

**ALGORITMO PARA CLASIFICAR MODOS DE PROPAGACION DE ONDAS
GUIADAS**

AUTOR

SANTIAGO MENDEZ LOPEZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER

FACULTAD DE INGENIERIAS FISICO-MECANICAS

ESCUELA DE INGENIERIA MECANICA

BUCARAMANGA

2024

**ALGORITMO PARA CLASIFICAR MODOS DE PROPAGACION DE ONDAS
GUIADAS**

SANTIAGO MENDEZ LOPEZ

Trabajo de grado para optar a título de Ingeniero Mecánico

**DIRECTOR
JABID EDUARDO QUIROGA MENDEZ
Ingeniero Mecánico**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO-MECANICAS
ESCUELA DE INGENIERIA MECANICA
BUCARAMANGA**

2024

Dedicatoria

A mi madre, cuyo amor y sacrificio me han guiado en cada paso que he dado, mostrándome la verdadera esencia de la perseverancia y la dedicación. A mi padre, que con su sabiduría y enseñanzas ha sido la brújula que me orienta en los momentos más oscuros. A mi hermano mayor, cuya fortaleza y apoyo y consejos incondicional han sido el pilar que me sostiene en cada desafío que enfrento. Y a mi abuela que siempre me ha apoyado y ha confiado en mí sin importar nada me ha ayudado a perseverar y a estar preparado para cualquier reto.

Esta tesis es un tributo a ustedes, a su amor, a su apoyo y a todas las enseñanzas que me han dado. Gracias por ser mi motivación y por creer en mí incluso cuando yo mismo no lo hacía.

Contenido

1.	INTRODUCCIÓN.....	11
3.	OBJETIVOS.....	14
3.1.	Objetivo general.....	14
3.2.	Objetivos Específicos.....	14
4.	MARCO TEÓRICO	15
4.1.	Antecedente.....	15
4.2.	Referentes teóricos.....	16
4.2.1.	Curvas de dispersión	17
4.2.2.	Perfil de ondas	17
4.2.3.	Ensayos no destructivos (END)	19
4.2.4.	Análisis Modal	19
4.2.5.	Generación y propagación de señales ultrasónicas.....	20
4.2.6.	Método Scaled Boundary Finite Element Method (SBFEM).....	22
5.	DISEÑO METODOLÓGICO	23
5.1.	Pre-fase A: Análisis de la Propuesta de Diseño:	24
5.1.1.	Condiciones de Trabajo del Algoritmo:	27
5.2.	Fase A: Diseño Conceptual del Algoritmo:	28
5.2.1.	Matrices	28
5.2.2.	Autovalores	31
5.2.3.	Filtrado de perfiles de onda.....	32
5.2.4.	Pre-ordenamiento.....	34
5.2.5.	Visualización grafica de los resultados de los perfiles de onda.....	38
5.2.6.	Definición de funciones y resolución de valores propios	40
5.3.	Fase B: Definición de Parámetros del Algoritmo	44
5.4.	Fase C: Optimización de Parámetros y Diseño Detallado.....	46
5.4.1.	Métodos de cálculo optimizados	46
5.4.2.	Optimización de la resolución de valores propios.....	47
5.4.3.	Visualización y representación de datos	47
5.4.4.	Cumplimiento y adaptabilidad.....	48
5.5.	Fase D: Desarrollo y Pruebas del Algoritmo	48

5.5.1. Pruebas del algoritmo.....	48
5.5.2. Metodología de Ordenamiento y Seguimiento de Modos en Diagramas de Dispersión de Ondas Guiadas	55
6. PRESUPUESTO	61
7. CONCLUSIONES.....	62
8. OBSERVACIONES	64
9. RECOMENDACIONES.....	65
10. BIBLIOGRAFIA	67
ANEXOS	69

