

SOLUCIÓN NUMÉRICA DE LA ECUACIÓN DE ONDA
ELÁSTICA EN MEDIOS CON ANISOTROPÍA TIPO VTI Y TTI
BASADO EN ESQUEMAS NUMÉRICOS DE DIFERENCIAS
FINITAS USANDO MALLA INTERCALADA MEDIANTE
COMPUTACIÓN DE ALTO DESEMPEÑO

Autor

WILMER ANDRES MANTILLA SUAREZ

Director

JOSÉ ORLANDO MALDONADO BAUTISTA

CoDirector

*WILSON ALBERTO CONTRERAS ESPINOSA
JAIRO ALBERTO TORRES PEÑA*

**UNIVERSIDAD DE PAMPLONA
FACULTAD INGENIERÍAS Y ARQUITECTURA
DEPARTAMENTO DE
ELECTRÓNICA, ELÉCTRICA, TELECOMUNICACIONES Y SISTEMAS**

PAMPLONA, 7 de diciembre de 2017

Agradecimientos

Le Agradezco a Dios por haberme acompañado y guiado a lo largo de mi carrera, A la UNIVERSIDAD DE PAMPLONA por darme la oportunidad de estudiar y ser un profesional y mi más profundo y sincero agradecimiento a mi directores de tesis, JOSÉ ORLANDO MALDONADO, WILSON ALBERTO CONTRERAS y JAIRO ALBERTO TORRES quienes con sus conocimientos, experiencia, dedicación, paciencia y su motivación ha logrado motivarme en en el desarrollo de este proyecto. Y, por supuesto, el agradecimiento más profundo y sentido va para mi familia. Sin su apoyo, colaboración e inspiración habría sido imposible llevar a cabo esta ardua tarea.

Índice general

0.1.	Resumen	6
0.2.	Introducción	6
0.2.1.	Objetivos	7
0.2.1.1.	Objetivo General	7
0.2.1.2.	Objetivos Específicos	7
0.3.	Marco Teórico	7
0.3.1.	Modelado Sísmico	7
0.3.2.	Medio isótropo	8
0.3.3.	Anisotropía Sísmica	8
0.3.4.	Modelo de Velocidad	8
0.3.5.	Ondas de Cuerpo o de Volumen	9
0.3.5.1.	Ondas P	9
0.3.5.2.	Onda Secundaria	9
0.3.6.	Ondas Superficiales	9
0.3.6.1.	Ondas Love y Ondas Rayleigh	9
0.3.7.	Ecuación de la onda elástica	9
0.3.8.	Simetría en medios elásticos	11
0.4.	Isotropía transversal TI	12
0.4.1.	Medio Elástico con Simetría Isotropa	12
0.4.2.	Ecuaciones de Campo Para un Medio Elástico Isotropo	12
0.4.2.0.1.	Ecuaciones de Conservación del Momentum.	12
0.4.2.0.2.	Relaciones de Esfuerzo-Deformación	13
0.4.3.	Medio Elástico con simetría Anisotropa Polar o VTI	13
0.4.4.	Ecuaciones de Campo Para un Medio Elástico con Simetría anisotropía Polar o VTI	14
0.4.4.0.3.	Ecuaciones de Conservación del Momentum.	14
0.4.4.0.4.	Relaciones de Esfuerzo-Deformación	15
0.4.5.	Medio TTI	15
0.4.6.	Ecuaciones de Campo Para un Medio Elástico con Simetría TTI	16
0.4.6.0.5.	Ecuaciones de Conservación del Momentum.	16
0.4.6.0.6.	Relaciones de Esfuerzo-Deformación	16
0.5.	Modelado Numérico	17
0.5.1.	Diferencias Finitas	17
0.5.2.	Derivadas en Dos Dimensiones	19
0.5.3.	Malla intercalada 2D	20
0.5.3.1.	Aproximación Temporal y Espacial	21
0.5.4.	Solución Numérica para un Medio isótropo	24

0.5.5.	Solución Numérica para un Medio VTI	24
0.5.6.	Solución Numérica para un Medio TTI	25
0.5.7.	Condición de Estabilidad	25
0.5.8.	Condición de dispersión	26
0.5.9.	Condición de contorno	27
0.5.10.	Fuente	28
0.5.10.1.	Ondicula	28
0.6.	Computación de alto rendimiento	29
0.6.1.	Paralelismo	29
0.6.2.	Arquitecturas Paralelas	30
0.6.3.	Paradigmas de programación orientados a la arquitectura	30
0.6.3.1.	Paradigma de memoria compartida	30
0.6.3.2.	Paradigma de Paso de Mensajes	30
0.6.4.	Computación acelerada con GPU	30
0.6.5.	Procesamiento paralelo con CUDA	31
0.6.6.	Modelo CUDA	31
0.6.6.1.	CUDA Threads	32
0.7.	Experimentos numéricos	34
0.7.1.	Implementación Computacional	34
0.7.1.1.	Kernel en CPU que permite calcular las velocidades.	35
0.7.1.2.	Kernel en CPU para calcular las tensiones.	36
0.7.1.3.	Kernel en CPU paralelizada que permite calcular las velocidades.	36
0.7.1.4.	Kernel en CPU paralelizada para calcular las tensiones.	37
0.7.1.5.	Kernel en GPU para calcular las velocidades	38
0.7.1.6.	Kernel en GPU para calcular las tensiones	39
0.7.2.	Medio homogéneo	41
0.7.3.	Medio no-homogéneo	43
0.7.3.1.	MARMOUSI2	43
0.7.3.2.	HESS VTI	46
0.8.	Análisis de Resultados	50
0.8.1.	Comparación del tiempo de ejecución: CPU vs GPU	50
0.9.	Conclusiones y Recomendaciones	51

Índice de figuras

1.	Ejemplo de un modelo de velocidades	8
2.	Medio con anisotropía polar vertical (VTI)	13
3.	interpretación geométrica de las aproximaciones de diferencias finitas hacia adelante, hacia atrás y diferencias centrada	18
4.	Ubicación de las componentes de la velocidad y tensión en una grilla escalonada	20
5.	Ejemplo de inestabilidad	26
6.	Ejemplo de dispersión numérica	27
7.	Esquema tomado de (25) se muestra un diagrama esquemático de las condiciones de contorno propuesta por (18) . Donde A_1, A_2, A_3 y A_4 es contorno del modelo original y B_1, B_2, B_3 y B_4 es el área de contorno artificial que atenúa las reflexión.	28
8.	GPU,CPU	32
9.	Jerarquía de Threads	32
10.	Diagrama de flujo del algoritmo implementado	34
11.	Implementación computacional de cada una de las componentes principales a) CPU b) GPU	35
12.	Capturas para la componente vertical en un medio VTI tanto en CPU como GPU	41
13.	Capturas para la componente vertical en un medio TTI con un ángulo de inclinación de 30° tanto en CPU como GPU	42
14.	Capturas para la componente vertical en un medio TTI con un ángulo de inclinación de 45° tanto en CPU como GPU	42
15.	Capturas para la componente vertical en un medio TTI con un ángulo de inclinación de 60° tanto en CPU como GPU	43
16.	Marmousi2. Versión elástica isotrópica de los datos originales de Marmousi.	44
17.	Capturas de Frente de onda modelo MARMOUSI2 (a) a (f)	45
18.	Modelo Hess Corporation VTI. Disponible desde SEG.	46
19.	Capturas de Frente de onda modelo HESS VTI (a) a (f)	48
20.	Capturas para la componente vertical en un medio VTI tanto en CPU como GPU	49
21.	Capturas para la componente vertical en un medio VTI tanto en CPU como GPU fuente ubicada en el centro del modelo	49
22.	Tiempo de procesamiento CPU vs GPU	51
23.	Tiempo de procesamiento CPU vs GPU	51

Índice de tablas

1.	Tipos de simetría en medios elásticos (22)	11
2.	Constantes elásticas para un medio TI,tomadas de (11)	41
3.	Parámetros que se tuvieron en cuenta durante la simulación.	41
4.	Parámetros que se tuvieron en cuenta durante la simulación.	47
5.	Tiempos promedio de procesamiento de CPU y GPU para un tamaño de dado de malla.	50
6.	Tiempo de procesamiento modelo HESS VTI	51

Capítulo 1

0.1. Resumen

El presente proyecto inicia con la revisión teórica del fenómeno ondulatorio en un medio continuo elástico en 2D, posteriormente se procede a la investigación e implementación de la solución de la ecuación de onda elástica en medio isótropo mediante el uso de esquemas numéricos basados en diferencias finitas de malla intercalada con algoritmos secuenciales, A partir de dicha revisión se extiende a simetría tipo VTI y TTI aproximando mediante esquemas numéricos de diferencias finitas usando malla intercalada mediante computación de alto desempeño utilizando la plataforma CUDA sobre GPUs de arquitectura NVIDIA.

0.2. Introducción

Los métodos explícitos de diferencias finitas han asumido una papel en el modelado directo en sismología computacional debido a su capacidad para modelar con precisión la propagación de ondas en medios lateralmente heterogéneos. Lamentablemente los esquemas explícitos son computacionalmente costosos, requieren grandes cantidades de memoria de la computadora para modelar la escala de problemas de exploración. Actualmente son factibles solo casos bidimensionales (2-D) y pequeño problemas tridimensionales (3-D). Madariaga (1976) desarrollo en una cuadrícula escalonada el esquema numérico a partir del sistema de ecuaciones elásticas e hiperbólicas de primer orden expresadas en términos de velocidades y tensiones de una partícula, Virieux (1984, 1986) adaptó este esquema al modelado directo general de ondas SH y P-SV en un sistema cartesiano bidimensional (17).

Este trabajo se ha centrado en el estudio de la ecuación de onda anisotrópica completa en medios que exhiben ciertas simetrías, como en los medios con isotropía transversal TI. La isotropía transversal TI depende de la orientación del eje de simetría porque tiene una simetría más extendida y se usa en el procesamiento de datos y la construcción de modelos de velocidad con una variedad de aplicaciones prácticas. De manera similar ocurre con la isotropía transversal vertical VTI, la isotropía transversal horizontal HTI y la isotropía transversal inclinada TTI. La implementación de estos fenómenos basados en algoritmos numéricos con bucles iterativos es computacionalmente costosa, lo que limita la eficiencia de la exploración presentamos la implementación de la solución de la ecuación de onda en medios isotrópos mediante el uso de esquemas numéricos basados en diferencias finitas con el método de cuadrícula escalonada mediante algoritmos secuenciales implementados en la CPU. A partir de esta revisión extendimos a la solución numérica de la ecuación de onda elástica en medios con anisotropía de tipo VTI y TTI utilizando esquemas nu-

méricos de diferencias finitas con una precisión de octavo orden en el espacio y segundo orden en el tiempo $O(\Delta t^2, \Delta h^8)$. Esta simulación se realizó a través de la computación de alto rendimiento utilizando la plataforma CUDA en arquitecturas GPU NVIDIA. Hoy la tecnología GPU nos ha permitido realizar varias simulaciones físicas como las ondas sísmicas obteniendo cálculos más rápidos en un tiempo de ejecución más corto que solo el uso de la CPU. Los resultados muestran el rendimiento obtenido con la implementación, lo que permite una gran reducción en los tiempos de procesamiento, con beneficio directo en el modelado de este tipo de fenómenos.

0.2.1. Objetivos

0.2.1.1. Objetivo General

- Diseñar un algoritmo que implemente una solución numérica de la ecuación de onda elástica en medios con anisotropía tipo VTI y TTI basado en esquemas numéricos de diferencias finitas usando malla intercalada mediante computación de alto desempeño.

0.2.1.2. Objetivos Específicos

- Revisión de la teoría del medio continuo elástico en 1D Y 2D.
- Revisión de los esquemas numéricos basados en diferencias finitas de malla intercalada
- Diseñar e implementar el algoritmo para la ecuación de ondas elástica 2D en medios con anisotropía VTI y TTI en un modelo secuencial
- Diseñar e implementar el algoritmo para la ecuación elástica 2D en medios con anisotropía VTI y TTI mediante computación de alto desempeño en CUDA.
- Comparar los resultados en términos de rapidez de ejecución, precisión y grado de paralelismo (CUDA)

0.3. Marco Teórico

0.3.1. Modelado Sísmico

Una de las principales preocupaciones en sismica se centra necesariamente en la determinación del verdadero medio superficial ya que las aproximaciones isotrópicas conducen a inexactitudes del posicionamiento de las zonas en el interior de la tierra. Claramente, la precisión de esta información tiene un impacto significativo en todos los aspectos del proceso de exploración. Las estructuras geológicas del subsuelo son generalmente heterogéneas y anisotrópicas; varias rocas y materiales exhiben un comportamiento anisotrópico (Thomsen 1986), es decir sus propiedades físicas dependen de la dirección a través del material en que la medida se hace. La isotropía transversal (TI) es uno de los más comunes modelos

anisotrópicos encontrados en el subsuelo y principalmente asociado con formaciones sedimentarias y capas finas periódicas con orientación paralela de minerales de arcilla con forma de placa (9). Las capas isotrópicas son planos paralelos en el caso en que el eje de simetría sea vertical, se hace referencia a un medio con isotropía transversal vertical (VTI), mientras que si el eje de simetría y el eje vertical forman un ángulo θ , este es descrito por modelo de isotropía transversal inclinada (TTI).

0.3.2. Medio isótropo

Los materiales son “isótropos” cuando sus propiedades físicas se mantienen constantes a través de ellos sin cambiar su magnitud, independientemente del ángulo u orientación en el cual se estén midiendo. Esta palabra proviene de ‘iso’, que significa lo mismo, y ‘tropikos’, que significa algo que ver con dar vuelta. La idea es que los materiales isotrópicos tengan el mismo aspecto en todas las direcciones: no tienen orientación y podemos realizar mediciones en cualquier dirección y obtener el mismo resultado independientemente de un medio homogéneo, que hace claridad en la uniformidad de la composición. Los modelos elásticos isótropos se describen por densidad $\rho(x, y, z)$, velocidad de compresión $Vp(x, y, z)$ y velocidad de corte $Vs(x, y, z)$. los modelos elásticos soportan dos campos de ondas, uno de los cuales es una onda de compresión y el otra es una onda de corte. Las ondas compresionales en dichos modelos son idénticas a las de modelos acústicos.

0.3.3. Anisotropía Sísmica

Se define la anisotropía sísmica como la variación de la velocidad con la dirección en la cual se propagan las ondas y esta se presenta en rocas como las arcillas, estratos delgados superpuestos y formaciones con alto grado de fracturamiento (referencia) los modelos anisótropos representan la tierra en su forma mas compleja, se dice que es anisótropo siempre que la velocidad de la onda esta en función del ángulo de propagación (5)

0.3.4. Modelo de Velocidad

Los modelos de velocidad son una herramienta importante utilizada por los geólogos y geofísicos para representar las capas al interior de la tierra por donde las ondas sísmicas viajan. Estos son creados a partir de información recibida por geófonos en forma de trazas, los modelos de velocidad están conformados por capas y velocidades asociadas a cada una de estas, tienen una cobertura en distancia horizontalmente y otra en profundidad.(8)

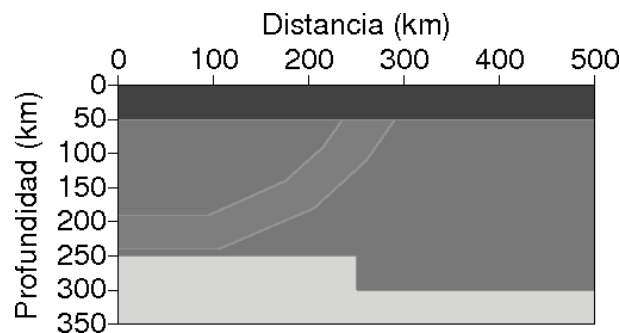


Figura 1: Ejemplo de un modelo de velocidades

0.3.5. Ondas de Cuerpo o de Volumen

Las ondas elásticas generan fuerzas y deformaciones que obedecen la teoría de la elasticidad, en la cual los cuerpos sólidos tienen la propiedad de resistir cambios de tamaño o de forma, y de regresar a la condición no deformada cuando se eliminan las fuerzas externas. Los sismos generan dos tipos de ondas elásticas que se propagan a través del medio: las ondas de cuerpo o de volumen y las ondas superficiales, la velocidad de propagación depende de la densidad del medio y de sus propiedades elásticas tales como el módulo de incompresibilidad y el módulo de rigidez.

0.3.5.1. Ondas P

La principal característica de las ondas P es que comprime y expande la roca en forma alternada en la misma dirección en que se propaga son ampliamente utilizadas en la búsqueda de hidrocarburos, y otros materiales de la corteza terrestre de la tierra ya que las ondas sísmicas P pueden viajar a través de todo tipo de medios incluyendo sólidos, líquidos, y gases, pero sus velocidades cambian dependiendo de los módulos y densidad del material.

0.3.5.2. Onda Secundaria

Onda secundaria u onda S cuya onda viaja con menor velocidad que la onda P esta se propaga lateralmente respecto a su trayectoria.

0.3.6. Ondas Superficiales

0.3.6.1. Ondas Love y Ondas Rayleigh

Este tercer tipo de ondas son comúnmente llamada ondas superficiales tienen la característica de propagarse sobre la parte más superficial de la corteza terrestre disminuyendo la amplitud de su movimiento a medida que la profundidad aumenta. Las ondas Love deforman las rocas de la misma manera que las ondas S Las ondas rayleigh tienen su movimiento vertical similar al del mar, en comparación a las ondas internas las ondas superficiales viajan más despacio que las ondas internas.

