870	0.6200
2.0000	2.0000	3.10000	3.10000	7.0000	7.0000	2.4900	1.1200	1.0700
2.0000	2.0000	4.60000	4.60000	4.0000	4.00000	3.5900	1.6700	1.3900

Tabla 6.2: Velocidades e interfaces reales para el modelo 1

Velocidades (V)						Desplazamientos verticales (H)		
1		2		3		1	2	3
Bajada	Subida	Bajada	Subida	Bajada	Subida			
2.6000	2.6000	2.9900	2.9900	15.600	15.600	0.3210	0.2961	0.1860
2.6000	2.6000	4.0300	4.0300	9.1000	9.1000	0.7470	0.3360	0.3210
2.6000	2.6000	5.9800	5.9800	5.2000	5.2000	1.0770	0.5010	0.4170

Tabla 6.3: Punto inicial de velocidades y desplazamientos verticales para el modelo 1

En primer lugar se consideró el problema de estimar sólo velocidades partiendo de los puntos iniciales dados en la tabla 6.3, las soluciones de esta primera estimación se pueden observar en la tabla 6.4.

Velocidades (V)					
1		2		3	
Bajada	Subida	Bajada	Subida	Bajada	Subida
2.00000	2.00000	2.29999	2.30000	12.00000	11.99999
1.99999	2.00000	3.10000	3.09999	7.000000	6.999999
2.00000	1.99999	4.59999	4.60000	4.00000	4.000000

Tabla 6.4: Punto solución de velocidades para el modelo 1

Para el primer vector de velocidades se requirió de 28 iteraciones, 30 evaluaciones del funcional y 29 evaluaciones de gradiente, obteniéndose la solución en 20 centésimas de seg. tiempo de CPU. El segundo vector de velocidades se obtuvo en 39 centésimas de seg., y tomó 54 iteraciones, 56 evaluaciones del funcional y 55 evaluaciones del gradiente. El tercer y último vector de velocidades se obtuvo en 31 iteraciones, 34 evaluaciones del funcional y 32 evaluaciones del gradiente en un tiempo de 39 centésimas de seg.

Después se consideró el problema de estimar interfaces tomando las velocidades reales. Se tomaron los puntos iniciales de la tabla 6.3, correspondientes a los desplazamientos verticales iniciales.

Para hacer esto se consideró utilizar el conjunto de valores de desplazamientos verticales 1, 2 y 3 utilizando el vector de velocidades iniciales de la columna 3 de la tabla 6.3, para observar el número de iteraciones, evaluación del gradiente y funcional junto con el tiempo de convergencia.

Para el primer vector de desplazamientos verticales se requirió de 3.306 iteraciones, 4.758 evaluaciones de la función y 3.307 evaluaciones del gradiente en un tiempo de 35 seg. 50 centésimas de seg. El segundo vector de desplazamientos generó 2.985 iteraciones, 4.136 evaluaciones del funcional y 2.986 evaluaciones del gradiente en 43 seg. 87 centésimas de seg. tiempo de CPU. Y por último se tiene el tercer vector de desplazamientos que produjo 6.739 iteraciones, 9.963 evaluaciones del funcional y 6.740 evaluaciones del gradiente en 2 min. 2 seg. 65 centésimas de seg.

Observe que el tercer vector de desplazamientos verticales fue el que requirió de mayor número de iteraciones, evaluaciones y tiempo de convergencia; este vector corresponde a interfaces cuya separación es pequeña.

Para finalizar, se consideró el problema de estimar velocidades e interfaces en conjunto, tomando como punto inicial las perturbaciones hechas en los dos casos anteriores.

En primer lugar se usaron los conjuntos de velocidades 1, 2 y 3, y se compararon entre sí usando el conjunto de desplazamientos verticales 3 de la tabla 6.3, obteniéndose los resultados mostrados en la tabla 6.5.

Velocidades (V)						Desplazamientos verticales (H)
1		2		3		
Bajada	Subida	Bajada	Subida	Bajada	Subida	
1.999999	2.00000	2.30000	2.30000	12.0000	11.99999	0.0000000
2.000000	1.99999	3.10000	3.10000	7.00000	6.999999	0.0000000
2.000000	2.00000	4.60000	4.60000	3.99999	3.999999	0.0000000

Tabla 6.5: Punto solución de velocidades e interfaces para el modelo 1

El primer conjunto de velocidades requirió de 1 seg. 70 centésimas de segundo, y 287 iteraciones, 298 evaluaciones del funcional y 288 evaluaciones del gradiente. El segundo conjunto de velocidades tardó 3 seg. 18 centésimas de segundo, utilizando 416 iteraciones, 444 evaluaciones del funcional y 417 evaluaciones del gradiente. El tercer conjunto requirió de 6.869 iteraciones, 10.792 evaluaciones del funcional, 6.870 evaluaciones del gradiente y 2 min. 1 seg. 20 centésimas de segundo.