0.3.7. Ecuación de la onda elástica

La propagación de la onda dentro de la tierra se ve afectada por las propiedades elásticas de la roca. La teoría de la elasticidad relaciona la fuerza aplicada y los cambios en la forma o tamaño del medio, queda expresado en términos de esfuerzo y deformación. La ecuación que representa la ley de Newton para un medio continuo linealmente elástico tridimensional.

$$\rho \frac{\partial^2 U_i}{\partial t^2} = \sum_{j=1}^3 \frac{\partial \tau_{ij}}{\partial x_j}, \quad i = \{1, 2, 3\} \quad (1)$$

Donde U_i es el desplazamiento de una partícula en un tiempo t y posición x_j , τ_{ij} es la tensión (fuerza por unidad de área) escalar. Conociendo que el desplazamiento y la tensión

en función de la posición y el tiempo $u(x, t)$, $\tau(x, t)$. La expresión de la deformación $\varepsilon_{k,l}$ en función de los desplazamientos

$$\varepsilon_{k,l} = \frac{1}{2} \left(\frac{\partial U_k}{\partial x_l} + \frac{\partial U_l}{\partial x_k} \right) \quad k, l = 1, 2, 3 \quad (2)$$

donde los esfuerzos $\varepsilon_{k,l}$ se calculan a partir de desplazamientos U_i . la ley que permite relacionar la fuerza aplicada con la deformación resultante es la ley de Hooke quien expresa las componentes del tensor de esfuerzos $\sigma_{i,j}$ como una combinación lineal de las componentes del tensor de deformación $\varepsilon_{k,l}$ se puede expresar como:

$$\sigma_{i,j} = C_{i,j,kl} \varepsilon_{k,l} \quad i, j, k, l = 1, 2, 3 \quad (3)$$

Las Constante $C_{i,j,kl}$. describe las propiedades del medio los dos primeros subíndices están relacionados con la tension y los dos últimos con la deformación Las componentes del tensor de elasticidad o tensor de rigidez $C_{i,j,kl}$, dado que varían de 1 a 3, el numero de componentes del tensor de elasticidad es $3^4 = 81$, pero no todas las 81 componentes son diferentes debido a la existencia de simetría de los tensores de esfuerzo ($\sigma_{i,j} = \sigma_{j,i}$) y deformación ($\varepsilon_{k,l} = \varepsilon_{l,k}$), y además por ser un fenómeno adiabático $C_{i,j,kl} = C_{k,l,i,j}$ esto implica que el tensor C las constantes elásticas independientes se reduzcan a 21 (Love (1927)).

Usando la ley de conservación del momentum y sustituyendo x, y, z por i, j, k

$$\begin{aligned} \frac{1}{\rho} \left[\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xz}}{\partial z} + \frac{\partial \sigma_{xy}}{\partial y} - F_x \right] &= \frac{\partial^2 U_x}{\partial t^2} \\ \frac{1}{\rho} \left[\frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \sigma_{zy}}{\partial z} + \frac{\partial \sigma_{xy}}{\partial x} - F_y \right] &= \frac{\partial^2 U_y}{\partial t^2} \\ \frac{1}{\rho} \left[\frac{\partial \sigma_{zz}}{\partial z} + \frac{\partial \sigma_{xz}}{\partial x} + \frac{\partial \sigma_{zy}}{\partial y} - F_z \right] &= \frac{\partial^2 U_z}{\partial t^2} \end{aligned} \quad (4)$$

la aceleración de la partícula puede ser calculada.

Usando la configuración $1 \leftarrow xx, 2 \leftarrow yy, 3 \leftarrow zz, 4 \leftarrow zy, 5 \leftarrow xz, 6 \leftarrow xy$ la matrices σ y ε se convierten en vectores de 6 elementos y el tensor de rigidez C se convierte en matriz de 6x6.

Entonces se puede escribir la relación esfuerzo-deformación como sigue:

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{zy} \\ \sigma_{xz} \\ \sigma_{xy} \end{bmatrix} = \begin{pmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_{12} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{13} & C_{23} & C_{33} & C_{34} & C_{35} & C_{36} \\ C_{14} & C_{24} & C_{34} & C_{44} & C_{45} & C_{46} \\ C_{15} & C_{25} & C_{35} & C_{45} & C_{55} & C_{56} \\ C_{16} & C_{26} & C_{36} & C_{46} & C_{56} & C_{66} \end{pmatrix} \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \varepsilon_{zy} \\ \varepsilon_{xz} \\ \varepsilon_{xy} \end{bmatrix} \quad (5)$$

tomado de (1).

Ya que sólo los componentes en el plano $x - z$ se excitan por lo tanto xx, xz, zz son los únicos componentes que no son nulas y las dos velocidades de las partículas desconocidas V_x y V_z , debido a la invariancia en la dirección y todas las derivadas con respecto a esta desaparecen.

0.3.8. Simetría en medios elásticos

De acuerdo al tipo de simetría que puede tener un sólido (las rocas), existe un número de constantes elásticas necesarias para describir su comportamiento elástico. Entre menos complejo sea el tipo de simetría, menos constantes se necesitan para caracterizar. En total hay seis tipos de simetría en los medios elásticos.

Tipo de Simetría	# Cte	Ejemplos de casos geofísicos contienen este tipo de simetría
Isótropo	2	-
Cúbica	3	Muy pocos casos geofísicos, ejemplo, cristal de Halita
Hexagonal	5	Esquistos, secuencias de capas delgadas, o bien sistema de fracturas paralelas y verticales
Ortorrómica	9	Capas delgadas horizontales afectadas por un sistema de fracturas paralelas y verticales
Monoclínica	13	Material afectado por dos familias no ortogonales de fracturas paralelas.
Triclínica	21	Medios con múltiples sistemas de fracturas

Tabla 1: Tipos de simetría en medios elásticos (22)

Tradicionalmente los datos sísmicos se interpretan asumiendo que el medio superficial es isotrópico la propagación de las ondas sísmicas en un medio anisotrópico son diferentes de propagación en un medio isotrópico. Para definir completamente un medio anisotrópico son necesarias 21 constantes elásticas, mientras que en el caso de un medio isotrópico solo son necesarias 2 constantes elásticas independientes en los medios transversalmente isotrópicos son necesarias 5 constantes elásticas independientes. (7) Esta investigación se enfocó en la simetría transversal o hexagonal, la cual muestra como casos geofísicos sistemas de fracturas verticales y paralelas.

Capitulo 2

0.4. Isotropía transversal TI

Este tipo de simetría también es llamado anisotropía azimutal (Thomsen, 2002) y es el tipo más simple de simetría real porque tiene un solo eje de simetría de rotación (aquí 2 de los 3 ejes son equivalentes entre sí, y diferentes del tercero).

0.4.1. Medio Elástico con Simetría Isotropa

Un medio isótropo es un caso especial de un medio transversal isotrópico, una diferencia importante entre la propagación de ondas en medios isotrópicos y transversalmente isotrópicos es que en este último caso la curva de velocidad de fase y el frente de onda no son congruentes ni circulares (16). Para un medio isótropo resulta el conjunto de ecuaciones de conservación del momentum y las de campo de esfuerzo deformación dadas respectivamente:

$$C_{\text{Isotropo}} = \begin{pmatrix} C_{11} & C_{12} & C_{12} & & & \\ C_{12} & C_{11} & C_{12} & & & \\ C_{12} & C_{12} & C_{11} & & & \\ & & & C_{44} & & \\ & & & & C_{44} & \\ & & & & & C_{44} \end{pmatrix} \quad (6)$$

0.4.2. Ecuaciones de Campo Para un Medio Elástico Isotropo

0.4.2.0.1 Ecuaciones de Conservación del Momentum.

$$\begin{aligned} \rho \frac{\partial v_x}{\partial t} &= \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{xz}}{\partial z} \\ \rho \frac{\partial v_z}{\partial t} &= \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{zz}}{\partial z} \end{aligned} \quad (7)$$

0.4.2.0.2 Relaciones de Esfuerzo-Deformación

$$\begin{aligned}\frac{\partial \tau_{xx}}{\partial t} &= C_{11} \frac{\partial v_x}{\partial x} + C_{12} \frac{\partial v_z}{\partial z} \\ \frac{\partial \tau_{zz}}{\partial t} &= C_{12} \frac{\partial v_x}{\partial x} + C_{11} \frac{\partial v_z}{\partial z} \\ \frac{\partial \tau_{xz}}{\partial t} &= C_{44} \left(\frac{\partial v_x}{\partial z} + \frac{\partial v_z}{\partial x} \right)\end{aligned}\tag{8}$$

para un medio isótropo los coeficientes están dados por $C_{11} = C_{33} = \lambda + 2\mu$, $C_{44} = \mu$, $C_{13} = \lambda$ y $C_{15} = C_{35} = 0$ donde λ y μ son los parámetros de Lamé relacionados a la onda P y S, las velocidades de la partícula V_p y V_s de un medio isótropo. están dadas respectivamente:

$$V_p = \sqrt{\frac{c_{11}}{\rho}} = \sqrt{\frac{\lambda + 2\mu}{\rho}}\tag{9}$$

$$V_s = \sqrt{\frac{c_{44}}{\rho}} = \sqrt{\frac{\mu}{\rho}}\tag{10}$$

0.4.3. Medio Elástico con simetría Anisotropa Polar o VTI

La isotropía transversal vertical (VTI), proporciona una descripción adecuada de la propagación de onda en un medio de capas delgadas con interfaces horizontales (rocas estratificadas) y formaciones de esquistos estratificados. Las ondas que se propagan en medio de este tipo exhiben una simetría alrededor vertical o profundidad eje z .

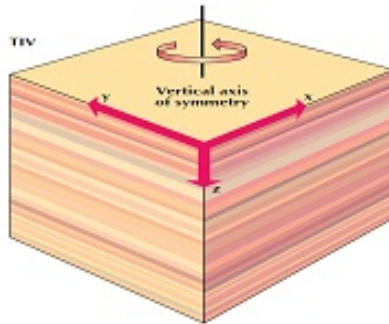


Figura 2: Medio con anisotropía polar vertical (VTI)

Un medio Elástico con simetría Anisotropa Polar o VTI esta completamente especificado por cinco “constantes” independientes elásticas.

$$C_{\text{Aniso Polar}} = \begin{pmatrix} C_{11} & C_{12} & C_{13} & & & \\ C_{12} & C_{11} & C_{13} & & & \\ C_{13} & C_{13} & C_{33} & & & \\ & & & C_{44} & & \\ & & & & C_{44} & \\ & & & & & C_{66} \end{pmatrix} \quad (11)$$

Thomsen introdujo tres parámetros anisótropos, ϵ , δ , y γ , para describir la anisotropía débil. Estos parámetros pueden ser calculados con el tensor de rigidez del medio anisótropo. α es la velocidad de la onda P y β es la velocidad de la onda S. Además el coeficiente C_{66} esta expresado en función de C_{11} y C_{12} por:

$$C_{66} = \frac{C_{11} - C_{12}}{2}$$

$$\begin{aligned} \alpha &= \sqrt{C_{33}/\rho} \\ \beta &= \sqrt{C_{44}/\rho} \\ \epsilon &= \frac{C_{11} - C_{33}}{2C_{33}} \\ \gamma &= \frac{C_{66} - C_{44}}{2C_{44}} \\ \delta &= \frac{(C_{13} + C_{44})^2 - (C_{33} - C_{44})^2}{2C_{33}(C_{33} - C_{44})} \end{aligned}$$

La matriz de Voigt en 2D se reduce a lo siguiente

$$C_{\mathbf{ij}} = \begin{pmatrix} C_{11} & C_{13} & 0 \\ C_{13} & C_{33} & 0 \\ 0 & 0 & C_{55} \end{pmatrix} \quad (12)$$

0.4.4. Ecuaciones de Campo Para un Medio Elástico con Simetría anisotropía Polar o VTI

Para nuestro medio anisótropo polar resulta el conjunto de ecuaciones de conservación del momentum y las de campo de esfuerzo deformación dadas respectivamente:

0.4.4.0.3 Ecuaciones de Conservación del Momentum.

$$\begin{aligned} \rho \frac{\partial v_x}{\partial t} &= \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{xz}}{\partial z} \\ \rho \frac{\partial v_z}{\partial t} &= \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{zz}}{\partial z} \end{aligned} \quad (13)$$

0.4.4.0.4 Relaciones de Esfuerzo-Deformación

$$\begin{aligned}
\frac{\partial \tau_{xx}}{\partial t} &= C_{11} \frac{\partial v_x}{\partial x} + C_{13} \frac{\partial v_z}{\partial z} \\
\frac{\partial \tau_{zz}}{\partial t} &= C_{13} \frac{\partial v_x}{\partial x} + C_{33} \frac{\partial v_z}{\partial z} \\
\frac{\partial \tau_{xz}}{\partial t} &= C_{44} \left(\frac{\partial v_x}{\partial z} + \frac{\partial v_z}{\partial x} \right)
\end{aligned} \tag{14}$$

0.4.5. Medio TTI

Para obtener una simetría TTI tomados de partimos del eje de simetría del medio VTI el cual es rotado por un ángulo de inclinación ϕ en el plano $x - z$ cabe mencionar que ϕ tiene un valor positivo durante la rotación en el sentido de las agujas del reloj y valor negativo en el sentido contrario. La matriz de constantes elásticas tienen seis componentes independientes y se pueden reescribir desde la matriz del medio VTI haciendo uso del tensor de rotación.

$$C_{\mathbf{TTI}} = \begin{pmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ & & C_{33} & C_{34} & C_{35} & C_{36} \\ & & & C_{44} & C_{45} & C_{46} \\ & 0 & & & C_{55} & C_{56} \\ & & & & & C_{66} \end{pmatrix} \tag{15}$$

En 2D se reduce a lo siguiente

$$C_{\mathbf{ij}} = \begin{pmatrix} C_{11} & C_{13} & C_{15} \\ C_{13} & C_{33} & C_{35} \\ C_{15} & C_{35} & C_{55} \end{pmatrix} \tag{16}$$

los coeficientes de la matriz $C_{\mathbf{ij}}^{TTI}$ se pueden expresar de la siguiente forma:

$$C_{11} = (2(\epsilon \cos^2 \phi + \delta \sin^2 \phi) \cos^2 \phi + 1) C_{33}^{VTI} \tag{17}$$

$$C_{13} = (1 + \delta + 2(\epsilon - \delta) \sin^2 \phi \cos^2 \phi) C_{33}^{VTI} - 2C_{55}^{VTI} \tag{18}$$

$$C_{15} = (\epsilon \cos^2 \phi - 0,5\delta \cos 2\phi) C_{33}^{VTI} \sin 2\phi \tag{19}$$

$$C_{33} = (1 + 2\epsilon \sin^4 \phi + 2\delta \sin^2 \phi \cos^2 \phi) C_{33}^{VTI} \tag{20}$$

$$C_{35} = (\epsilon \sin^2 \phi \sin 2\phi + 0,5\delta \sin 2\phi \cos 2\phi) C_{33}^{VTI} \tag{21}$$

$$C_{55} = 0,5(\epsilon - \delta) \sin^2 2\phi C_{33}^{VTI} + C_{55}^{VTI} \tag{22}$$

0.4.6. Ecuaciones de Campo Para un Medio Elástico con Simetría TTI

Las cinco ecuaciones que describen la propagación de las ondas sísmicas en un medio elástico 2-D TTI son:

0.4.6.0.5 Ecuaciones de Conservación del Momentum.

$$\begin{aligned}\rho \frac{\partial v_x}{\partial t} &= \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{xz}}{\partial z} \\ \rho \frac{\partial v_z}{\partial t} &= \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{zz}}{\partial z}\end{aligned}\tag{23}$$

0.4.6.0.6 Relaciones de Esfuerzo-Deformación

$$\begin{aligned}\frac{\partial \tau_{xx}}{\partial t} &= C_{11} \frac{\partial v_x}{\partial x} + C_{13} \frac{\partial v_z}{\partial z} + C_{15} \left(\frac{\partial v_x}{\partial z} + \frac{\partial v_z}{\partial x} \right) \\ \frac{\partial \tau_{zz}}{\partial t} &= C_{13} \frac{\partial v_x}{\partial x} + C_{33} \frac{\partial v_z}{\partial z} + C_{35} \left(\frac{\partial v_x}{\partial z} + \frac{\partial v_z}{\partial x} \right) \\ \frac{\partial \tau_{xz}}{\partial t} &= C_{13} \frac{\partial v_x}{\partial x} + C_{35} \frac{\partial v_z}{\partial z} + C_{55} \left(\frac{\partial v_x}{\partial z} + \frac{\partial v_z}{\partial x} \right)\end{aligned}\tag{24}$$

Capítulo 3

0.5. Modelado Numérico

En el modelado de ondas sísmicas se han propuesto diferentes esquemas para la implementación uno de ellos que ha sido ampliamente utilizados es por medio del método de diferencias finitas este permite modelar problemas de mayor complejidad y modelar una frontera en medios acústicos y elásticos de una forma mas estable. El método de diferencias finitas en tal sentido se pueden destacar los esquemas explícitos en el dominio temporal de segundo orden, que modelan la propagación de ondas en casos homogéneos y heterogéneos con isotropía o los esquemas de malla escalonada de segundo orden basados en las ecuaciones de onda elástica. El algoritmo de diferencias finitas con cuadrícula escalonada resuelve el efecto elastodinámico de ecuaciones de primer orden expresadas en términos de velocidad y tensión cuya formulación de velocidad-tension fue utilizada por primera vez por Madariaga (1976) para modelar fallo-dinámica de ruptura. Esta técnica se extendió al modelado de propagación de ondas sísmicas en 2D (Virieux, 1984, 1986, Levander, 1988) y medios 3D (Randall, 1989). las ventajas de esta formulación son:

1. La inserción de la fuente puede expresarse por Velocidad o Tension;
2. Se extiende fácilmente aproximaciones de alto orden.
3. El algoritmo se implementa fácilmente de forma escalar y vectorial.