Al usar los vectores de desplazamientos verticales 1 y 2 se obtuvieron resultados muy parecidos a los de la tabla 6.5 pero con variaciones en el número de iteraciones, evaluaciones del gradiente, evaluaciones del funcional y en el tiempo tomado para la convergencia. En la tabla 6.6 puede observarse estos datos.

Observe que para la fila correspondiente al primer vector de velocidades, se observa que el menor número de iteraciones ocurre para el tercer vector de desplazamientos verticales. Para la segunda fila correspondiente al segundo vector de velocidades el menor número de iteraciones es obtenido nuevamente por el tercer vector de H y el mayor número de iteraciones por el primer vector de desplazamientos verticales H . Hasta aquí parece haber una relación, en la cual mientras menor sea el H más rápida es la convergencia, pero para la tercera fila correspondiente al vector 3 de velocidades, las cosas cambian radicalmente, el menor número de iteraciones sucede para el primer vector de desplazamientos H y el mayor número de iteraciones para el segundo vector H .

Velocidades		Desplazamientos verticales		
		1	2	3
1	Iteraciones	809	788	287
	Evalf	898	954	298
	Evalg	810	789	288
	Tiempo CPU	4,17 seg.	12,04 seg.	1,70 seg.
2	Iteraciones	1.659	486	416
	Evalf	2.088	509	444
	Evalg	1.660	487	417
	Tiempo CPU	12, 84 seg.	5,81 seg.	3, 18 seg.
3	Iteraciones	3.728	20.092	6.869
	Evalf	5.378	32.105	10.792
	Evalg	3.729	20.093	6.870
	Tiempo CPU	40,26 seg.	5 min. 23 seg.	2 min. 1 seg.

Tabla 6.6: Convergencia de los puntos solución de velocidades e interfaces para el modelo 1

Ahora observe la primera columna, correspondiente al primer vector de desplazamientos H , el menor número de iteraciones ocurre para el primer vector de velocidades y el mayor número de iteraciones para el tercer vector de velocidades, sucede lo mismo para la tercera columna correspondiente al tercer vector de H . Pero para la segunda columna ocurre que el menor número de iteraciones sucede para el segundo vector de velocidades y el mayor número de iteraciones para el tercer vector de velocidades.

Observe que el último problema, en el cual se intenta estimar los parámetros que determinan las velocidades y desplazamientos en conjunto, es evidentemente más

complicado, esto se refleja en el significativo incremento del número de iteraciones y evaluaciones que realiza el algoritmo.

6.2 Modelo 2

Este modelo consta de 9 fuentes, 18 receptores y 4 capas, por lo que existen 648 rayos. La distribución de las fuentes y receptores fue escogida de manera aleatoria. En la tabla 6.7 se presenta el modelo real de velocidades e interfaces, en donde cada columna de la izquierda representa un vector de velocidades y cada columna de la derecha un vector correspondiente a la coordenada z de cada interfaz.

Velocidades (V)						Interfaz (coordenada z)		
1		2		3		1	2	3
Bajada	Subida	Bajada	Subida	Bajada	Subida			
2.0000	2.0000	2.30000	2.30000	12.000	12.000	1.0700	0.9870	0.6200
2.0000	2.0000	3.10000	3.10000	7.000	7.0000	2.4900	1.1200	1.0700
2.0000	2.0000	4.60000	4.60000	4.0000	4.0000	3.5900	1.6700	1.3900
2.0000	2.0000	5.80000	5.80000	2.2500	2.2500	3.9800	2.3200	1.9400

Tabla 6.7: Velocidades e interfaces reales para el modelo 2

Velocidades (V)						Desplazamientos verticales (H)		
1		2		3		1	2	3
Bajada	Subida	Bajada	Subida	Bajada	Subida			
2.6000	2.6000	2.9900	2.9900	15.600	15.600	0.3210	0.2961	0.1860
2.6000	2.6000	4.0300	4.0300	9.1000	9.1000	0.7470	0.3360	0.3210
2.6000	2.6000	5.9800	5.9800	5.2000	5.2000	1.0770	0.5010	0.4170
2.6000	2.6000	7.5400	7.5400	2.9250	2.9250	1.1940	0.6960	0.5820

Tabla 6.8: Punto inicial de velocidades y desplazamientos verticales para el modelo 2

En primer lugar se consideró el problema de estimar sólo velocidades partiendo del punto inicial mostrado en la tabla 6.8, las soluciones se presentan en la tabla 6.9.

Velocidades (V)					
1		2		3	
Bajada	Subida	Bajada	Subida	Bajada	Subida
1.99999	2.00000	2.30000	2.30000	12.00000	11.99999
2.00000	1.99999	3.09999	3.09999	7.000000	6.999999
1.99999	2.00000	4.59999	4.59999	3.999999	4.000000
2.00000	1.99999	5.80000	5.80000	2.249999	2.250000

Tabla 6.9: Punto solución de velocidades para el modelo 2

Para el primer conjunto de velocidades se requirió de 74 iteraciones, 76 evaluaciones del funcional y 75 evaluaciones del gradiente en un tiempo de 8,76 seg. Para el segundo conjunto de velocidades cuyo tiempo de ejecución fue de 12,59 seg., se requirió de 94 iteraciones, 97 evaluaciones del funcional y 95 evaluaciones de gradiente. Finalmente, el último conjunto de velocidades utilizó 109 iteraciones, 113 evaluaciones del funcional y 110 evaluaciones del gradiente en un tiempo de 21,17 seg.

Observe que el mayor número de iteraciones sucede para el tercer vector de velocidades, con poca diferencia le sigue el segundo vector de velocidades y por último con el menor número de iteraciones se tiene el primer vector de velocidades.

Después se consideró el problema de estimar los desplazamientos H de las interfaces solamente. Se tomó el punto inicial de la tabla 6.8 correspondiente a los desplazamientos verticales H .

Para hacer esto se consideró utilizar el conjunto de valores de desplazamientos verticales 1, 2 y 3 utilizando el vector de velocidades iniciales de la columna 3 de la tabla 6.8, para observar el número de iteraciones, evaluación del gradiente y funcional junto con el tiempo de convergencia.

Para el primer vector de H se requirió de 56.704 iteraciones, 94.029 evaluaciones del funcional y 56.705 evaluaciones del gradiente en un tiempo de 3 h. 51

min. y 34 seg. El segundo vector de H tomó 1h. 11 min. 27 seg. en obtener la solución, con un total de 17.737 iteraciones, 27.076 evaluaciones de función y 17.738 evaluaciones del gradiente. El tercer vector de H tardó 1 h. 21 min. 59 seg., usando 20.794 iteraciones, 31.920 evaluaciones del funcional y 20.795 evaluaciones del gradiente.

Al igual que para el primer modelo, el menor número de iteraciones sucedió para el segundo vector de desplazamientos, pero a diferencia del modelo 1, el mayor número de iteraciones no fue para el tercer vector de desplazamientos verticales sino para el primero.

Para finalizar se consideró el problema de estimar velocidades y desplazamientos verticales en conjunto, tomando como punto inicial las perturbaciones hechas en los dos casos anteriores.

En primer lugar se usaron los conjuntos de velocidades 1, 2 y 3 para ser comparados entre sí usando el conjunto de desplazamientos verticales 3 de la tabla 6.8, obteniéndose los resultados mostrados en la tabla 6.10.

Velocidades (V)						Desplazamientos verticales (H)
1		2		3		
Bajada	Subida	Bajada	Subida	Bajada	Subida	
2.000000	1.99999	2.2999999	2.29999	12.0000	11.99999	0.0000000
1.999999	2.00000	3.1000000	3.10000	6.99999	7.000000	0.0000000
1.999999	2.00000	4.6000000	4.60000	4.00000	4.000000	0.0000000
2.000000	1.99999	5.8000000	5.80000	2.24999	2.24999	0.0000000

Tabla 6.10: Punto solución de velocidades e interfaces para el modelo 2

El primer conjunto de velocidades requirió de 577 iteraciones, 619 evaluaciones del funcional y 578 iteraciones para el gradiente en un tiempo de 1 min. 6 seg. El segundo conjunto de velocidades se obtuvo en 3 min. y 52 seg., requiriendo de 1.731 iteraciones, 2.038 evaluaciones del funcional y 1.732 evaluaciones del gradiente. Finalmente el tercer y último conjunto de velocidades precisó de 69.265 iteraciones,

104.704 evaluaciones del funcional y 69.266 evaluaciones del gradiente en un tiempo de 4 h. 33 min. y 8 seg.

Observe primero la tabla 6.11 por filas y note que el mayor número de iteraciones sucede para el primer vector de desplazamientos y que el menor número de iteraciones se produce para el tercer vector de desplazamientos, excepto para la última fila, en la cual el mayor número de iteraciones se produce para el vector 2 de H y el menor para el primer vector de H .

Ahora observe por columna y note que el menor número de iteraciones se produce para el vector de velocidades 1 y el mayor número de iteraciones para el vector de velocidades 3.

De igual forma que para el modelo 1, el problema de estimar velocidades y desplazamientos en conjunto requiere un mayor número de iteraciones, evaluaciones y por supuesto tiempo en comparación con el problema de estimar sólo velocidades o sólo desplazamientos verticales.

Velocidades		Desplazamientos verticales		
		1	2	3
1	Iteraciones	9.778	2.110	577
	Evalf	12.985	2.684	619
	Evalg	9.779	2.111	578
	Tiempo	17 min. 48 seg.	7 min. 54 seg.	1 min. 6 seg.
2	Iteraciones	14.305	4.298	1.731
	Evalf	19.333	5.174	2.038
	Evalg	14.306	4.299	1.732
	Tiempo	25 min. 47 seg.	15 min. 57 seg.	3 min. 52 seg.
3	Iteraciones	59.371	142.949	69.265
	Evalf	97.338	181.686	104.704
	Evalg	59.372	142.949	69.266
	Tiempo	4h. 9 min. 46 seg.	6h. 53 min. 9 seg.	4h. 33 min. 8 seg.