0.5.1. Diferencias Finitas

La idea principal de la representación del método de diferencias finitas nace de la definición de la derivada en el calculo elemental:

$$U'(x) = \lim_{\Delta x \rightarrow 0} \frac{U(x_i + \Delta x) - U(x_i)}{\Delta x}$$

Computacionalmente para resolver el limite de una función cuando $\Delta x \rightarrow 0$ por lo tanto procedemos aproximar la función en una grilla discreta cuyas aproximaciones de diferencias finitas se pueden realizar hacia adelante, hacia atrás y central.

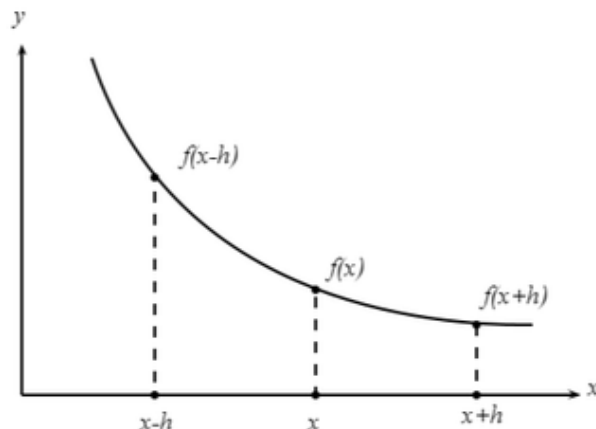


Figura 3: interpretación geométrica de las aproximaciones de diferencias finitas hacia adelante, hacia atrás y diferencias centrada

Usamos la notación $U_i = U(x_i)$ denota la función evaluada en un punto x_i , los nodos son divididos a lo largo de los ejes en intervalos de tamaño $\Delta x_i = x_{i+1} - x_i$

Euler hacia adelante

$$U'(x) = \frac{U(x_i + \Delta x) - U(x_i)}{\Delta x}$$

reemplazando

$$U'(x) = \frac{U(x_i + x_{i+1} - x_i) - U(x_i)}{\Delta x}$$

y finalmente

$$U'(x) = \frac{U(x_{i+1}) - U(x_i)}{\Delta x} = \frac{U_{i+1} - U_i}{\Delta x} + O(\Delta x)$$

El término $O(\Delta x)$ corresponde al error de truncamiento de la aproximación de la derivada. En este caso el error es de orden Δx .

de igual forma obtenemos Euler hacia atrás

$$U'(x) = \frac{U(x_i) - U(x_i - \Delta x)}{\Delta x} = \frac{U_i - U_{i-1}}{\Delta x} + O(\Delta x)$$

y diferencias central

$$U'(x) = \frac{U(x_i + \Delta x) - U(x_i - \Delta x)}{2\Delta x} = \frac{U_{i+1} - U_{i-1}}{2\Delta x} + O(\Delta x)$$

Las series de Taylor han sido ampliamente utilizadas para estudiar el comportamiento de aproximaciones numéricas en ecuaciones diferenciales cuando una función y sus derivadas son funciones de un solo valor, finitas y continuas en x entonces la aproximación del método diferencias finitas se puede expresar de manera más formal por expansión de series de Taylor $U(x)$ siguiendo:

$$U(x + \Delta x) = U(x) + \frac{\Delta x}{1!} \frac{\partial U}{\partial x} + \frac{\Delta x^2}{2!} \frac{\partial^2 U}{\partial x^2} + \frac{\Delta x^3}{3!} \frac{\partial^3 U}{\partial x^3} + \dots$$

Diferencia central: Desarrollando la función mediante la serie de Taylor hasta el tercer orden, para $U(x + h)$ y $U(x - h)$

$$\begin{cases} U(x+h) = U(x) + \frac{\partial U(x)}{\partial x}h + \frac{1}{2!}\frac{\partial^2 U(x)}{\partial x^2}h^2 + \frac{1}{3!}\frac{\partial^3 U(x)}{\partial x^3}h^3 + \dots \\ U(x-h) = U(x) - \frac{\partial U(x)}{\partial x}h + \frac{1}{2!}\frac{\partial^2 U(x)}{\partial x^2}h^2 - \frac{1}{3!}\frac{\partial^3 U(x)}{\partial x^3}h^3 + \dots \end{cases}$$

sumando estas expresiones obtenemos

$$U(x+h) + U(x-h) = 2U(x) + h^2U''(x) + O(h^4) + \dots$$

donde $O(h)^4$ denota el termino que contiene el cuarto orden, de los siguiente obtenemos

$$U''(x) = \left(\frac{d^2u}{dx^2}\right)_{x=x} = \frac{1}{h^2} \{U(x+h) - 2U(x) + U(x-h)\}$$

$$U'(x) = \left(\frac{du}{dx}\right)_{x=x} = \frac{1}{2h} \{U(x+h) - U(x-h)\}$$

con error h^2

0.5.2. Derivadas en Dos Dimensiones

De forma análoga se construyen aproximaciones en diferencias finitas de primer y segundo orden en dos dimensiones. Usando el teorema de Taylor para funciones en dos variables x y y , es posible escribir de forma exacta para el punto x_i y y_j

$$f(x_i + \Delta x, y_j) = f(x_i, y_j) + \Delta x \frac{\partial f(x_i, y_j)}{\partial x} + \frac{\Delta x}{2} \frac{\partial^2 f(x_i + \Delta x, y_j)}{\partial x^2} + \dots$$

$$f(x_i, y_j + \Delta y) = f(x_i, y_j) + \Delta y \frac{\partial f(x_i, y_j)}{\partial y} + \frac{\Delta y}{2} \frac{\partial^2 f(x_i, y_j + \Delta y)}{\partial y^2} + \dots$$

Así, la aproximación en diferencias hacia delante de $\frac{\partial f}{\partial x}$ y $\frac{\partial f}{\partial y}$ es

$$\frac{\partial f(x_i, y_i)}{\partial x} = \frac{f(x_i + \Delta x, y_i) - f(x_i, y_i)}{\Delta x}$$

$$\frac{\partial f(x_i, y_i)}{\partial y} = \frac{f(x_i, y_i + \Delta y) - f(x_i, y_i)}{\Delta y}$$

o en su forma simplificada (para simplificar la notación, asociamos $\Delta x = h$ y $\Delta y = k$), tenemos

$$\frac{\partial f(x_i, y_i)}{\partial x} = \frac{f_{i+1,j} - f_{i,j}}{h}$$

$$\frac{\partial f(x_i, y_i)}{\partial y} = \frac{f_{i,j+1} - f_{i,j}}{k}$$

Así, la aproximación en diferencias hacia atrás de $\frac{\partial f}{\partial x}$ y $\frac{\partial f}{\partial y}$ es

$$\frac{\partial f(x_i, y_i)}{\partial x} = \frac{f(x_i, y_i) - f(x_i - \Delta x, y_i)}{\Delta x}$$

$$\frac{\partial f(x_i, y_i)}{\partial y} = \frac{f(x_i, y_i) - f(x_i, y_i - \Delta y)}{\Delta y}$$

o en su forma simplificada, tenemos

$$\frac{\partial f(x_i, y_i)}{\partial x} = \frac{f_{i,j} - f_{i-1,j}}{h}$$

$$\frac{\partial f(x_i, y_i)}{\partial y} = \frac{f_{i,j} - f_{i,j-1}}{k}$$

La aproximación en diferencias centradas de $\frac{\partial f}{\partial x}$ y $\frac{\partial f}{\partial y}$ es

$$\frac{\partial f(x_i, y_i)}{\partial x} = \frac{f(x_i + \Delta x, y_i) - f(x_i - \Delta x, y_i)}{2\Delta x}$$

$$\frac{\partial f(x_i, y_i)}{\partial y} = \frac{f(x_i, y_i + \Delta y) - f(x_i, y_i - \Delta y)}{2\Delta y}$$

o en su forma simplificada, tenemos

$$\frac{\partial f(x_i, y_i)}{\partial x} = \frac{f_{i+1,j} - f_{i-1,j}}{2h}$$

$$\frac{\partial f(x_i, y_i)}{\partial y} = \frac{f_{i,j+1} - f_{i,j-1}}{2k}$$

0.5.3. Malla intercalada 2D

La malla intercalada define algunos de los componentes de velocidad y tensión desplazados desde la ubicación de otros componentes por la mitad de la longitud de la rejilla en el espacio. Una de las características atractivas del enfoque con malla intercalada es que los operadores de diferencias están todos centrados naturalmente en el mismo punto en espacio y tiempo. Los sistemas no sólo están escalonados en una cuadrícula espacial sino también temporal, lo que significa que el campo de velocidad se puede actualizar independientemente del campo de tensión. Esto hace de un esquema eficiente y conciso.

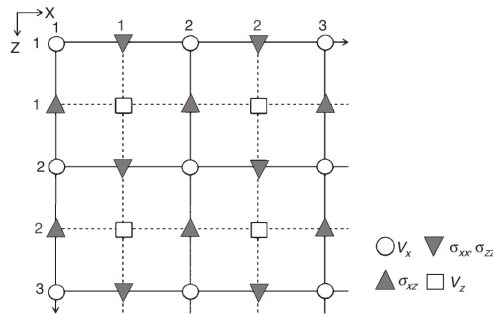


Figura 4: Ubicación de las componentes de la velocidad y tensión en una grilla escalonada

0.5.3.1. Aproximación Temporal y Espacial

El sistema de ecuaciones diferenciales son aproximados por medio de diferencias finitas de segundo orden en el tiempo y octavo orden en el espacio $O(\Delta t^2, \Delta x^8)$

Ampliamos la función U en $h = \pm \frac{\Delta x}{2}$

$$\begin{aligned}
 U(x + \frac{1}{2}\Delta x) &= U(x) + \frac{1}{2}\Delta x \frac{\partial U}{\partial x} + \frac{1}{8}\Delta x^2 \frac{\partial^2 U}{\partial x^2} + \frac{1}{48}\Delta x^3 \frac{\partial^3 U}{\partial x^3} + \frac{1}{384}\Delta x^4 \frac{\partial^4 U}{\partial x^4} + \frac{1}{3840}\Delta x^5 \frac{\partial^5 U}{\partial x^5} \\
 &\quad + \frac{1}{46080}\Delta x^6 \frac{\partial^6 U}{\partial x^6} + \frac{1}{645120}\Delta x^7 \frac{\partial^7 U}{\partial x^7} + O(\Delta x^8)
 \end{aligned}
 \tag{25}$$

$$\begin{aligned}
 U(x - \frac{1}{2}\Delta x) &= U(x) - \frac{1}{2}\Delta x \frac{\partial U}{\partial x} + \frac{1}{8}\Delta x^2 \frac{\partial^2 U}{\partial x^2} - \frac{1}{48}\Delta x^3 \frac{\partial^3 U}{\partial x^3} + \frac{1}{384}\Delta x^4 \frac{\partial^4 U}{\partial x^4} - \frac{1}{3840}\Delta x^5 \frac{\partial^5 U}{\partial x^5} \\
 + &\quad \frac{1}{46080}\Delta x^6 \frac{\partial^6 U}{\partial x^6} - \frac{1}{645120}\Delta x^7 \frac{\partial^7 U}{\partial x^7} + O(\Delta x^8)
 \end{aligned}
 \tag{26}$$

Ampliamos la función U en $h = \pm \frac{3}{2}$

$$\begin{aligned}
 U(x + \frac{3}{2}\Delta x) &= U(x) + \frac{3}{2}\Delta x \frac{\partial U}{\partial x} + \frac{9}{8}\Delta x^2 \frac{\partial^2 U}{\partial x^2} + \frac{9}{16}\Delta x^3 \frac{\partial^3 U}{\partial x^3} + \frac{27}{128}\Delta x^4 \frac{\partial^4 U}{\partial x^4} + \frac{81}{1280}\Delta x^5 \frac{\partial^5 U}{\partial x^5} \\
 + &\quad \frac{81}{5120}\Delta x^6 \frac{\partial^6 U}{\partial x^6} + \frac{243}{71680}\Delta x^7 \frac{\partial^7 U}{\partial x^7} + O(\Delta x^8)
 \end{aligned}
 \tag{27}$$

$$\begin{aligned}
 U(x - \frac{3}{2}\Delta x) &= U(x) - \frac{3}{2}\Delta x \frac{\partial U}{\partial x} + \frac{9}{8}\Delta x^2 \frac{\partial^2 U}{\partial x^2} - \frac{9}{16}\Delta x^3 \frac{\partial^3 U}{\partial x^3} + \frac{27}{128}\Delta x^4 \frac{\partial^4 U}{\partial x^4} - \frac{81}{1280}\Delta x^5 \frac{\partial^5 U}{\partial x^5} \\
 + &\quad \frac{81}{5120}\Delta x^6 \frac{\partial^6 U}{\partial x^6} - \frac{243}{71680}\Delta x^7 \frac{\partial^7 U}{\partial x^7} + O(\Delta x^8)
 \end{aligned}
 \tag{28}$$

Ampliamos la función U en $h = \pm \frac{5\Delta x}{2}$

$$\begin{aligned}
 U(x + \frac{5}{2}\Delta x) &= U(x) + \frac{5}{2}\Delta x \frac{\partial U}{\partial x} + \frac{25}{8}\Delta x^2 \frac{\partial^2 U}{\partial x^2} + \frac{125}{48}\Delta x^3 \frac{\partial^3 U}{\partial x^3} + \frac{625}{384}\Delta x^4 \frac{\partial^4 U}{\partial x^4} + \frac{625}{768}\Delta x^5 \frac{\partial^5 U}{\partial x^5} \\
 + &\quad \frac{3125}{9216}\Delta x^6 \frac{\partial^6 U}{\partial x^6} + \frac{242203001}{2000000000}\Delta x^7 \frac{\partial^7 U}{\partial x^7} + O(\Delta x^8)
 \end{aligned}$$

(29)

$$\begin{aligned}
U(x - \frac{5}{2}\Delta x) = & U(x) - \frac{5}{2}\Delta x \frac{\partial U}{\partial x} + \frac{25}{8}\Delta x^2 \frac{\partial^2 U}{\partial x^2} - \frac{125}{48}\Delta x^3 \frac{\partial^3 U}{\partial x^3} + \frac{625}{384}\Delta x^4 \frac{\partial^4 U}{\partial x^4} - \frac{625}{768}\Delta x^5 \frac{\partial^5 U}{\partial x^5} \\
& + \frac{3125}{9216}\Delta x^6 \frac{\partial^6 U}{\partial x^6} - \frac{242203001}{2000000000}\Delta x^7 \frac{\partial^7 U}{\partial x^7} + O(\Delta x^8)
\end{aligned}$$

(30)

Ampliamos la función U en $h = \pm \frac{7\Delta x}{2}$

$$\begin{aligned}
U(x + \frac{7}{2}\Delta x) = & U(x) + \frac{7}{2}\Delta x \frac{\partial U}{\partial x} + \frac{49}{8}\Delta x^2 \frac{\partial^2 U}{\partial x^2} + \frac{343}{48}\Delta x^3 \frac{\partial^3 U}{\partial x^3} + \frac{2401}{384}\Delta x^4 \frac{\partial^4 U}{\partial x^4} + 4 \frac{376822917}{1000000000}\Delta x^5 \frac{\partial^5 U}{\partial x^5} \\
& + 2 \frac{553146701}{1000000000}\Delta x^6 \frac{\partial^6 U}{\partial x^6} + 1 \frac{276573351}{1000000000}\Delta x^7 \frac{\partial^7 U}{\partial x^7} + O(\Delta x^8)
\end{aligned}$$

(31)

$$\begin{aligned}
U(x - \frac{7}{2}\Delta x) = & U(x) - \frac{7}{2}\Delta x \frac{\partial U}{\partial x} + \frac{49}{8}\Delta x^2 \frac{\partial^2 U}{\partial x^2} - \frac{343}{48}\Delta x^3 \frac{\partial^3 U}{\partial x^3} + \frac{2401}{384}\Delta x^4 \frac{\partial^4 U}{\partial x^4} + O(\Delta x^5) - 4 \frac{376822917}{1000000000}\Delta x^5 \frac{\partial^5 U}{\partial x^5} \\
& + 2 \frac{553146701}{1000000000}\Delta x^6 \frac{\partial^6 U}{\partial x^6} - 1 \frac{276573351}{1000000000}\Delta x^7 \frac{\partial^7 U}{\partial x^7} + O(\Delta x^8)
\end{aligned}$$

(32)

usando

$$D1 = U(x + \frac{\Delta x}{2}) - U(x - \frac{\Delta x}{2}) = \Delta x \frac{\partial U}{\partial x} + \frac{1}{24}\Delta x^3 \frac{\partial^3 U}{\partial x^3} + \frac{1}{1920}\Delta x^5 \frac{\partial^5 U}{\partial x^5} + \frac{1}{322560}\Delta x^7 \frac{\partial^7 U}{\partial x^7} + O(\Delta x^8)$$

$$D2 = U(x + \frac{3}{2}\Delta x) - U(x - \frac{3}{2}\Delta x) = 3\Delta x \frac{\partial U}{\partial x} + \frac{9}{8}\Delta x^3 \frac{\partial^3 U}{\partial x^3} + \frac{81}{640}\Delta x^5 \frac{\partial^5 U}{\partial x^5} + \frac{243}{35840}\Delta x^7 \frac{\partial^7 U}{\partial x^7} + O(\Delta x^8)$$

$$D3 = U(x + \frac{5}{2}\Delta x) - U(x - \frac{5}{2}\Delta x) = 5\Delta x \frac{\partial U}{\partial x} + \frac{125}{24}\Delta x^3 \frac{\partial^3 U}{\partial x^3} + \frac{625}{384}\Delta x^5 \frac{\partial^5 U}{\partial x^5} + \frac{242203001}{1000000000}\Delta x^7 \frac{\partial^7 U}{\partial x^7} + O(\Delta x^8)$$

$$D4 = U(x + \frac{7}{2}\Delta x) - U(x - \frac{7}{2}\Delta x) = 7\Delta x \frac{\partial U}{\partial x} + \frac{343}{24}\Delta x^3 \frac{\partial^3 U}{\partial x^3} + 8 \frac{376822917}{500000000}\Delta x^5 \frac{\partial^5 U}{\partial x^5} + 2 \frac{276573351}{500000000}\Delta x^7 \frac{\partial^7 U}{\partial x^7} + O(\Delta x^8)$$

en su forma general la anterior aproximación lo expresamos siguiendo

$$\frac{\partial f}{\partial x} = \frac{1}{\Delta x} \sum_{n=1}^N C_n^{(N)} \left\{ f \left[x + \frac{\Delta x}{2}(2n-1) \right] - f \left[x - \frac{\Delta x}{2}(2n-1) \right] \right\} + O(\Delta x^{2N})$$

dónde los coeficientes de aproximación son $C_n^{(N)}$ se pueden obtenerse mediante la siguiente matriz

$$\begin{pmatrix} 1 & \frac{1}{24} & \frac{1}{1920} & \frac{1}{322560} \\ 3 & \frac{8}{9} & \frac{640}{81} & \frac{35840}{243} \\ 5 & \frac{125}{24} & \frac{625}{384} & \frac{242203001}{100000000} \\ 7 & \frac{343}{24} & \frac{376822917}{500000000} & \frac{276573351}{500000000} \end{pmatrix} \begin{bmatrix} C_1^{(N)} \\ C_2^{(N)} \\ C_3^{(N)} \\ C_4^{(N)} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (33)$$

Resolviendo el sistema para $2N=8$ de la forma $A \setminus B$ obtenemos los coeficientes $C_n^{(N)}$ de aproximación para un esquema de octavo orden

$$C_1^{(4)} = 1,196289; C_2^{(4)} = -0,0797526; C_3^{(4)} = 0,009570313; C_4^{(4)} = -0,0006975447$$

Finalmente diferencias finitas de 8 orden en el espacio $0(\Delta x^8)$

$$\frac{\partial U}{\partial x} + O(\Delta x^8) \simeq \frac{C_1 [U(x + \frac{\Delta x}{2}) - U(x - \frac{\Delta x}{2})] + C_2 [U(x + \frac{3\Delta x}{2}) - U(x - \frac{3\Delta x}{2})] + C_3 [U(x + \frac{5\Delta x}{2}) - U(x - \frac{5\Delta x}{2})] + C_4 [U(x + \frac{7\Delta x}{2}) - U(x - \frac{7\Delta x}{2})]}{\Delta x} \quad (34)$$

D_t : Derivada parcial de primer orden con respecto del tiempo en diferencias finitas centradas de segundo orden $O(\delta h^2)$ (Virieux, 1986). Para las componentes de la velocidad

$$D_t [v_x]_{i,k}^n \simeq \frac{[v_x]_{i,k}^{n+1/2} - [v_x]_{i,k}^{n-1/2}}{\delta t} \quad (35)$$

$$D_t [v_z]_{i,k}^n \simeq \frac{[v_z]_{i,k}^{n+1/2} - [v_z]_{i,k}^{n-1/2}}{\delta t} \quad (36)$$

Para las componentes de las tensiones:

$$D_t [\tau_{xx}]_{i,k}^n \simeq \frac{[\tau_{xx}]_{i,k}^{n+1/2} - [\tau_{xx}]_{i,k}^{n-1/2}}{\delta t} \quad (37)$$

$$D_t [\tau_{zz}]_{i,k}^n \simeq \frac{[\tau_{zz}]_{i,k}^{n+1/2} - [\tau_{zz}]_{i,k}^{n-1/2}}{\delta t} \quad (38)$$

$$D_t [\tau_{xz}]_{i,k}^n \simeq \frac{[\tau_{xz}]_{i+1/2,k+1/2}^{n+1/2} - [\tau_{xz}]_{i+1/2,k+1/2}^{n-1/2}}{\delta t} \quad (39)$$

D_x, D_z : Derivada parcial de primer orden con respecto de la posición x, z en diferencias finitas centradas de octavo orden $O(\delta h^8)$

Para las componentes de la velocidad

$$D_x [v_x]_{i,k}^n \simeq \frac{1}{\delta x} \left[C_1 \left([v_x]_{i+1/2,k}^n - [v_x]_{i-1/2,k}^n \right) + C_2 \left([v_x]_{i+3/2,k}^n - [v_x]_{i-3/2,k}^n \right) + C_3 \left([v_x]_{i+5/2,k}^n - [v_x]_{i-5/2,k}^n \right) + C_4 \left([v_x]_{i+7/2,k}^n - [v_x]_{i-7/2,k}^n \right) \right] \quad (40)$$

$$D_z [v_z]_{i,k}^n \simeq \frac{1}{\delta z} \left[C_1 \left([v_z]_{i,k+1/2}^n - [v_z]_{i,k-1/2}^n \right) + C_2 \left([v_z]_{i,k+3/2}^n - [v_z]_{i,k-3/2}^n \right) + C_3 \left([v_z]_{i,k+5/2}^n - [v_z]_{i,k-5/2}^n \right) + C_4 \left([v_z]_{i,k+7/2}^n - [v_z]_{i,k-7/2}^n \right) \right] \quad (41)$$

Para las componentes de las tensiones:

$$D_x[\tau_{xx}]_{i,k}^n \simeq \frac{1}{\delta x} \left[C1 \left([\tau_{xx}]_{i+1/2,k}^n - [\tau_{xx}]_{i-1/2,k}^n \right) + C2 \left([\tau_{xx}]_{i+3/2,k}^n - [\tau_{xx}]_{i-3/2,k}^n \right) \right. \\ \left. + C3 \left([\tau_{xx}]_{i+5/2,k}^n - [\tau_{xx}]_{i-5/2,k}^n \right) + C4 \left([v_x]_{i+7/2,k}^n - [v_x]_{i-7/2,k}^n \right) \right] \quad (42)$$

$$D_z[\tau_{zz}]_{i,k}^n \simeq \frac{1}{\delta z} \left[C1 \left([\tau_{zz}]_{i,k+1/2}^n - [\tau_{zz}]_{i,k-1/2}^n \right) + C2 \left([\tau_{zz}]_{i,k+3/2}^n - [\tau_{zz}]_{i,k-3/2}^n \right) \right. \\ \left. + C3 \left([\tau_{zz}]_{i,k+5/2}^n - [\tau_{zz}]_{i,k-5/2}^n \right) + C4 \left([\tau_{zz}]_{i,k+7/2}^n - [\tau_{zz}]_{i,k-7/2}^n \right) \right] \quad (43)$$

$$D_x[\tau_{xz}]_{i,k}^n \simeq \frac{1}{\delta x} \left[C1 \left([\tau_{xz}]_{i+1/2,k+1/2}^n - [\tau_{xz}]_{i-1/2,k-1/2}^n \right) + C2 \left([\tau_{xz}]_{i+3/2,k+3/2}^n - [\tau_{xz}]_{i-3/2,k-3/2}^n \right) \right. \\ \left. + C3 \left([\tau_{xz}]_{i+5/2,k+5/2}^n - [\tau_{xz}]_{i-5/2,k-5/2}^n \right) + C4 \left([\tau_{xz}]_{i+7/2,k+7/2}^n - [\tau_{xz}]_{i-7/2,k-7/2}^n \right) \right] \quad (44)$$

0.5.4. Solución Numérica para un Medio isótropo

Para solucionar las ecuaciones para un medio con simetría Isótropa, utilizamos las aproximaciones obtenidas en, definimos $\delta h = \delta x = \delta z$, donde $b_{i,k} = \frac{1}{\rho(x, z)}$ como el inverso de la densidad. obtenemos:

$$[v_x]_{i,k}^n = [v_x]_{i,k}^n + b_{i,k} \delta t \left(D_x[\tau_{xx}]_{i+1/2,k}^n + D_z[\tau_{xz}]_{i,k+1/2}^n \right) \\ [v_z]_{i+1/2,k+1/2}^{n+1} = [v_z]_{i,k+1/2}^{n+1} + b_{i,k} \delta t \left(D_x[\tau_{xz}]_{i,k}^n + D_z[\tau_{zz}]_{i,k}^n \right) \quad (45)$$

de las velocidades

$$[\tau_{xx}]_{i+1/2,k}^{n+1} = [\tau_{xx}]_{i+1/2,k}^{n+1} + \delta t \left((C11)_{i,k} D_x[v_x]_{i,k}^{n+1/2} + (C12)_{i,k} D_z[v_z]_{i,k}^{n+1/2} \right) \\ [\tau_{zz}]_{i+1/2,k}^{n+1} = [\tau_{zz}]_{i+1/2,k}^{n+1} + \delta t \left((C12)_{i,k} D_x[v_x]_{i,k}^{n+1/2} + (C11)_{i,k} D_z[v_z]_{i,k}^{n+1/2} \right) \\ [\tau_{xz}]_{i+1/2,k}^{n+1} = [\tau_{xz}]_{i+1/2,k}^{n+1} + \delta t (C35)_{i,k} \left(D_z[v_x]_{i,k}^{n+1/2} + D_x[v_z]_{i,k}^{n+1/2} \right) \quad (46)$$

0.5.5. Solución Numérica para un Medio VTI

Para solucionar las ecuaciones para un medio con eje de simetría Isótropia verticalmente, utilizamos las aproximaciones obtenidas en, definimos nuevamente $\delta h = \delta x = \delta z$, donde $b_{i,k} = \frac{1}{\rho(x, z)}$ como el inverso de la densidad. obtenemos:

$$[v_x]_{i,k}^n = [v_x]_{i,k}^n + b_{i,k} \delta t \left(D_x[\tau_{xx}]_{i+1/2,k}^n + D_z[\tau_{xz}]_{i,k+1/2}^n \right) \\ [v_z]_{i+1/2,k+1/2}^{n+1} = [v_z]_{i,k+1/2}^{n+1} + b_{i,k} \delta t \left(D_x[\tau_{xz}]_{i,k}^n + D_z[\tau_{zz}]_{i,k}^n \right) \quad (47)$$

de las velocidades

$$\begin{aligned}
[\tau_{xx}]_{i+1/2,k}^{n+1} &= [\tau_{xx}]_{i+1/2,k}^{n+1} + \delta t((C_{11})_{i,k}D_x[v_x]_{i,k}^{n+1/2} + (C_{13})_{i,k}D_z[v_z]_{i,k}^{n+1/2}) \\
[\tau_{zz}]_{i+1/2,k}^{n+1} &= [\tau_{zz}]_{i+1/2,k}^{n+1} + \delta t((C_{13})_{i,k}D_x[v_x]_{i,k}^{n+1/2} + (C_{33})_{i,k}D_z[v_z]_{i,k}^{n+1/2}) \\
[\tau_{xz}]_{i+1/2,k}^{n+1} &= [\tau_{xz}]_{i+1/2,k}^{n+1} + \delta t(C_{44})_{i,k}(D_z[v_x]_{i,k}^{n+1/2} + D_x[v_z]_{i,k}^{n+1/2})
\end{aligned} \tag{48}$$

0.5.6. Solución Numérica para un Medio TTI

Para solucionar las ecuaciones para un medio con eje de simetría Isotropía verticalmente inclinada, utilizamos las aproximaciones obtenidas en, definimos nuevamente $\delta h = \delta x = \delta z$, donde $b_{i,k} = \frac{1}{\rho(x,z)}$ como el inverso de la densidad. obtenemos:

$$\begin{aligned}
[v_x]_{i,k}^n &= [v_x]_{i,k}^n + b_{i,k}\delta t \left(D_x[\tau_{xx}]_{i+1/2,k}^n + D_z[\tau_{xz}]_{i,k+1/2}^n \right) \\
[v_z]_{i+1/2,k+1/2}^{n+1} &= [v_z]_{i,k+1/2}^{n+1} + b_{i,k}\delta t \left(D_x[\tau_{xz}]_{i,k}^n + D_z[\tau_{zz}]_{i,k}^n \right)
\end{aligned} \tag{49}$$

de las velocidades

$$\begin{aligned}
[\tau_{xx}]_{i+1/2,k}^{n+1} &= [\tau_{xx}]_{i+1/2,k}^{n+1} + \delta t((C_{11})_{i,k}D_x[v_x]_{i,k}^{n+1/2} + (C_{13})_{i,k}D_z[v_z]_{i,k}^{n+1/2}) + (C_{15})_{i,k}(D_z[v_x]_{i,k}^{n+1/2} + D_x[v_z]_{i,k}^{n+1/2}) \\
[\tau_{zz}]_{i+1/2,k}^{n+1} &= [\tau_{zz}]_{i+1/2,k}^{n+1} + \delta t((C_{13})_{i,k}D_x[v_x]_{i,k}^{n+1/2} + (C_{33})_{i,k}D_z[v_z]_{i,k}^{n+1/2}) + (C_{15})_{i,k}(D_z[v_x]_{i,k}^{n+1/2} + D_x[v_z]_{i,k}^{n+1/2}) \\
[\tau_{xz}]_{i+1/2,k}^{n+1} &= [\tau_{xz}]_{i+1/2,k}^{n+1} + \delta t((C_{13})_{i,k}D_x[v_x]_{i,k}^{n+1/2} + (C_{35})_{i,k}D_z[v_z]_{i,k}^{n+1/2}) + (C_{55})_{i,k}(D_z[v_x]_{i,k}^{n+1/2} + D_x[v_z]_{i,k}^{n+1/2})
\end{aligned} \tag{50}$$

0.5.7. Condición de Estabilidad

Cuando se utilizan esquemas explícitos para la solución numérica del número Courant (Courant et al., 1967) da una condición para la convergencia. El número de Courant se utiliza para restringir el paso del tiempo en simulaciones por computadora. Por ejemplo, si una onda está cruzando una distancia discreta de la rejilla (Δx), entonces el paso del tiempo debe ser menor que el tiempo necesario para que la onda viaje a un punto de rejilla adyacente, de lo contrario la simulación producirá resultados incorrectos.(20)

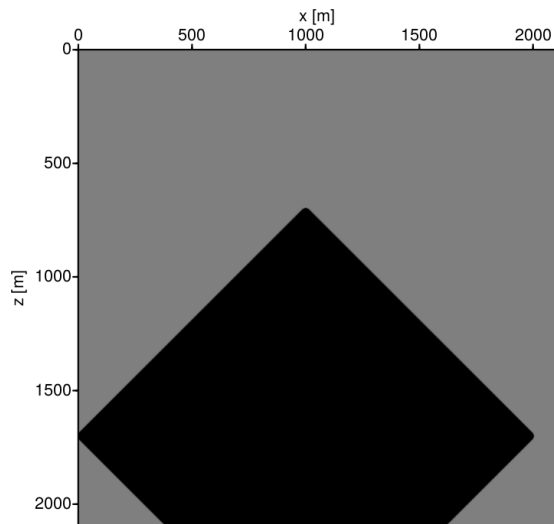


Figura 5: Ejemplo de inestabilidad

En general, el criterio de estabilidad para la grilla escalonada establece

$$\Delta t \max(v) \sqrt{\frac{1}{\Delta x^2} + \frac{1}{\Delta z^2}} \leq \frac{1}{\sum_{i=1}^N |c_i|}$$

se define $C = \frac{1}{\sum_{i=1}^N |c_i|}$ como el inverso de la sumatoria tomando el valor absoluto de los coeficientes para una aproximación de 8avo-orden.

$$N = 4, C = 0,7774$$

el criterio de estabilidad se reduce a:

$$\Delta t < \frac{0,5497048 * dh}{Vp_{max}} \quad (51)$$

Donde Δt es el intervalo de tiempo, Vp_{max} es la máxima velocidad de la onda P y dh es el espaciado de la grilla en dirección x y z asumiendo que ambas son iguales.

0.5.8. Condición de dispersión

Además de las soluciones inestables también puede ocurrir la dispersión del campo de ondas. La frecuencia máxima que puede ser exacta por modelado está relacionada con la mínima de toda la frecuencia f calculada para cada celda unitaria.

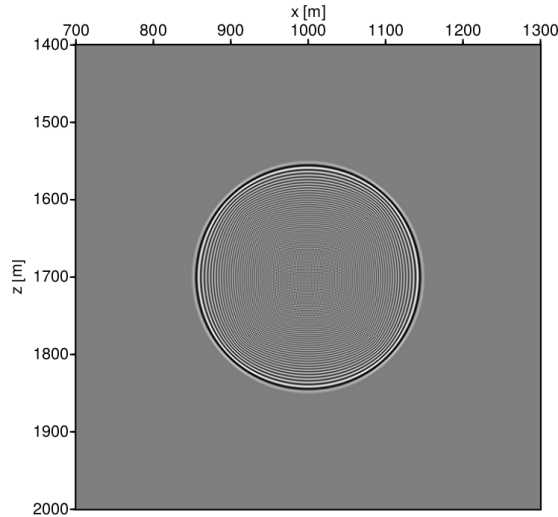


Figura 6: Ejemplo de dispersión numérica

Para reducir la dispersión numérica para un esquema de $O(\Delta t^2, \Delta x^8)$ la relación de dispersión es:

$$\max(\Delta x, \Delta z) < \frac{\min(v)}{3,5 f_{max}} \quad (52)$$

donde se define 3.5 puntos por longitud de onda y $\min(v)$ como el mínimo de la velocidad entre la onda P y S.