Tabla 6.11: Convergencia de los puntos solución de velocidades e interfaces para el modelo 2

6.3 Modelo 3

Este modelo es más complejo que los anteriores, presenta una capa más y se incrementa el número de fuentes y receptores. El modelo consta de 5 capas de subsuelo, 10 fuentes y 34 receptores, dando como resultado 1.700 rayos.

Siguiendo la metodología aplicada en los modelos anteriores, la distribución de las fuentes y receptores fue escogida de forma aleatoria y se consideraron tres problemas, el de estimar sólo las velocidades, sólo los desplazamientos verticales y el de estimar velocidades y desplazamientos en conjunto para tres conjuntos de datos, los cuales difieren en las velocidades e interfaces originales, como se muestra en la tabla 6.12.

Velocidades (V)						Interfaz (coordenada z)		
1		2		3		1	2	3
Bajada	Subida	Bajada	Subida	Bajada	Subida			
2.0000	2.0000	2.30000	2.30000	12.000	12.000	1.0700	0.9870	0.6200
2.0000	2.0000	3.10000	3.10000	9.0000	9.0000	2.4900	1.1200	1.0700
2.0000	2.0000	4.60000	4.60000	7.000	7.0000	3.5900	1.6700	1.3900
2.0000	2.0000	5.80000	5.80000	4.0000	4.0000	3.9800	2.3200	1.9400
2.0000	2.0000	6.20000	6.20000	2.2500	2.2500	4.7500	2.5100	2.4300

Tabla 6.12: Velocidades e interfaces reales para el modelo 3

En la tabla 6.13 se pueden observar los valores perturbados de la tabla 6.12.

Velocidades (V)						Desplazamientos verticales (H)		
1		2		3		1	2	3
Bajada	Subida	Bajada	Subida	Bajada	Subida			
2.6000	2.6000	2.9900	2.9900	15.600	15.600	0.3210	0.2961	0.1860
2.6000	2.6000	4.0300	4.0300	11.700	11.700	0.7470	0.3360	0.3210
2.6000	2.6000	5.9800	5.9800	9.1000	9.1000	1.0770	0.5010	0.4170
2.6000	2.6000	7.5400	7.5400	5.2000	5.2000	1.1940	0.6960	0.5820
2.6000	2.6000	8.0600	8.0600	2.9250	2.9250	1.4250	0.7530	0.7229

Tabla 6.13: Punto inicial de velocidades y desplazamientos verticales para el modelo 3

En primer lugar se consideró el problema de estimar sólo velocidades partiendo del punto inicial mostrado en la tabla 6.13, cuyas soluciones son mostradas en la tabla 6.14.

Velocidades (V)					
1		2		3	
Bajada	Subida	Bajada	Subida	Bajada	Subida
2.00000	1.99999	2.30000	2.29999	11.99999	12.00000
2.00000	1.99999	3.10000	3.09999	8.999999	8.999999
1.99999	2.00000	4.59999	4.60000	6.999999	7.000000
2.00000	1.99999	5.79999	5.80000	3.999999	4.000000
2.00000	1.99999	6.20000	6.20000	2.250000	2.249999

Tabla 6.14: Punto solución de velocidades para el modelo 3

Para el primer conjunto de velocidades se requirió de 86 iteraciones, 88 evaluaciones del funcional y 87 evaluaciones del gradiente en un tiempo de 41 seg. 81

centésimas de seg. Para el segundo conjunto de velocidades cuyo tiempo de ejecución fue de 1 min. 48 seg., requirió de 192 iteraciones, 201 evaluaciones del funcional y 193 evaluaciones de gradiente. Finalmente, el último conjunto de velocidades utilizó 195 iteraciones, 207 evaluaciones del funcional y 196 evaluaciones del gradiente en un tiempo de 2 min. 31 seg.

Después se consideró el problema de estimar los desplazamientos H de las interfaces solamente. Se tomó el punto inicial de la tabla 6.13

Para hacer esto se consideró utilizar el conjunto de valores de desplazamientos verticales 1, 2 y 3 utilizando el vector de velocidades iniciales de la columna 3 de la tabla 6.12, para observar el número de iteraciones, evaluación del gradiente y funcional junto con el tiempo de convergencia.

El primer vector de desplazamientos verticales requirió de 5 h. 51 min. 28 seg. para encontrar la solución en 33.979 iteraciones, 45.896 evaluaciones del funcional y 33.980 evaluaciones de gradiente. El segundo vector de desplazamientos necesitó de 42.737 iteraciones, 49.204 evaluaciones de funcional y 42.738 evaluaciones del gradiente en un tiempo de 7 h. 12 min. 12 seg. El tercer y último vector de desplazamientos necesito 43.030 iteraciones, 51.114 evaluaciones del funcional y 43.031 evaluaciones del gradiente requiriendo para la convergencia 7 h. 56 min. 7 seg.

Para finalizar se consideró el problema de estimar velocidades y desplazamientos verticales en conjunto, tomando como punto inicial las perturbaciones hechas en los dos casos anteriores.

En primer lugar se usaron los conjuntos de velocidades 1, 2 y 3 y se compararon con el conjunto de desplazamientos verticales 3 de la tabla 6.12, obteniéndose los resultados mostrados en la tabla 6.15.

El primer conjunto de velocidades se obtuvo en 1.343 iteraciones, 1.573 evaluaciones del funcional, 1.344 evaluaciones del gradiente, y en 11 min. 18 seg. El segundo conjunto de velocidades tardó 29 min. 28 s., con 8.813 iteraciones, 11.437 evaluaciones del funcional y 8.814 evaluaciones de gradiente. El tercer y último conjunto de velocidades requirió de 142.349 iteraciones, 184.502 evaluaciones del funcional y 142.350 evaluaciones del gradiente en un tiempo 21 h. 34 min. y 42 seg.

Velocidades (V)						Desplazamientos verticales (H)
1		2		3		
Bajada	Subida	Bajada	Subida	Bajada	Subida	
2.000000	1.99999	2.30000	2.29999	11.9999	12.00000	0.00000
2.000000	1.99999	3.10000	3.09999	8.99999	8.999999	0.00000
2.000000	1.99999	4.60000	4.59999	6.99999	7.000000	0.00000
2.000000	1.99999	5.79999	5.80000	3.99999	4.000000	0.00000
2.00000	2.00000	6.19999	6.20000	2.25000	2.249999	0.00000

Tabla 6.15: Punto solución de velocidades e interfaces para el modelo 3

Velocidades		Desplazamientos verticales		
		1	2	3
1	Iteraciones	8.660	8.974	1.343
	Evalf	11.876	11.961	1.573
	Evalg	8.661	8.975	1.344
	Tiempo	1h. 16min. 17seg.	2h. 17min. 25seg.	11 min. 18 seg.
2	Iteraciones	17.063	15.678	8.813
	Evalf	23.031	19.213	11.437
	Evalg	17.064	15.679	8.814
	Tiempo	2h. 18min. 35seg.	2h. 5 min. 55 seg.	29 min. 28 seg.
3	Iteraciones	126.865	No converge	142.349
	Evalf	151.201	No converge	184.502
	Evalg	126.866	No converge	142.350
	Tiempo	18h. 50min. 45seg.	No converge	21h. 34min. 42seg.

Tabla 6.16: Convergencia de los puntos solución de velocidades e interfaces para el modelo 3

Observe que el último problema, en el cual se intenta estimar los parámetros que determinan las velocidades y desplazamientos en conjunto, es evidentemente más complicado, esto se refleja en el significativo incremento del número de iteraciones y evaluaciones que realiza el algoritmo.

6.3 Modelo 4

A diferencia de los modelos anteriores, este modelo trabajará con capas planas horizontales y buzantes a la vez. Para ello consideraremos un modelo de 3 capas, 3 fuentes y 8 receptores con los valores de velocidad e interfaces mostrados en la tabla 6.17.

Capa #	Velocidades (V)		Interfaz (coordenada z)		
1	3.5000000	3.5000000	1.0000	1.4000	1.8000
2	2.0000000	2.0000000	2.8900	3.1200	3.6500
3	1.5000000	1.5000000	4.3300	4.3300	4.3300

Tabla 6.17: Velocidades e interfaces reales para el modelo 4

Velocidades (V)		Desplazamientos verticales (H)
Bajada	Subida	
4.5500	4.5500	0.30000
2.6000	2.6000	0.86700
0.4500	0.4500	1.29900

Tabla 6.18: Punto inicial de velocidades y desplazamientos verticales para el modelo 4

En primer lugar se consideró el problema de estimar sólo velocidades partiendo del punto inicial mostrado en la tabla 6.18. Los resultados correspondientes se muestran en la tabla 6.19.

Velocidades (V)		Iteraciones	2.575
3.499999	3.500000	Evalf	3.859
1.999999	2.000000	Evalg	2.576
1.500000	1.499999	Tiempo	1 min. 30 seg.

Tabla 6.19: Punto solución de velocidades para el modelo 4

Después se consideró el problema de estimar los desplazamientos H de las interfaces solamente. Se tomó el punto inicial de la tabla 6.18.

Se utilizaron 45.927 iteraciones, 67.017 evaluaciones de funcional y 45.928 evaluaciones del gradiente en un tiempo de 26 min. 48 seg.

Para finalizar se consideró el problema de estimar velocidades y desplazamientos verticales en conjunto, tomando como punto inicial las perturbaciones hechas en los dos casos anteriores. Los resultados se muestran en la tabla 6.20.

Velocidades (V)		Iteraciones	37.892
3.500000	12.00000	Evalf	56.724
2.000000	1.999999	Evalg	37.893
1.499999	1.500000	Tiempo	22 min. 14 seg.