0.5.9. Condición de contorno

De acuerdo a las condiciones de contorno tomamos la que fue propuesta por (18), el principio consiste en atenuar la reflexión exponencialmente con un límite artificial multiplicando por factor $d(u)$ menor que 1.

Comúnmente el factor es:

$$d(u) = \exp(-[0,015 * (nb - i)]^2), u = x, z(i\Delta x \text{ o } i\Delta z) \quad (53)$$

donde nb es el tamaño del límite artificial en cada lado del modelo.

Generalmente, elegimos $nb = 20 \sim 30$.

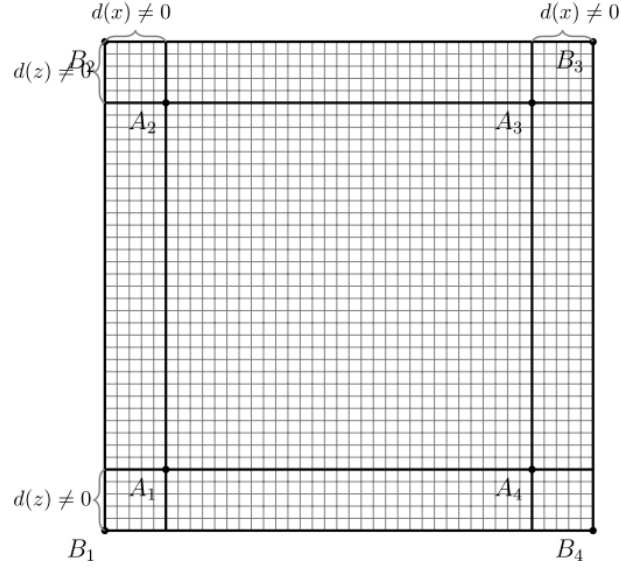


Figura 7: Esquema tomado de (25) se muestra un diagrama esquemático de las condiciones de contorno propuesta por (18) . Donde A_1, A_2, A_3 y A_4 es contorno del modelo original y B_1, B_2, B_3 y B_4 es el área de contorno artificial que atenúa las reflexión.

0.5.10. Fuente

La fuente de excitación se utiliza para inicializar la propagación de la onda, se implementa simplemente añadiendo una función $S(t)$ en el tiempo aplicada a las tensiones como:

$$\tau_{xx}(ix, iz) = S(t), \quad \text{o} \quad \tau_{zz}(ix, iz) = S(t)$$

0.5.10.1. Ondicula

Las Tres funciones que se utilizan comúnmente como fuente de excitación son:
La Función gaussiana

$$g(t) = \exp(-\alpha t^2)$$

la primera derivada de la función gaussiana

$$g(t) = -2\alpha t \exp(-\alpha t^2)$$

y la segunda derivada de la función gaussiana (la Ricker wavelet),

$$g(t) = (4\alpha^2 t^2 - 2\alpha) \exp(-\alpha t^2)$$

Esta última función se puede escribir como función en el dominio de la frecuencia f_d , es decir

$$g(t) = (1 - 2\pi^2 f_d^2 t^2) \exp(-\pi f_d t^2)$$

Capítulo 4

0.6. Computación de alto rendimiento

El objetivo principal de la computación de alto desempeño es mejorar el tiempo de cálculo haciendo uso de múltiples procesadores y computadoras permitiendo así realizar tareas complejas simultáneamente con alto rendimiento, la computación paralela se puede definir como aquella en la que muchas operaciones son realizadas simultáneamente, bajo el principio de que grandes problemas pueden ser divididos en pequeñas partes, cada una de las cuales se resuelve concurrentemente. Sin embargo, desde el punto de vista del programador surge el interrogante de cómo desarrollar los algoritmos que realicen los cálculos de dicha forma sobre las máquinas. Por tanto la computación paralela involucra habitualmente dos áreas de la tecnología informática: La arquitectura de computadores, cuyo enfoque se centra en cómo soportar el paralelismo a nivel de hardware. La programación en paralelo, la cual se enfoca en resolver el problema de la concurrencia para poder utilizar completamente el poder del cómputo del hardware. Es decir se centra en el desarrollo apropiado del software.

Como es bien conocido, la arquitectura básica de los sistemas de cómputo consta de cuatro partes fundamentales: La CPU o unidad central de procesamiento (Unidad de control y unidad aritmética lógica), la memoria (de datos e instrucciones), y los periféricos de entrada y salida (Chengm Grossman, McKercher, 2014). El componente clave en la computación de alto rendimiento (HPC por sus siglas en inglés : High-performance Computing), en la unidad central de proceso, habitualmente llamada core. En los primeros años de la computación moderna, las máquinas eran monoprocesador, es decir solo poseían un core o chip. Hoy en la tendencia en el diseño de chips es integrar varios cores sobre un solo procesador, conocido como tecnología multicore, para soportar el paralelismo a nivel de arquitectura. Por lo tanto, el problema de la programación en paralelo puede ser visto como el diseño de algoritmos que realice el procesamiento de un problema sobre los cores disponibles de tal forma que obtengan una ejecución en paralelo.

0.6.1. Paralelismo

Existen dos tipos fundamentales de paralelismo, paralelismo de tareas y paralelismo de datos, El paralelismo de tareas se utiliza cuando existe una gran cantidad de tareas o funciones que pueden ser ejecutadas en forma independiente por extensos periodos en forma paralela. El paralelismo de tareas se encarga de distribuir estas funciones entre los múltiples núcleos o cores. Por otro lado, el paralelismo de datos aplica cuando se tiene gran cantidad de datos que pueden ser procesados al mismo tiempo. El paralelismo de datos se enfoca en distribuir los datos entre múltiples cores.

0.6.2. Arquitecturas Paralelas

Aunque existen varias formas de clasificar las arquitecturas de computadoras, un esquema de clasificación ampliamente usando en la bibliografía es la taxonomía de Flynn, en el cual el autor considera dos aspectos: El flujo de instrucciones y el flujo de datos (Picolli Fabiana, 2011)

- SISD- Single Instruction Single Data: Son las computadoras con un único flujo de instrucciones y un único flujo de datos. Se refiere a la computación tradicional basada en arquitectura serial, con una sola CPU.
- MISD- Multiple instruction Single Data: Es una arquitectura poco común en la cual se tiene múltiples flujo de instrucciones y un único flujo de datos
- MIMD- Multiple Instruction Multiple Data: Se refiere a un tipo de arquitectura en la cual las computadoras tiene múltiples cores que operan sobre múltiples flujo de datos. Son todas unidades de procesamiento independientes con sus propios flujos de instrucciones y de datos. Se pueden encontrar dos tipos de MIMD, los multiprocesadores arquitecturas con memoria compartida, y las multicomputadoras , arquitectura con memoria distribuida.

0.6.3. Paradigmas de programación orientados a la arquitectura

0.6.3.1. Paradigma de memoria compartida

Aplica en el caso de programación de un multiprocesador. El código y los datos necesarios para su ejecución son almacenados en la memoria compartida. De esta forma, cada unidad de procesamiento accede a la memoria para ejecutar el código, obtener los datos para su ejecución o guardar los resultados obtenidos. Los programas desarrollados siguiente este modelo se basan en constructores provistos por un lenguaje de programación. Fork y OpenMP son ejemplos de herramientas utilizadas para trabajar sobre un paradigma. Recientemente, con el auge de las unidades de procesamiento gráfico GPU, surgen herramientas como CUDA, que ha sido una de las mas difundidas por su flexibilidad, rápido aprendizaje y posibilidad de programar aplicación de propósito general.

0.6.3.2. Paradigma de Paso de Mensajes

Se aplica en el caso de programas sobre una multicomputadora, lo que requiere la división del problema en partes o tareas, cada una de las cuales pueden ejecutarse en una computadora o procesador diferente. La comunicación entre la distintas tareas se realizan a través del paso de mensajes, utilizando algún tipo de librería que ofrezca las herramientas necesarias. En este paradigma el paso de mensajes se especifica dentro del código que resuelve el problema y los datos no son compartidos si no que son propios de cada tareas. Para este tipo de problemas se utilizan librerías que suministran las herramientas necesarias para la comunicación entre procesos, como puede ser MPI.

0.6.4. Computación acelerada con GPU

Una unidad de procesamiento gráfico, mas conocida como GPU por su siglas en ingles (Graphics Processor Unit) resulta ser un coprocesador dedicado al procesamiento de gráficos o a la ejecución de operaciones de como flotante, las GPU ofrece procesadores de varios núcleos que ofrecen alto rendimiento, de tal forma que se reduzca el trabajo que debe realizar el procesador central en aplicaciones con alto costo computacional tal como vídeo juegos, programas de diseño

o modelado 3D, entre otros. Dicha tecnología al ofrecer grandes prestaciones en trabajos que implicaban gran carga de procesamiento, fue adoptada para el procesamiento de aplicaciones que requieren alto coste, pero alejadas de los gráficos esto es la computación científica para el modelado de grandes volúmenes de datos, solución numérica modelos a través de ecuaciones diferenciales, problemas de aprendizaje profundo, entre otras.

Se ha comprobado que el cálculo mediante una GPU acelerar el tiempo de ejecución en comparación con CPU tradicional. Una CPU se basa en un pequeño número de núcleos, con velocidad de ejecución más rápida realizada por cada núcleo. Una GPU depende en un gran número de núcleos con una velocidad de ejecución más lenta. Sin embargo, si todos los núcleos capaces de realizar un cálculo simultáneamente, el uso de un alto número de núcleos será una ventaja. Este concepto ha sido conocido como cálculo paralelo.

0.6.5. Procesamiento paralelo con CUDA

Como hemos visto anteriormente existen múltiples modelos para mejorar el rendimiento de los programas, la tecnología que más repunta en la actualidad es la basada en GPU (unidad de procesamiento gráfico) el cual ofrece un incremento extraordinario del rendimiento del sistema para problemas con alto grado de paralelismo.

Nvidia ha desarrollado un conjunto de herramientas para el realizar cálculos en paralelo haciendo uso de las GPU esta tecnología es más conocida como CUDA que esta conformada por un conjunto de librerías, APIS y otras herramientas que pueden facilitar el desarrollo de programas con paralelismo en GPU: Py CUDA, Anaconda, CUDA toolkit, Fortran CUDA, OpenACC son algunas de ellas. La plataforma de cálculo paralelo CUDA proporciona extensiones de C y C++ que permiten implementar el paralelismo en el procesamiento de tareas y datos con diferentes niveles de granularidad. Cada desarrollador puede implementar aplicaciones paralelas mediante diferentes lenguajes de alto nivel como C, C++ y Fortran o mediante estándares abiertos como las directivas de OpenACC.

En la actualidad la plataforma CUDA se utiliza en miles de aplicaciones aceleradas en la GPU y en miles de artículos de investigación publicados, entre las aplicaciones que aprovechan al máximo esta tecnología pasan por distintos campos como:

- El procesamiento de vídeo e imágenes
- La biología y la química computacional
- La simulación de la dinámica de fluidos.
- El análisis sísmico.
- La reconstrucción de imágenes, entre otros.

0.6.6. Modelo CUDA

Tradicionalmente, una CPU contiene algunos núcleos (unidades lógicas aritméticas ALU) comparten su memoria en una cache común y una memoria virtual como (Dram) las GPU funcionan un poco de forma diferente en el sentido que una GPU consta de miles de unidades lógicas aritméticas, estas unidades lógicas aritméticas comparten la misma memoria y no se puede acceder a la memoria directamente de un bloque a otro. De la misma manera que funcionan una memoria DRAM, las GPU funcionan con una cantidad de memoria razonable grande de memoria global. Dependiendo de las necesidades de usuario, las ALUS pueden ser ordenadas en

1,2,3 dimensiones, aparte de la memoria global una GPU tiene memoria constante y memoria de textura que tiene un efecto de rendimiento si se usa apropiadamente.

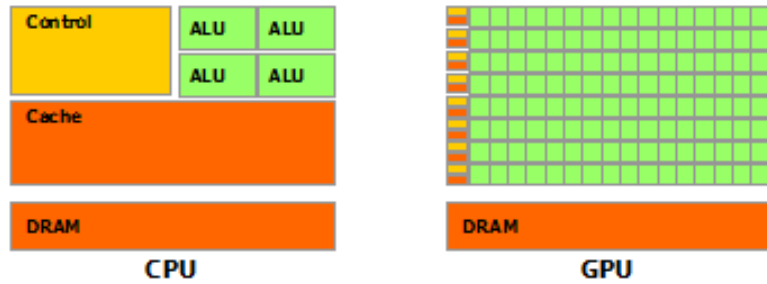


Figura 8: GPU,CPU

0.6.6.1. CUDA Threads

La Unidad básica de operación en el device o GPU es el thread o hilos el cual son subprocesos extremadamente livianos que tienen un id que se utiliza para computar direcciones de memoria y tomar direcciones de control.

La jerarquía en la cual están los threads organizados es la siguiente:

- Los threads están organizados en bloques.
 - Estos están organizados dentro de bloques y se distribuyen en forma de matriz de la cual puede tomar 1, 2 ó 3 dimensiones, cada hilo toma un índice para ser para ser identificado mediante threadIdx.
- Los bloques están organizados en Grid (malla de bloques)
 - Los bloques se agrupan dentro de un grid que de igual forma a la distribución de los threads se distribuyen en forma de matriz que pueden ser direccionado por 1D,2D y 3D dimensiones (3D desde Fermi) mediante el índice blockIdx. Asimismo, la dimensión del bloque también se puede obtener desde dentro del kernel mediante blockDim.
- Un grid solo puede ejecutar un kernel

lo anterior se resumen en la siguiente figura

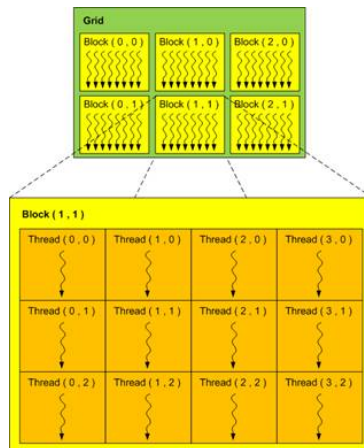


Figura 9: Jerarquía de Threads

CUDA Kernel se ejecuta mediante un array de Threads. Todos los Threads ejecutan el mismo código. Cada Thread tiene un ID para distinguir entre cada otro que se usa para direccionar la memoria y tomar las decisiones de control aunque las instrucciones del kernel son lo mismo.

- Los grids pueden ser de 1 o 2 dimensiones.
- Los hilos de un mismo bloque pueden cooperar entre sí mediante el uso de memoria compartida. sincronizando su ejecución para coordinar los accesos a memoria y lo contrario Threads en diferentes bloques no pueden cooperar directamente.

Básicamente el el flujo de procesamiento de un programa CUDA se resume en lo siguiente

- Copiar los datos del host al device
- Ejecutar al kernel que opera los datos previamente almacenados en la GPU
- Trae de vuelta los datos del device al host previamente operados

El modelo de programación de CUDA asume que los CUDA threads se ejecutan en un device que actúa como coprocesador de un host que ejecuta un programa. También asume que host y device poseen su propia DRAM, host memory y device memory. Cuda proporciona instrucciones para reservar, liberar, copiar memoria en la memoria del device, así como transferir datos entre el host y el device

Capítulo 5

0.7. Experimentos numéricos

0.7.1. Implementación Computacional

El algoritmo prácticamente obtiene los C_{ij} para un medio VTI o TTI, se crea la fuente que va ser aplicada y se controla que los parámetros dados cumplan el criterio de estabilidad y dispersión, finalmente se da inicio a la iteración en el paso del tiempo actualizando las velocidades y tensiones. El siguiente diagrama muestra el flujo del algoritmo.

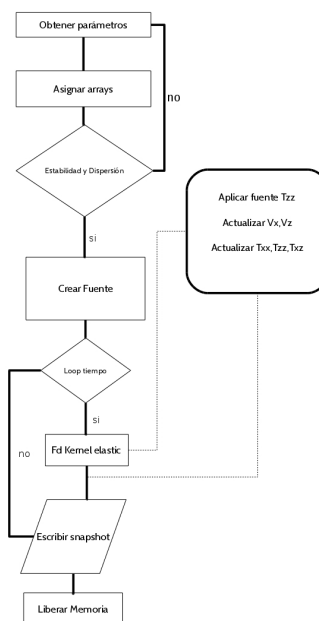


Figura 10: Diagrama de flujo del algoritmo implementado

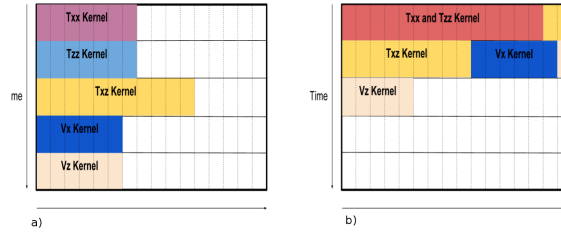


Figura 11: Implementacion computacional de cada una de las componentes principales a) CPU b) GPU

De acuerdo al tipo de implementación realizada (CPU, GPU) estas permiten que las componentes se calculen o no de forma concurrente de tal manera que se optimicen al máximo los recursos computacionales..

Un kernel es una subrutina que debe ser ejecutada, hemos dividido el cálculo en un conjunto natural de dos núcleos, uno para las velocidades y otro para las tensiones, Se muestran a continuación el código en C y CUDA C de cada uno de los Kernels implementados en CPU y GPU que dan solución a la ecuación elastodinámica con eje de simetría tipo VTI,TTI.

0.7.1.1. Kernel en CPU que permite calcular las velocidades.

```

/* calcula vx para todos los puntos de la grilla , excepto puntos ficticios
*/

    for (ix=3; ix<Nx-3; ix++) {
        for (iz=3; iz<Nz-3; iz++) {
Dx =    c1*(Txx(ix+1,iz)- Txx(ix , iz))  + c2*(Txx(ix+2,iz)-Txx(ix-1,iz)) + c3
        *(Txx(ix+3,iz)-Txx(ix-2,iz)) + c4*(Txx(ix+4,iz)-Txx(ix-3,iz));
Dz =    c1*(Txz(ix , iz+1)- Txz(ix , iz) ) + c2*(Txz(ix , iz+2)-Txz(ix , iz-1)) + c3
        *(Txz(ix , iz+3)-Txz(ix , iz-2)) + c4*(Txz(ix , iz+4)-Txz(ix , iz-3));

Vx(ix , iz) =    Vx(ix , iz) + (dt/dh)*( Dx + Dz )/Rho(ix , iz);
                }
        }

/* calcula vz para todos los puntos de la grilla , excepto puntos ficticios
*/

    for (ix=4; ix<Nx-3; ix++) {
        for (iz=4; iz<Nz-3; iz++) {
Dzz = c1*(Tzz(ix , iz+1) - Tzz(ix , iz) ) + c2*(Tzz(ix , iz+2) - Tzz(ix , iz-1))
      + c3*(Tzz(ix , iz+3)-Tzz(ix , iz-2)) + c4*(Tzz(ix , iz+4)-Tzz(ix , iz-3));
Dxz = c1*(Txz(ix+1,iz) - Txz(ix , iz) ) + c2*(Txz(ix+2,iz) - Txz(ix-1,iz))
      + c3*(Txz(ix+3,iz)-Txz(ix-2,iz)) + c4*(Txz(ix+4,iz)-Txz(ix-3,iz));

Vz(ix , iz) = Vz(ix , iz) + (dt/dh)*( Dzz + Dxz )/Rho(ix , iz);
                }
        }

```

0.7.1.2. Kernel en CPU para calcular las tensiones.

```

/* calcula Txx, Tzz para todos los puntos de la grilla , excepto puntos
ficticios */

    for (ix=5; ix<Nx-3; ix++) {
        for (iz=4; iz<Nz-3; iz++) {

dxvx = c1*(Vx(ix , iz)-Vx(ix -1, iz ) ) +    c2*(Vx(ix+1, iz ) - Vx(ix -2, iz ) ) +
        c3*(Vx(ix+2, iz ) - Vx(ix -3, iz ) ) +    c4*(Vx(ix+3, iz ) - Vx(ix -4, iz ) );
dzvz = c1*(Vz(ix , iz)-Vz(ix , iz -1) ) +    c2*(Vz(ix , iz+1) - Vz(ix , iz -2) ) +
        c3*(Vz(ix , iz+2) - Vz(ix , iz -3) ) +    c4*(Vz(ix , iz+3) - Vz(ix , iz -4) );

dxvx2 =c1*(Vx(ix , iz)-Vx(ix , iz -1)) + c2*(Vx(ix , iz+1)-Vx(ix , iz -2)) + c3*(Vx(
ix , iz+2)-Vx(ix , iz -3)) + c4*(Vx(ix , iz+3)-Vx(ix , iz -4));
dzvz2 =c1*(Vz(ix , iz)-Vz(ix -1, iz)) + c2*(Vz(ix+1, iz)-Vz(ix -2, iz)) + c3*(Vz(
ix+2, iz)-Vz(ix -3, iz)) + c4*(Vz(ix+3, iz)-Vz(ix -4, iz));

Txx(ix , iz) = Txx(ix , iz) + (dt/dh)*(C11(ix , iz)*dxvx) + (dt/dh)*(C13(ix , iz)*
dzvz) + (dt/dh)*(C15(ix , iz)*(dxvx2+dzvz2));
Tzz(ix , iz) = Tzz(ix , iz) + (dt/dh)*(C13(ix , iz)*dxvx) + (dt/dh)*(C33(ix , iz)*
dzvz) + (dt/dh)*(C35(ix , iz)*(dxvx2+dzvz2));
        }
    }

    for (ix=4; ix<Nx-3; ix++) {
        for (iz=5; iz<Nz-3; iz++) {

dxvx = c1*(Vx(ix , iz)-Vx(ix -1, iz ) ) +    c2*(Vx(ix+1, iz ) - Vx(ix -2, iz ) ) +
        c3*(Vx(ix+2, iz ) - Vx(ix -3, iz ) ) +    c4*(Vx(ix+3, iz ) - Vx(ix -4, iz ) );
dzvz = c1*(Vz(ix , iz)-Vz(ix , iz -1) ) +    c2*(Vz(ix , iz+1) - Vz(ix , iz -2) ) +
        c3*(Vz(ix , iz+2) - Vz(ix , iz -3) ) +    c4*(Vz(ix , iz+3) - Vz(ix , iz -4) );

dxvx2 =c1*(Vx(ix , iz)-Vx(ix , iz -1)) + c2*(Vx(ix , iz+1)-Vx(ix , iz -2)) + c3*(Vx(
ix , iz+2)-Vx(ix , iz -3)) + c4*(Vx(ix , iz+3)-Vx(ix , iz -4));
dzvz2 =c1*(Vz(ix , iz)-Vz(ix -1, iz)) + c2*(Vz(ix+1, iz)-Vz(ix -2, iz)) + c3*(Vz(
ix+2, iz)-Vz(ix -3, iz)) + c4*(Vz(ix+3, iz)-Vz(ix -4, iz));

Txz(ix , iz) = Txz(ix , iz) + (dt/dh)*C15(ix , iz)*(dxvx) + (dt/dh)*C35(ix , iz)*
dzvz + (dt/dh)*C55(ix , iz)*(dxvx2+dzvz2);
        }
    }

```

0.7.1.3. Kernel en CPU paralelizada que permite calcular las velocidades.

```

/* calculate vx for all grid points except on the virtual boundary
*/
#pragma parallel
#pragma omp for private (ix , iz ,Dx,Dz) nowait schedule(
    guided,1)
    for (ix=3; ix<Nx-3; ix++) {
#pragma ivdep
        for (iz=3; iz<Nz-3; iz++) {
            Dx = c1*(Txx(ix+1,iz)- Txx(ix ,iz)) + c2
                *(Txx(ix+2,iz)-Txx(ix -1,iz)) + c3*(Txx(
                    ix+3,iz)-Txx(ix -2,iz)) + c4*(Txx(ix+4,iz
                    )-Txx(ix -3,iz));
            Dz = c1*(Txz(ix ,iz+1)- Txz(ix ,iz ) ) + c2
                *(Txz(ix ,iz+2)-Txz(ix ,iz -1)) + c3*(Txz(
                    ix ,iz+3)-Txz(ix ,iz -2)) + c4*(Txz(ix ,iz
                    +4)-Txz(ix ,iz -3));

            Vx(ix ,iz ) = Vx(ix ,iz ) + (dt/dh)*( Dx +
                Dz )/Rho(ix ,iz );
        }
    }

/* calculate vz for all grid points except on the virtual
boundary */

#pragma parallel
#pragma omp for private (ix , iz ,Dzz,Dxz) nowait schedule(
    guided,1)
    for (ix=4; ix<Nx-3; ix++) {
#pragma ivdep
        for (iz=4; iz<Nz-3; iz++) {
Dzz = c1*(Tzz(ix ,iz+1) - Tzz(ix ,iz ) ) + c2*(Tzz(ix ,iz+2) - Tzz(ix ,iz -1))
+ c3*(Tzz(ix ,iz+3)-Tzz(ix ,iz -2)) + c4*(Tzz(ix ,iz+4)-Tzz(ix ,iz -3));
Dxz = c1*(Txz(ix+1,iz) - Txz(ix ,iz ) ) + c2*(Txz(ix+2,iz) - Txz(ix -1,iz ) )
+ c3*(Txz(ix+3,iz)-Txz(ix -2,iz)) + c4*(Txz(ix+4,iz)-Txz(ix -3,iz));
            Vz(ix ,iz ) = Vz(ix ,iz ) + (dt/dh)*( Dzz +
                Dxz )/Rho(ix ,iz );
        }
    }
}

```

0.7.1.4. Kernel en CPU paralelizada para calcular las tensiones.

```

/* calculate Txx/tzz for all grid points except on the virtual
boundary */
#pragma parallel
#pragma omp for private (ix , iz ,dxvx ,dzvz ,dxvx2 ,dzvz2)
    nowait schedule(guided,1)
    for (ix=5; ix<Nx-3; ix++) {
#pragma ivdep
        for (iz=4; iz<Nz-3; iz++) {
dxvx = c1*(Vx(ix ,iz)-Vx(ix -1,iz ) ) + c2*(Vx(ix+1,iz) - Vx(ix -2,iz ) ) +
c3*(Vx(ix+2,iz) - Vx(ix -3,iz ) ) + c4*(Vx(ix+3,iz) - Vx(ix -4,iz ) );

```

```

dzvz = c1*(Vz(ix , iz)-Vz(ix , iz -1) ) + c2*(Vz(ix , iz+1) - Vz(ix , iz -2) ) +
c3*(Vz(ix , iz+2) - Vz(ix , iz -3) ) + c4*(Vz(ix , iz+3) - Vz(ix , iz -4) );

dxvx2 =c1*(Vx(ix , iz)-Vx(ix , iz -1)) + c2*(Vx(ix , iz+1)-Vx(ix , iz -2)) + c3*(Vx(
ix , iz+2)-Vx(ix , iz -3)) + c4*(Vx(ix , iz+3)-Vx(ix , iz -4));
dzvz2 =c1*(Vz(ix , iz)-Vz(ix -1, iz)) + c2*(Vz(ix+1, iz)-Vz(ix -2, iz)) + c3*(Vz(
ix+2, iz)-Vz(ix -3, iz)) + c4*(Vz(ix+3, iz)-Vz(ix -4, iz));

Txx(ix , iz) = Txx(ix , iz) + (dt/dh)*(C11(ix , iz)*dxvx) + (dt/dh)*(C13(ix , iz)*
dzvz) + (dt/dh)*(C15(ix , iz)*(dxvx2+dzvz2));
Tzz(ix , iz) = Tzz(ix , iz) + (dt/dh)*(C13(ix , iz)*dxvx) + (dt/dh)*(C33(ix , iz)*
dzvz) + (dt/dh)*(C35(ix , iz)*(dxvx2+dzvz2));
}

}

#pragma parallel
#pragma omp for private (ix , iz ,Dx,Dz,Dx2,Dz2) nowait
schedule(guided ,1)
for (ix=4; ix<Nx-3; ix++) {
#pragma ivdep
for (iz=5; iz<Nz-3; iz++) {

dxvx = c1*(Vx(ix , iz)-Vx(ix -1, iz) ) + c2*(Vx(ix+1, iz) - Vx(ix -2, iz) ) +
c3*(Vx(ix+2, iz) - Vx(ix -3, iz) ) + c4*(Vx(ix+3, iz) - Vx(ix -4, iz) );
dzvz = c1*(Vz(ix , iz)-Vz(ix , iz -1) ) + c2*(Vz(ix , iz+1) - Vz(ix , iz -2) ) +
c3*(Vz(ix , iz+2) - Vz(ix , iz -3) ) + c4*(Vz(ix , iz+3) - Vz(ix , iz -4) );

dxvx2 =c1*(Vx(ix , iz)-Vx(ix , iz -1)) + c2*(Vx(ix , iz+1)-Vx(ix , iz -2)) + c3*(Vx(
ix , iz+2)-Vx(ix , iz -3)) + c4*(Vx(ix , iz+3)-Vx(ix , iz -4));
dzvz2 =c1*(Vz(ix , iz)-Vz(ix -1, iz)) + c2*(Vz(ix+1, iz)-Vz(ix -2, iz)) + c3*(Vz(
ix+2, iz)-Vz(ix -3, iz)) + c4*(Vz(ix+3, iz)-Vz(ix -4, iz));

Txz(ix , iz) = Txz(ix , iz) + (dt/dh)*C15(ix , iz)*(dxvx) + (dt/dh)*C35(ix , iz)*
dzvz + (dt/dh)*C55(ix , iz)*(dxvx2+dzvz2);

}

}

```

la implementación computacional en GPU esta realizada de tal forma por la jerarquía de grupos de hilos asumiendo el modelo de programación en CUDA donde cada hilo es responsable de tomar un punto desconocido de la malla quien esta dada por las dimensiones de modelo sobre el numero de bloques, teniendo en cuenta las limitaciones de hardware donde la cantidad de hilos por bloque no debe ser mayor a 1024.

0.7.1.5. Kernel en GPU para calcular las velocidades

```

__global__ void KernelVelocity(const float *Txx, const float *Tzz, const
float *Tzx, float *Vx, float *Vz, const float c1, const float c2, const float

```

```

    c3, const float c4, const float *Rho, const float dt, const float dh, const
    int Nx, const int Nz) {

float Dx, Dz, Dzz, Dxz;

int ix = threadIdx.x+blockIdx.x*blockDim.x;
int iz = threadIdx.y+blockIdx.y*blockDim.y;

unsigned int idx = ix*Nz + iz;

bool checkVx = (ix>=3 && ix<Nx-3) && (iz>=3 && iz<Nz-3);
bool checkVz = (ix>=4 && ix<Nx-3) && (iz>=4 && iz<Nz-3);

    if (checkVx) {

Dx = c1*(Txx(ix+1,iz)- Txx(ix , iz)) + c2*(Txx(ix+2,iz)-Txx(ix-1,iz)) + c3
    *(Txx(ix+3,iz)-Txx(ix-2,iz)) + c4*(Txx(ix+4,iz)-Txx(ix-3,iz));
Dz = c1*(Txz(ix , iz+1)- Txz(ix , iz) ) + c2*(Txz(ix , iz+2)-Txz(ix , iz-1)) + c3
    *(Txz(ix , iz+3)-Txz(ix , iz-2)) + c4*(Txz(ix , iz+4)-Txz(ix , iz-3));
__syncthreads();
Vx[idx] = Vx[idx] + (dt/dh)*( Dx + Dz )/Rho(ix , iz);
    }

    if (checkVz){

Dzz = c1*(Tzz(ix , iz+1) - Tzz(ix , iz) ) + c2*(Tzz(ix , iz+2) - Tzz(ix , iz-1))
    + c3*(Tzz(ix , iz+3)-Tzz(ix , iz-2)) + c4*(Tzz(ix , iz+4)-Tzz(ix , iz-3));
Dxz = c1*(Txz(ix+1,iz) - Txz(ix , iz) ) + c2*(Txz(ix+2,iz) - Txz(ix-1,iz))
    + c3*(Txz(ix+3,iz)-Txz(ix-2,iz)) + c4*(Txz(ix+4,iz)-Txz(ix-3,iz));
__syncthreads();
Vz[idx] = Vz[idx] + (dt/dh)*( Dzz + Dxz )/Rho(ix , iz);

    }

}

```

0.7.1.6. Kernel en GPU para calcular las tensiones

```

__global__ void KernelStress(const float *Vx, const float *Vz, float *Txx,
    float *Tzz, float *Txz,
    const float c1, const float c2, const float c3, const float c4,
    const float *C11, const float *C33, const float *C13, const
    float *C15, const float *C35, const float *C55, const float dt,
    const float dh, const int Nx, const int Nz) {

float dxvx, dzvz, dxvx2, dzvz2;

int ix = threadIdx.x+blockIdx.x*blockDim.x;
int iz = threadIdx.y+blockIdx.y*blockDim.y;

unsigned int idx = ix*Nz + iz;