Tabla 6.20: Convergencia para el modelo 4

CAPÍTULO 7

CONCLUSIONES Y RECOMENDACIONES

En este trabajo se adaptaron los algoritmos de las referencias [2] y [6] para resolver el problema planteado que consiste en encontrar el vector de velocidades de propagación de la onda sísmica y el vector de desplazamientos verticales de las diferentes interfaces del subsuelo. Para ello se realizó una amplia experimentación numérica, para la verificación de la metodología y la implementación propuesta.

- Los resultados numéricos muestran que el algoritmo resuelve el problema de inversión de velocidades y el problema de hallar los parámetros que determinan las interfaces de manera eficiente y precisa.
- La resolución del problema más general en el que se intenta estimar los parámetros que determinan las velocidades e interfaces en conjunto, es evidentemente más complicado. Esto se refleja en el significativo incremento del número de iteraciones que realiza el algoritmo.
- Los resultados indican que la estimación de interfaces es más complicada que la estimación de velocidades.
- A pesar de que los modelos son parecidos, la magnitud del tercer modelo es mayor que la de los otros dos, principalmente en la cantidad de rayos existentes. Y ésta es

la razón principal del incremento en el tiempo de procesamiento tomado por el algoritmo en el modelo 3.

- Para el modelo 4, en el cual se combinan capas planas horizontales y buzantes se obtuvieron buenos resultados, observándose al igual que para los modelos anteriores a éste que la estimación de las velocidades y desplazamientos verticales en conjunto es más complicada que la estimación de sólo los desplazamientos verticales, y que ésta a su vez es más complicada que la estimación de sólo velocidades, en lo que se refiere a número de iteraciones, evaluaciones del gradiente y funcional, y tiempo requerido para la convergencia.
- El criterio de parada utilizado 1×10^{-11} pudiera ser menos estricto, es decir, pudiera estar en un rango no mayor de 1×10^{-8} , con esto se obtienen buenos resultados en un número menor de iteraciones, evaluaciones y tiempo de convergencia.
- Los resultados obtenidos fueron lo suficientemente satisfactorios al compararlos con los logrados en un trabajo precedente a éste.
- A manera de recomendaciones o sugerencias para trabajos posteriores, sería interesante estudiar la relación entre las distribuciones de las fuentes y los receptores y la escogencia de los puntos que definen las interfaces con el costo computacional requerido para resolver el problema, y tratar de determinar la distribución óptima de las fuentes y receptores para acelerar la convergencia del algoritmo.

REFERENCIAS BIBLIOGRÁFICAS

[1] Birgin G. E., Martinez J. M., Raydan M. “Nonmonotone Spectral Projected Gradient Methods on convex sets”. SIAM (Society for Industrial and Applied Mathematics). 10(4):1196-1211, Junio 1999.

[2] Birgin G. E., Martinez J. M., Raydan M. “Algorithm 813: SPG: software for convex-constrained optimization”, Collected algorithms from ACM Transactions on Mathematical Software. 27(3): 340- 349, Septiembre 2001.

[3] Bishop T. N., Bube K. P., Cutler R. T., Langan R. T., Love P. L., Resnick J. R., Shuey R. T., Spindler D.A., Wyld H. W. “Tomographic determination of velocity and depth in laterally varying media”. Geophysics. 50(6): 903-923, Junio 1985.

[4] Castillo Zenaida. “Técnicas eficientes de optimización para la resolución del problema TRS”. Trabajo Especial de Grado, Facultad de Ciencias, Universidad Central de Venezuela, Caracas, Julio 1997.

[5] Chiu S. K. L., Kanasevich E. R., Phadke S. “Three-dimensional determination of structure and velocity by seismic tomography”. Geophysics, 51(8):1559-1571, Agosto 1986.

[6] Cores Débora, Fung Glenn M., Michelena Reinaldo J. “A fast and global two point low storage optimization technique for tracing rays in 2D and 3D isotropic media”. Journal of Applied Geophysics. 45: 273- 287, 2000.

[7] Domínguez Juan. “Estimación 3D de velocidades sísmicas e interfaces usando técnicas de optimización”. Proyecto de Grado, Departamento de Cómputo Científico y Estadística, Universidad Simón Bolívar, Sartenejas, Marzo 2003.

[8] Fung M. Glenn M. “Un trazador de rayos 3D usando técnicas de Optimización no-lineal”. Proyecto de Grado, Departamento de Cómputo Científico y Estadística, Universidad Simón Bolívar, Sartenejas, Noviembre 1996.

[9] Grippo L., Lampariello F. y Lucidi S. “A nonmonotone line search technique for Newton’s method”, SIAM J. Numer. Anal., 23: 707-716, 1986.

APÉNDICE A

Para la resolución del problema planteado, fue necesario adaptar los programas dados en las referencias [2] y [6], y crear módulos con funciones y subrutinas que complementaran e hicieran posible el cumplimiento de la metodología.