```



```

bool checkTxx_Tzz = (ix>=5 && ix<Nx-3)    && (iz>=4 && iz<Nz-3);
bool checkTxz    = (ix>=4 && ix<Nx-3)    && (iz>=5 && iz<Nz-3);

    if (checkTxx_Tzz) {

dxvx = c1*(Vx(ix,iz)-Vx(ix-1,iz)) +    c2*(Vx(ix+1,iz) - Vx(ix-2,iz)) +
      c3*(Vx(ix+2,iz) - Vx(ix-3,iz)) +    c4*(Vx(ix+3,iz) - Vx(ix-4,iz));
dzvz = c1*(Vz(ix,iz)-Vz(ix,iz-1)) +    c2*(Vz(ix,iz+1) - Vz(ix,iz-2)) +
      c3*(Vz(ix,iz+2) - Vz(ix,iz-3)) +    c4*(Vz(ix,iz+3) - Vz(ix,iz-4));

dxvx2 =c1*(Vx(ix,iz)-Vx(ix,iz-1)) + c2*(Vx(ix,iz+1)-Vx(ix,iz-2)) + c3*(Vx(
  ix,iz+2)-Vx(ix,iz-3)) + c4*(Vx(ix,iz+3)-Vx(ix,iz-4));
dzvz2 =c1*(Vz(ix,iz)-Vz(ix-1,iz)) + c2*(Vz(ix+1,iz)-Vz(ix-2,iz)) + c3*(Vz(
  ix+2,iz)-Vz(ix-3,iz)) + c4*(Vz(ix+3,iz)-Vz(ix-4,iz));

__syncthreads();
Txx[idx] = Txx[idx] + (dt/dh)*(C11(ix,iz)*dxvx) + (dt/dh)*(C13(ix,iz)*dzvz
) + (dt/dh)*(C15(ix,iz)*(dxvx2+dzvz2));
__syncthreads();
Tzz[idx] = Tzz[idx] + (dt/dh)*(C13(ix,iz)*dxvx) + (dt/dh)*(C33(ix,iz)*dzvz
) + (dt/dh)*(C35(ix,iz)*(dxvx2+dzvz2));

    }

    if (checkTxz) {

dxvx = c1*(Vx(ix,iz)-Vx(ix-1,iz)) +    c2*(Vx(ix+1,iz) - Vx(ix-2,iz)) +
      c3*(Vx(ix+2,iz) - Vx(ix-3,iz)) +    c4*(Vx(ix+3,iz) - Vx(ix-4,iz));
dzvz = c1*(Vz(ix,iz)-Vz(ix,iz-1)) +    c2*(Vz(ix,iz+1) - Vz(ix,iz-2)) +
      c3*(Vz(ix,iz+2) - Vz(ix,iz-3)) +    c4*(Vz(ix,iz+3) - Vz(ix,iz-4));

dxvx2 =c1*(Vx(ix,iz)-Vx(ix,iz-1)) + c2*(Vx(ix,iz+1)-Vx(ix,iz-2)) + c3*(Vx(
  ix,iz+2)-Vx(ix,iz-3)) + c4*(Vx(ix,iz+3)-Vx(ix,iz-4));
dzvz2 =c1*(Vz(ix,iz)-Vz(ix-1,iz)) + c2*(Vz(ix+1,iz)-Vz(ix-2,iz)) + c3*(Vz(
  ix+2,iz)-Vz(ix-3,iz)) + c4*(Vz(ix+3,iz)-Vz(ix-4,iz));

__syncthreads();
Txz[idx] = Txz[idx] + (dt/dh)*C15(ix,iz)*(dxvx) + (dt/dh)*C35(ix,iz)*dzvz
+ (dt/dh)*C55(ix,iz)*(dxvx2+dzvz2);

    }

}

```

0.7.2. Medio homogéneo

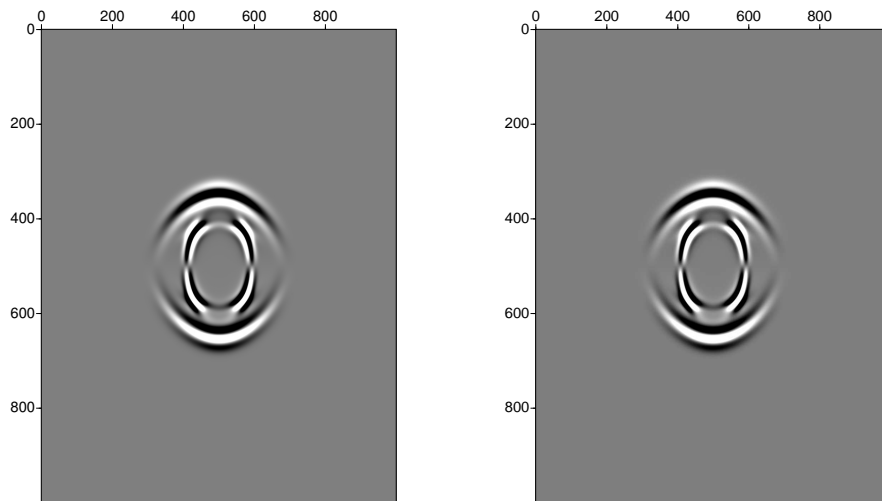
	$C_{11}(GPa)$	$C_{33}(GPa)$	$C_{55}(GPa)$	$C_{13}(GPa)$	$C_{15}(GPa)$	$C_{35}(GPa)$	$\rho(kg/m)$
VTI	12.67	8.80	3.17	2.89	0	0	2200
TTI (Ángulo de Inclinación de 30°)	11.14	9.20	3.73	3.45	1.16	0.51	2200
TTI (Ángulo de Inclinación de 45°)	9.98	9.98	3.92	3.64	0.97	0.97	2200
TTI (Ángulo de Inclinación de 60°)	9.2	11.14	3.73	3.45	0.51	1.16	2200

Tabla 2: Constantes elásticas para un medio TI, tomadas de (11)

Los parámetros que se tuvieron en cuenta durante la simulación se encuentran resumidos a continuación:

Tamaño del paso de tiempo	0.000800 [s]
Tamaño del paso en el espacio	5.000000 [m]
Tiempo simulado	0.5 [s]
Longitud	1000 [m]
Profundidad	1000 [m]
Frecuencia de la Ondicula	8 [Hz]

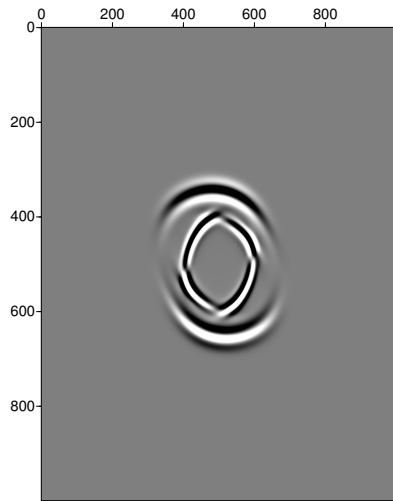
Tabla 3: Parámetros que se tuvieron en cuenta durante la simulación.



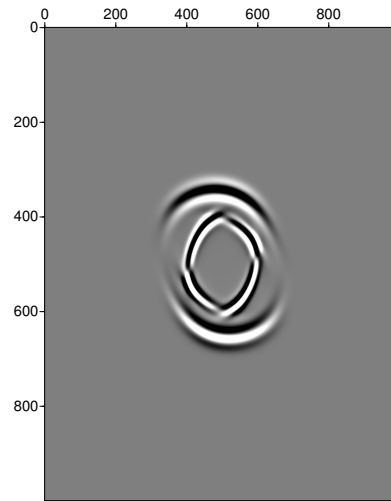
(a) VTI CPU

(b) VTI GPU

Figura 12: Capturas para la componente vertical en un medio VTI tanto en CPU como GPU

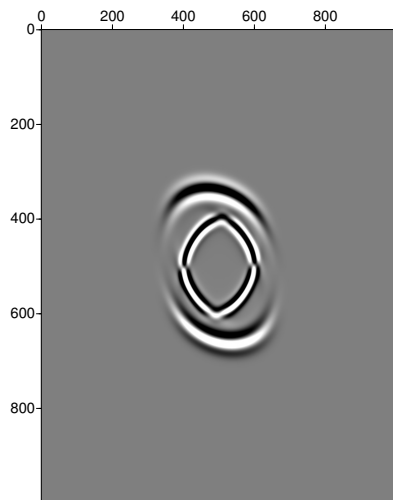


(a) TTI CPU 30°

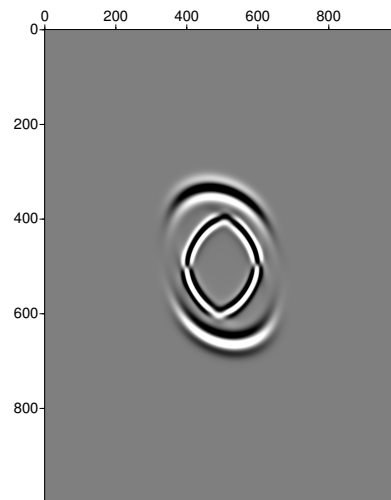


(b) TTI GPU 30°

Figura 13: Capturas para la componente vertical en un medio TTI con un ángulo de inclinación de 30° tanto en CPU como GPU

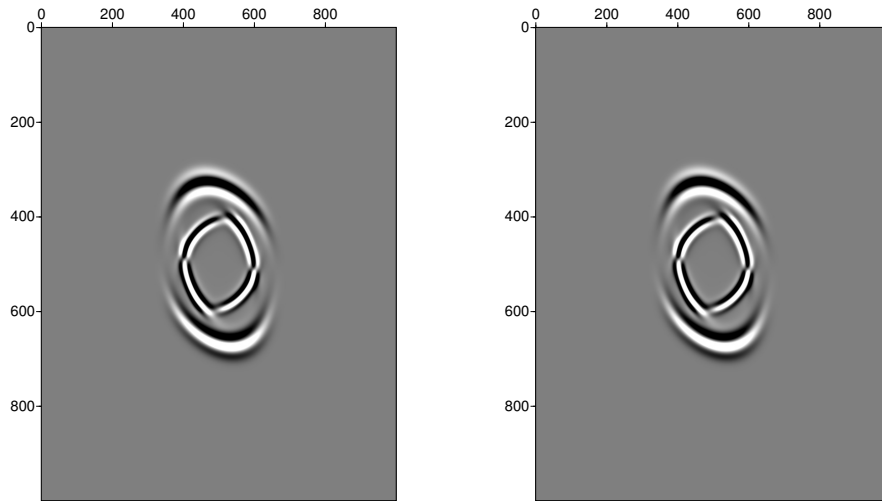


(a) TTI CPU 45°



(b) TTI GPU 45°

Figura 14: Capturas para la componente vertical en un medio TTI con un ángulo de inclinación de 45° tanto en CPU como GPU



(a) TTI CPU 60°

(b) TTI GPU 60°

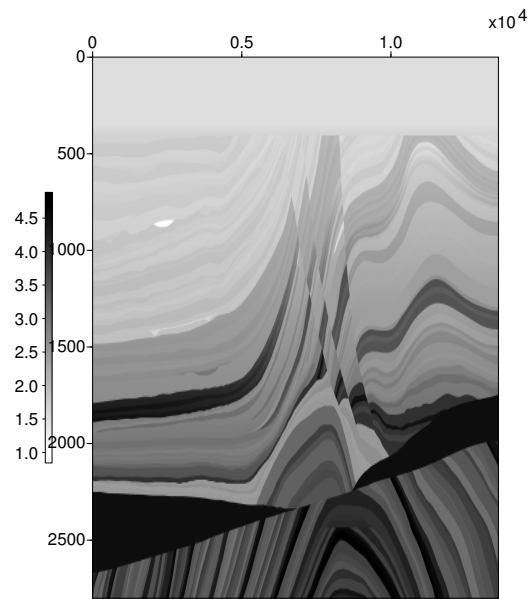
Figura 15: Capturas para la componente vertical en un medio TTI con un ángulo de inclinación de 60° tanto en CPU como GPU

0.7.3. Medio no-homogéneo

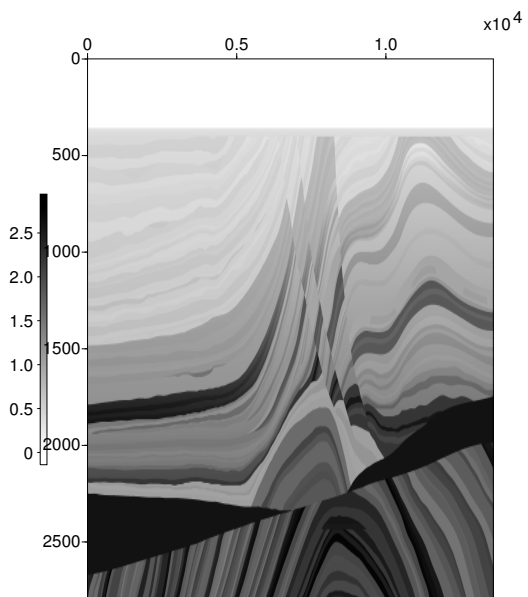
Para la simulación de un medio isótropo utilizamos el modelo de marmousi2 el cual es usado para aspectos prácticos en procesamiento de datos sísmicos como migración e inversión. Como segundo para modelar la propagación de ondas se usó el modelo de isotropía vertical transversal SEG 2D Hess VTI los modelos de isotropía transversal con eje de simetría vertical VTI son la representación más común para cuencas sedimentarias (Sayers, 2010). Estos muestran una buena aproximación para la anisotropía de las estructuras geológicas y son de especial utilidad cuando se desea poner a prueba procesos de migración (Liu Zhang, 2011).

0.7.3.1. MARMOUSI2

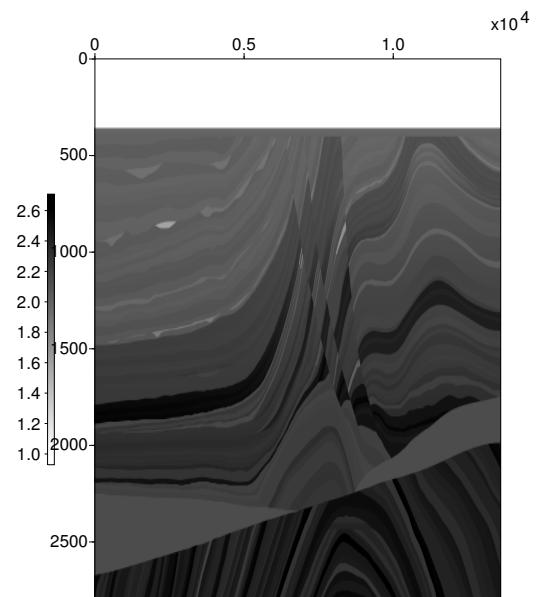
El modelo de Marmousi, fue creado inicialmente por el Instituto Francés del Petróleo (IFP), El conjunto de datos Marmousi2 es una extensión y mejora elástica isotrópica de los datos del modelo clásico de Marmousi. Fue creado por Allied Geophysical Laboratories (AGL).



(a) V_p

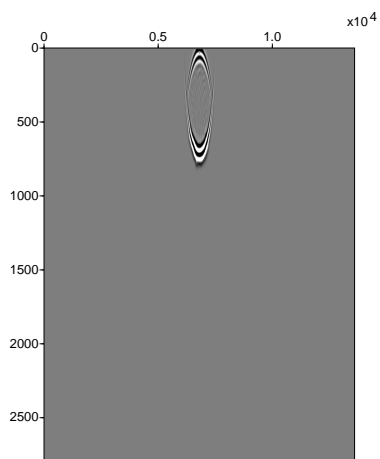


(b) V_s

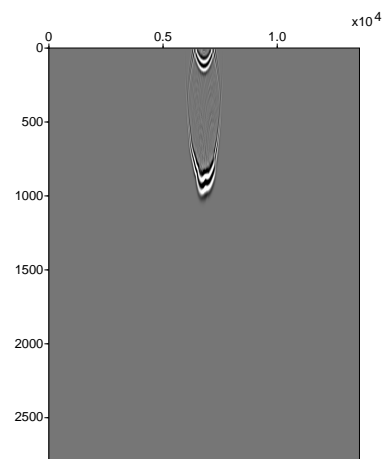


(c) ρ

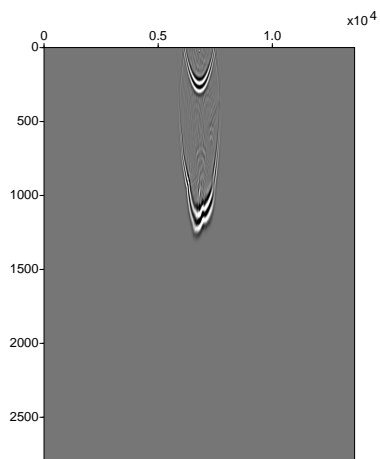
Figura 16: Marmousi2. Versión elástica isotrópica de los datos originales de Marmousi.



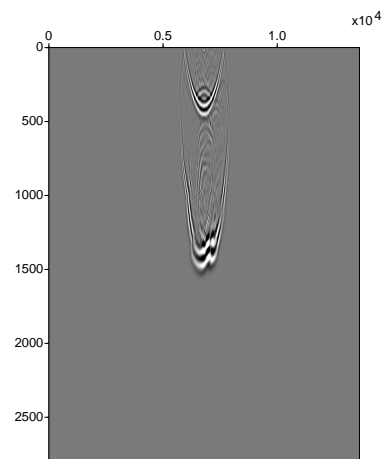
(a)



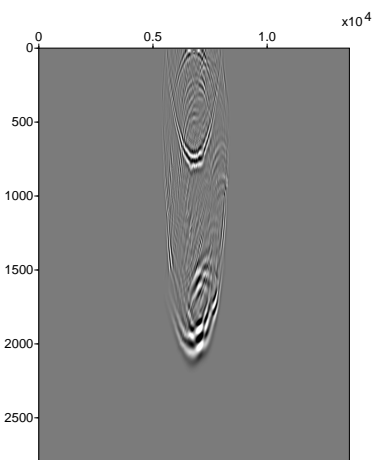
(b)



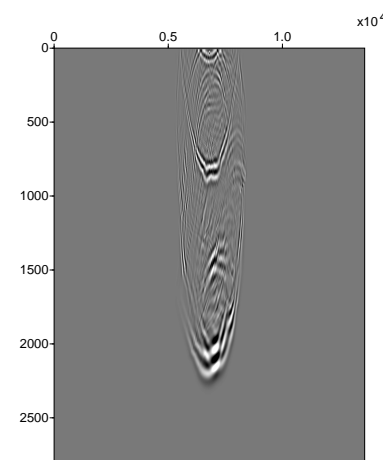
(c)



(d)



(e)

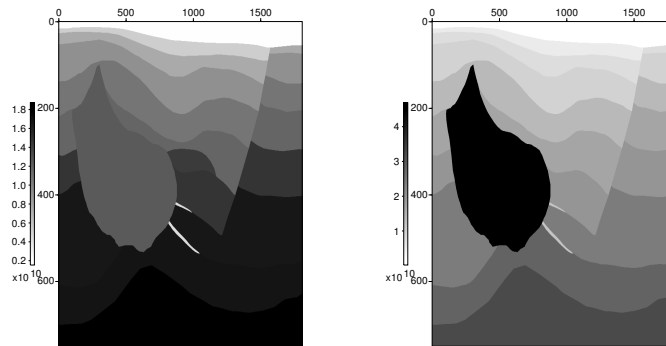


(f)

Figura 17: Capturas de Frente de onda modelo MARMOUSI2 (a) a (f)

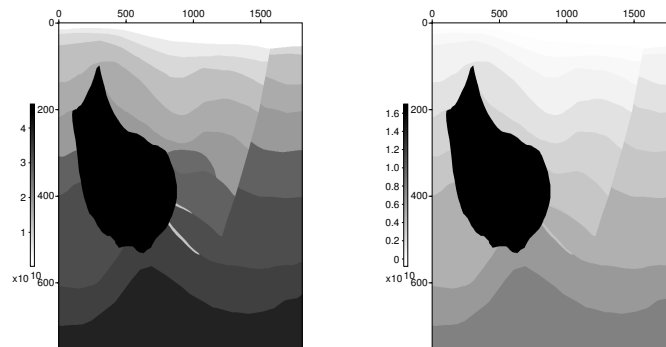
0.7.3.2. HESS VTI

Conjuntos de datos Hess VTI para probar la migración anisotrópica 2D.