En principio se especificarán cuáles cambios se hicieron y las razones para estos.

A.1 Trazador de rayos

Fue necesario modificar el programa trazador de rayos dado en la referencia [6], puesto que éste era un programa principal que consistía en leer de un archivo de datos la estructura del área de estudio y realizar cierto conjunto de operaciones, que a partir de la llamada de otras funciones y subrutinas retorna, después de un proceso iterativo, los tiempos de viaje de cada rayo junto con sus coordenadas, en un archivo de salida.

Esto planteaba un verdadero inconveniente puesto que se perdía un gran tiempo de máquina en leer constantemente de archivos cada vez que se requería al trazador o los datos provenientes del mismo.

Se transformó este programa principal en una subrutina, la cual no lee los datos que caracterizan la estructura del subsuelo mediante un archivo; en lugar de esto dichos datos son pasados como parámetros de entrada. Para la tarea de leer los datos se creó una subrutina nueva (*lectura.f*), la cual fue escrita siguiendo el mismo esquema del trazador de rayos no modificado y que además hace una verificación de los datos entrantes para que cumplan con las condiciones requeridas para el buen funcionamiento del programa, como que los puntos dados para las interfaces sean no colineales, que las velocidades, el número de capas, fuentes y receptores sean distintos de cero y que las interfaces estén espaciadas entre sí. De la misma forma los tiempos de viaje y las

coordenadas de los rayos que calcula el trazador se pasaron a la subrutina como parámetros de salida.

Esta nueva subrutina se llama *rayisoconv3d.f* y los parámetros de salida y entrada respectivamente, especificados en orden, son los siguientes:

Parámetros de Salida

- *tempo*: es un vector con los tiempos de viajes para cada rayo, su tamaño es $ni \cdot ns \cdot nr$.
- *cmatrix*: matriz con las coordenadas (x, y, z) de cada rayo, su tamaño es $nfilas \times 3$, donde *nfilas* es el número total de puntos (x, y, z) que definen los n rayos.

Parámetros de Entrada

- *nr*: número de receptores.
- *ns*: número de fuentes.
- *ni*: número de capas.
- *zextx1*: vector con las primeras coordenadas x del plano, para cada una de las interfaces $ni + 1$.
- *zextx2*: vector con las segundas coordenadas x del plano, para cada una de las interfaces $ni + 1$.
- *a*: vector de velocidades de cada capa, su tamaño es $2ni + 1$.
- *zexty1*: vector con las primeras coordenadas y del plano, para cada una de las interfaces $ni + 1$.
- *zexty2*: vector con las segundas coordenadas y del plano, para cada una de las interfaces $ni + 1$.
- *zextz1*: vector con las primeras coordenadas z del plano, para cada una de las interfaces $ni + 1$.
- *zextz2*: vector con las segundas coordenadas z del plano, para cada una de las interfaces $ni + 1$.

- $zextz3$: vector con las terceras coordenadas z del plano, para cada una de las interfaces $ni + 1$.
- $zextx3$: vector con las terceras coordenadas x del plano, para cada una de las interfaces $ni + 1$.
- $zexty3$: vector con las terceras coordenadas y del plano, para cada una de las interfaces $ni + 1$.
- xs : matriz con las coordenadas (x, y, z) con la ubicación de la(s) fuente(s), su tamaño es $ns \times 3$.
- xr : matriz con las coordenadas (x, y, z) con la ubicación los receptores, su tamaño es $nr \times 3$.

A.2 Optimizador (Gradiente Espectral Proyectado)

Para el código de gradiente espectral proyectado, dado en la referencia [2], fue necesario realizar modificaciones en tres subrutinas: `evalf.f90`, `evalg.f90` y `proj.f90`, las cuales tienen como tarea la evaluación del funcional (4.1), la evaluación del gradiente (4.2) y la proyección respectivamente.

En principio el módulo `spg.f90` tendrá además de los parámetros de entrada especificados para la subrutina trazador de rayos en A.1, los siguientes parámetros de entrada y salida:

Parámetros de Entrada

- x : vector de dimensión n con los valores perturbados de V y H .
- n : dimensión del vector x .
- M : número de búsquedas o *backtracking*.
- eps , $eps2$: tolerancia para el criterio de parada en el método GEP.
- $maxit$: número máximo de iteraciones.
- $maxfc$: número máximo de evaluaciones funcionales.
- $output$: indicador de cantidad de impresión en pantalla durante la ejecución del método GEP.

Parámetros de Salida

- x : vector de dimensión n con los valores finales de V y H .
- f : aproximación funcional.
- $fginf$, $fgtwon$: valores que satisfacen el criterio de parada.
- $Iter$: iteración para la cual sucede la convergencia.
- $fcnt$: número de evaluaciones de la función en el momento de la convergencia.
- $gcnt$: número de evaluaciones del gradiente en el momento de la convergencia.
- $inform$: valor entero que indica el éxito o fracaso en la ejecución del método GEP. En la tabla A.1 se puede observar los valores posibles para esta variable.