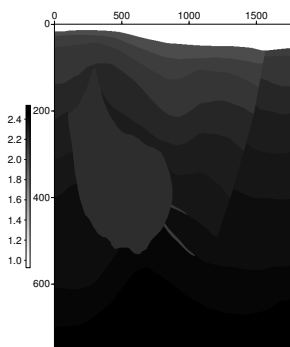
(a) C13

(b) C33



(c) C11

(d) C44



(e) RHO

Figura 18: Modelo Hess Corporation VTI. Disponible desde SEG.

Los parámetros que se tuvieron en cuenta durante la simulación se encuentran resumidos a continuación:

Tamaño del paso de tiempo	0.000100 [s]
Tamaño del paso en el espacio	40.000000 [m]
Tiempo simulado	0.15 [s]
Longitud	1809 [m]
Profundidad	750 [m]
Velocidad min	7063.183135 [m/s]
Velocidad max cmax	142651.614109 [m/s]
Frecuencia de la Ondicula	50 [Hz]

Tabla 4: Parámetros que se tuvieron en cuenta durante la simulación.

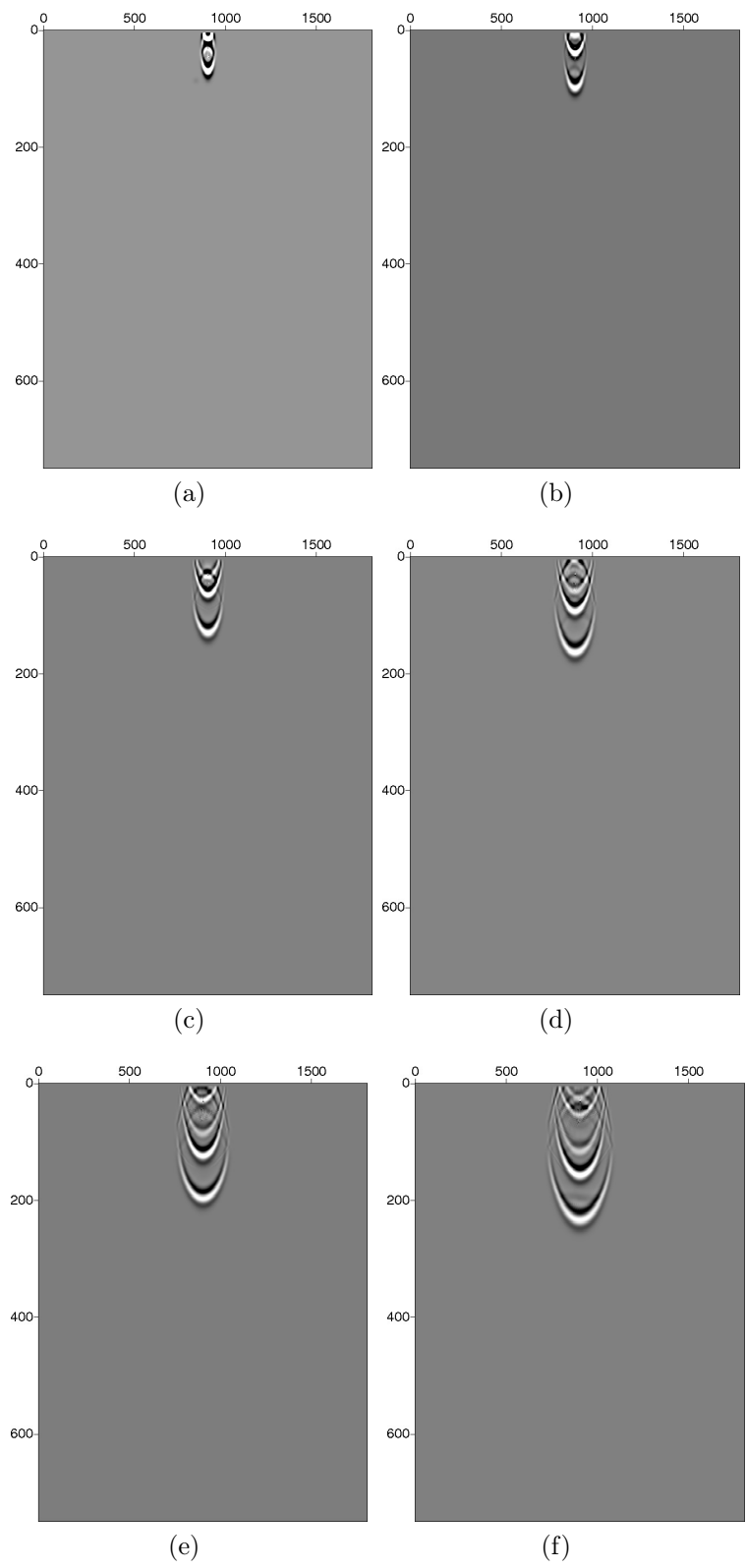
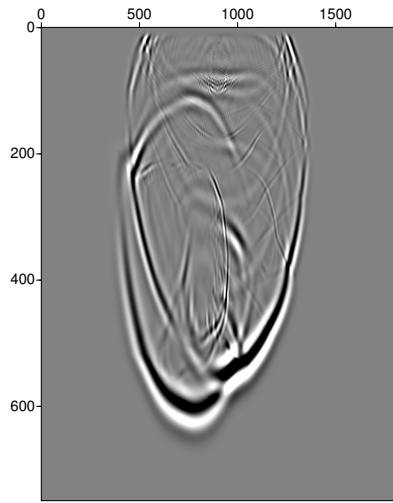
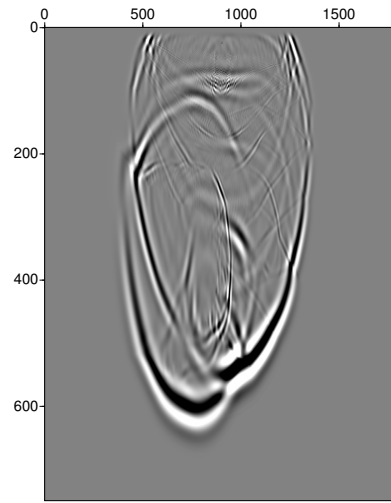


Figura 19: Capturas de Frente de onda modelo HESS VTI (a) a (f)

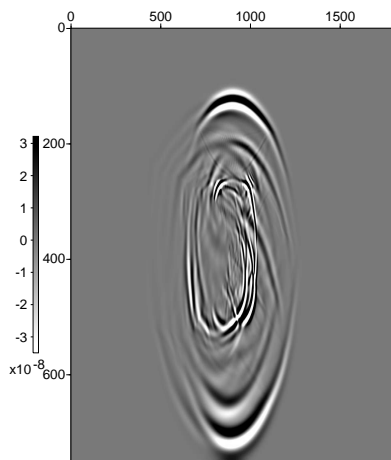


(a) CPU

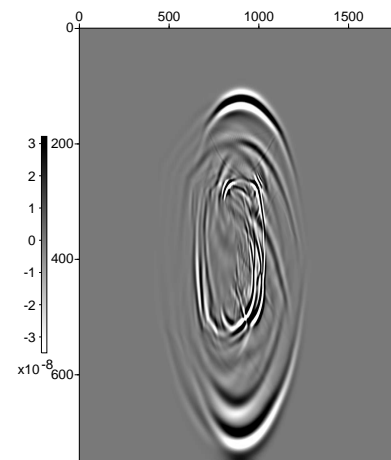


(b) GPU

Figura 20: Capturas para la componente vertical en un medio VTI tanto en CPU como GPU



(a) CPU



(b) GPU

Figura 21: Capturas para la componente vertical en un medio VTI tanto en CPU como GPU fuente ubicada en el centro del modelo

Capitulo 6

0.8. Análisis de Resultados

0.8.1. Comparación del tiempo de ejecución: CPU vs GPU

Esta comparación se centrará en la capacidad de la GPU en comparación con la CPU en el tiempo de ejecución que se definirá como el tiempo requerido para la actualización de componentes de velocidad y tensión en determinado tiempo. Para llevar a cabo la simulación utilizamos un medio homogéneo donde los coeficientes elásticos son mostrados en la **Tabla 2**, la fuente es introducida en el centro del modelo con una frecuencia principal de 8 Hz aplicada en T_{zz} el tamaño de paso de malla del modelo se establece en 5m tanto en direcciones x y z con paso del tiempo dt de 0,0008s en un tiempo total t de 0,5s parámetros mostrados en la **Tabla 3** podemos observar la comparación de los resultados obtenidos mostrados en las **Figuras 12,13,14 y 15** para una malla de 1000x1000 (m) tanto en CPU como en GPU.

Malla (m)	CPU(s)	GPU(s)
200x200	31	8.2279
400x400	62	14.4322
600x600	140	24.396
800x800	271.3147	38.6239
1200x1200	422.1184	56.7864
1400x1400	834.5114	106.6789
1600x1600	1141.8677	137.2006
1800x1800	1390.1018	164.4831
2000x2000	1748.022	214.1634

Tabla 5: Tiempos promedio de procesamiento de CPU y GPU para un tamaño de dado de malla.

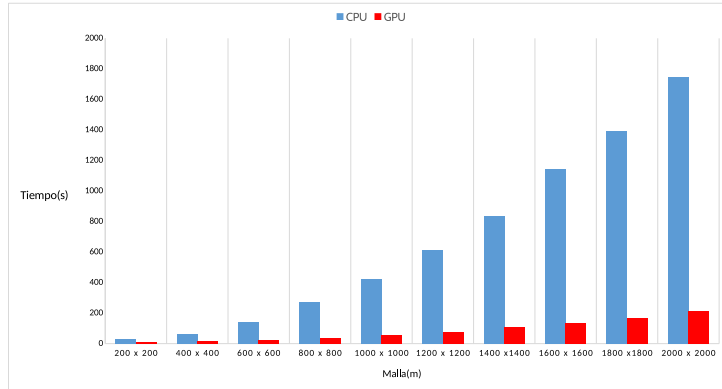


Figura 22: Tiempo de procesamiento CPU vs GPU

Posteriormente utilizando los parámetros mostrados en la Tabla 4 tomamos los tiempos empleados para el procesamiento en CPU Y GPU en modelo HESS VTI muestreando el modelo en un determinado tamaño de malla.

HESS VTI	CPU (s)	GPU(s)
150x1809	61.059699	10.282363
350x1809	152.603821	20.652543
550x1809	244.787497	32.251140
750x1809	305.529624	46.310420

Tabla 6: Tiempo de procesamiento modelo HESS VTI

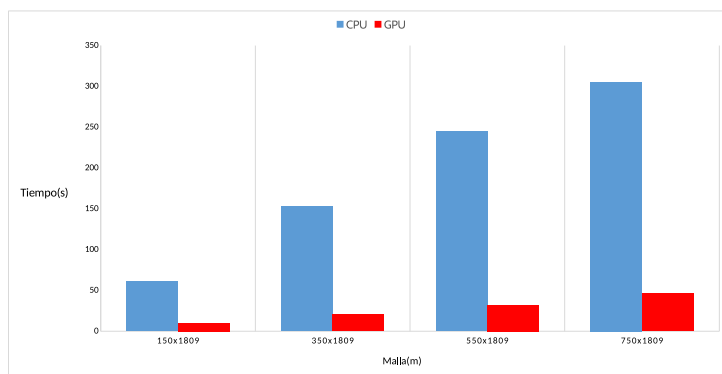


Figura 23: Tiempo de procesamiento CPU vs GPU

0.9. Conclusiones y Recomendaciones

De acuerdo a los resultados obtenidos es posible concluir que para grandes cantidades de datos el desarrollo de algoritmos utilizando la tecnología GPU/CUDA se convierte en un alternativa que permite sacar provecho de la arquitectura de la tarjeta junto a la programación en paralelo de CUDA.

Bibliografía

- [1] Laxmidhar Behera. Elastic anisotropic finite-difference full-wave modeling and imaging of the tilted transversely isotropic (TTI) media. 2008.
- [2] J. M. Carcione, D. Kosloff, and R. Kosloff. Wave-propagation simulation in an elastic anisotropic (transversely isotropic) solid. *Quarterly Journal of Mechanics and Applied Mathematics*, 41(3):319–346, 1988.
- [3] Antonio Carrillo and Omar Mendoza. Introducción al Método de Diferencias Finitas y su Implementación Computacional. page 59, 2015.
- [4] John Cheng, Max Grossman, and Ty McKercher. *Professional CUDA C Programming*, volume 53. 2013.
- [5] Primary Concerns. Seismic Modeling. pages 7–85.
- [6] John T. Etgen. Finite-difference elastic anisotropic wave propagation, 1987.
- [7] Eduardo L Faria and Paul L Stoffa. Finite-difference modeling in transversely isotropic media. *Geophysics*, 59(2):282–289, 1994.
- [8] Proyecto De Grado. No Title. 2009.
- [9] Vladimir Grechka, Andres Pech, Ilya Tsvankin, and Baoniu Han. Velocity analysis for tilted transversely isotropic media: A physical modeling example. *Geophysics*, 66(3):904, 2001.
- [10] Fiona Hall and Yanghua Wang. Elastic wave modelling by an integrated finite difference method. *Geophysical Journal International*, 177(1):104–114, 2009.
- [11] Byeongho Han, Soon Jee Seol, and Joongmoo Byun. Elastic modeling in 3D tilted transversely isotropic (TTI) media with convolutional perfectly matched layer (CPML) boundary conditions. *SEG Technical Program Expanded Abstracts 2011*, (2):3037–3041, 2011.
- [12] Byeongho Han, Soon Jee Seol, and Joongmoo Byun. Elastic modeling in 3D tilted transversely isotropic (TTI) media with convolutional perfectly matched layer (CPML) boundary conditions. *SEG Technical Program Expanded Abstracts 2011*, (2):3037–3041, 2011.
- [13] C Huang and L G Dong. High-order finite-difference method in seismic wave simulation with variable grids and local time-steps. *Chinese Journal of Geophysics-Chinese Edition*, 52(1):176–186, 2009.
- [14] Por Jairo, Alberto Torres, and Grupo D E Geof. Modelado Acústico Y Migración Pspi En. 2010.

- [15] Christopher Juhlin. Finitedifference elastic wave propagation in 2D heterogeneous transversely isotropic media. *Geophysical Prospecting*, 43(6):843–858, 1995.
- [16] C Kerner, B Dyer, and M Worthington. Wave propagation in a vertical transversely isotropic medium : field experiment and model study. pages 295–309, 1989.
- [17] Alan R Levander. Fourth-order finite-difference P-W seismograms. 53(11), 1988.
- [18] Short Note. +—————1 40. 50(4):705–708, 1985.
- [19] Felix Rubio, Mauricio Hanzich, Albert Farrés, Josep de la Puente, and José María Cela. Finite-difference staggered grids in GPUs for anisotropic elastic wave propagation simulation. *Computers and Geosciences*, 70:181–189, 2014.
- [20] Jan Thorbecke. 2D Finite-Difference Wavefield Modelling. *Book*, pages 1–47, 2013.
- [21] C[] Tsingas, A[] Vafidis, and E[] R[] Kanasewich. Elastic wave propagation in transversely isotropic media using finite differences. *Geophys. Prosp.*, 38(08):933–950, 1990.
- [22] D E Parámetros Anisótropos Una, Jenny Rueda, Hernan Madero, and Zully Calderón. Metodología Para Determinar Densidad De Microfracturas En Rocas , a Partir Methodology To Determine Density of Microfracture on Rocks From Anisotropics Parameters : an. 32, 2010.
- [23] Jean Virieux. P-SV wave propagation in heterogeneous media Velocity-stress finite difference method. *Geophysics*, 1986.
- [24] Yanghua Wang. Frequencies of the Ricker wavelet. *Geophysics*, 80(2):A31–A37, 2015.
- [25] Pengliang Yang. A numerical tour of wave propagation. *Madagascar*, 2(4):1–34, 2014.
- [26] Zhengsheng Yao and Gary F Margrave. Elastic wavefield modelling in 3D by fourth-order staggered- grid finite difference technique. *CREWES Research Report*, 12(1):10, 2000.