<i>inform</i>	Descripción
0 - 1	Convergencia del gradiente espectral proyectado
2	Iteraciones insuficientes para la convergencia
3	Evaluaciones insuficientes para la convergencia
4	Error en la subrutina Proj
5	Error en la subrutina Evalf
6	Error en la subrutina Evalg

Tabla A.1: Valores que puede tomar *inform*

A.2.1 Evaluación del Funcional (EVALF)

Para la evaluación del funcional (4.1), fue necesario pasar como parámetros de entrada, el vector T de los tiempos de viaje reales, y los parámetros correspondientes al trazador de rayos para el cálculo del vector de tiempos de viaje $T(V, H)$, ya discutidos en A.1.

A.2.2 Evaluación del Gradiente (EVALG)

Para la evaluación del gradiente (4.2), fue necesario pasar como parámetros de entrada, el vector T de los tiempos de viaje reales, y los parámetros correspondientes al trazador de rayos discutidos en A.1.

También fue necesario llamar a la subrutina *Jacobiano.f90* contenida en el módulo *fun_jac.f90* para el cálculo de la matriz jacobiana, necesario para el cálculo del gradiente.

A.2.3 Proyección (PROJ)

Los valores de V y H no permanecen constantes sino, por el contrario, cambian durante la ejecución del programa, siempre permaneciendo dentro de un intervalo, el cual es obtenido a través de una información a priori que regulariza la solución y que fue calculada mediante la función *cotas.f90* en el módulo *herramientas*. Dicha función calcula el máximo y el mínimo de los valores de V y H , para establecer las cotas de ambos por medio de porcentajes cuyo valor será definido en base a pruebas hechas con datos sintéticos, puesto que no se posee información del campo.

Cota inferior Velocidad (v):	50%
Cota superior Velocidad (v):	150%
Cota inferior profundidad (h):	0%
Cota superior profundidad (h):	150%

Tabla A.2: Porcentajes para los valores de V y H .

Los parámetros de entrada y salida para la subrutina *proj.f90* son los siguientes:

Parámetros de Entrada

- x : vector de dimensión n con los valores de V y H .
- n : dimensión del vector x .
- ni : número de capas.
- lv : cota inferior para la velocidad V .
- uv : cota superior para la velocidad V .
- lh : cota inferior para la profundidad H .
- uh : cota superior para la profundidad H .

Parámetros de Salida

- x : vector de dimensión n con los nuevos valores de V y H .
- *inform*: variable entera que indica un valor distinto de cero o uno en caso de error (ver cuadro A.1).

Ahora se procede a describir los módulos creados que complementan los códigos anteriormente descritos.

A.3 Jacobiano (Módulo fun_jac)

Este programa efectúa el cálculo de la matriz jacobiana de los tiempos de viaje de los rayos que viajan de un punto fuente x_s a un punto receptor x_r , y que se reflejan en las diferentes capas del subsuelo, como se especificó en la metodología del capítulo 4.

Este módulo está compuesto por varias funciones y subrutinas y es usada por la subrutina EVALG.

A.4 Herramientas (Módulo Herramientas)

A.4.1 Cotas

Esta subrutina calcula las cotas superiores e inferiores de los valores V y H que serán utilizados en la subrutina PROJ; estos valores fueron definidos como en la tabla A.2.

Parámetros de Entrada

- x : vector de dimensión n con los valores perturbados de V y H .
- $x2$: vector de dimensión n con los valores reales de V , no se incluye H debido a que inicialmente este vector es cero.
- n : dimensión del vector x .
- ni : número de capas.

Parámetros de Salida

- lv : cota inferior para la velocidad V .
- uv : cota superior para la velocidad V .
- lh : cota inferior para la profundidad H .
- uh : cota superior para la profundidad H .

A.4.2 Errores

Esta función verifica que se abran y lean los archivos de forma correcta.

A.4.3 Actualizar

Esta subrutina se encarga de trasladar verticalmente, según sea el valor de H , las interfaces.

Parámetros de Entrada

- x : vector de dimensión n con los valores perturbados de V y H .

- n : dimensión del vector x .
- ni : número de capas.
- $zextz1$: vector con las primeras coordenadas z del plano, para cada una de las interfaces $ni + 1$.
- $zextz2$: vector con las segundas coordenadas z del plano, para cada una de las interfaces $ni + 1$.
- $zextz3$: vector con las terceras coordenadas z del plano, para cada una de las interfaces $ni + 1$.

Parámetros de Salida

- $zextz1$: vector con las primeras coordenadas z del plano, para cada una de las interfaces $ni + 1$, al cual se le suma H .
- $zextz2$: vector con las segundas coordenadas z del plano, para cada una de las interfaces $ni + 1$, al cual se le suma H .
- $zextz3$: vector con las terceras coordenadas z del plano, para cada una de las interfaces $ni + 1$, al cual se le suma H .