Tabla de figuras

Figura 1 Diferentes perfiles de onda.....	19
Figura 2 Ilustración del algoritmo de rastreo de raíces utilizado por PCDISP.....	21
Figura 3(a) Fase y (b) velocidades de grupo de los primeros modos axisimétricos de la guía de ondas.	22
Figura 4 Dimensiones de la placa de acero 1020.....	25
Figura 5 Matriz de rigidez.....	29
Figura 6 Matriz T de Transformación.....	30
Figura 7 Autovalores.....	31
Figura 8 Filtro de perfiles de onda.....	33
Figura 9 Preordenamiento de modos	35
Figura 10 Inicialización de Matrices y Vectores para Análisis Modal en MATLAB	35
Figura 11 Bucle de Preprocesamiento y Ordenamiento de perfiles de onda en MATLAB	36
Figura 12 Ajuste y Ordenamiento de perfiles de onda a través de Frecuencias en MATLAB....	37
Figura 13 Código de Visualización para Gráfico de 'f vs k'	39
Figura 14 Código de Visualización para Gráfico de 'f vs Cp'	39
Figura 15 Código de Visualización para Gráfico de 'f vs Cg'.....	40
Figura 16 Cálculo de Propiedades Elásticas y Velocidades de Onda (prop).....	41
Figura 17 Función para Propiedades Complejas de Onda (prop_c)	41
Figura 18 Función para Cálculo de Autovalores de Rigidez (k_eig).....	42
Figura 19 Generador de Matriz de Forma (SF)	43
Figura 20 Función para Crear Matriz Transpuesta de Identidad (mT)	44
Figura 21 Función para Cálculo de Eigenvalores Laplacianos (lapiz_eig)	47

Figura 22 Comparativa de Velocidad de Grupo: Código Antiguo vs. Optimizado.....	49
Figura 23 Evolución de Velocidad de Fase: Antes y Después de la Optimización.....	50
Figura 24 Gráfico de Frecuencia vs Constante elastica (f vs k)	51
Figura 25 Ordenamiento de Modos de Propagación de Ondas Guiadas en Aluminio	53
Figura 26 Ordenamiento de Modos de Propagación de Ondas Guiadas en Cobre con Espesor de 5 mm	54
Figura 27 Seguimiento de Modos en Diagramas de Dispersión.....	56

Tablas

Tabla 1 Propiedades acero 1020	25
Tabla 2 Propiedades complejas del acero 1020	¡Error! Marcador no definido.
Tabla 3 Condiciones de trabajo.....	27
Tabla 4 Requerimientos	46
Tabla 5 Presupuesto del proyecto	61

Resumen

Título: Algoritmo para Clasificar Modos de Propagación de Ondas Guidas.

Autor: Santiago Mendez Lopez

Palabras Clave: Algoritmo, Modos de Ondas, Placas, Curvas de Dispersión, Número MAC, Análisis Modal, MATLAB.

Descripción: En el marco de esta investigación, se ha desarrollado un algoritmo con el propósito principal de agrupar modos de ondas en placas, basándose en la información proveniente de las curvas de dispersión. El estudio se centró en la aplicación de técnicas avanzadas de procesamiento de señales para extraer características esenciales de las mencionadas curvas. A través de la implementación de un método de clasificación basado en el Número MAC, se logró agrupar datos con precisión. La herramienta principal para el desarrollo y simulación del algoritmo fue MATLAB. Adicionalmente, se llevó a cabo una evaluación comparativa del algoritmo propuesto respecto a otros métodos de clasificación disponibles en la literatura. Este algoritmo facilita y agiliza la supervisión y análisis de datos relacionados con los modos de ondas en placas, aportando significativamente al campo de los ensayos no destructivos y a la seguridad estructural de las placas y tubos en diversas aplicaciones de ingeniería

Abstract

Title: Algorithm for Classifying Guided Wave Propagation Modes.

Author: Santiago Mendez Lopez

Keywords: Algorithm, Wave Modes, Plates, Dispersion Curves, MAC Number, Modal Analysis, MATLAB.

Description: Within the scope of this research, an algorithm has been developed with the primary purpose of grouping wave modes in plates, drawing upon the information from the dispersion curves. The study focused on the application of advanced signal processing techniques to extract essential features from the mentioned curves. Through the implementation of a classification method based on the MAC Number, precise data grouping was achieved. MATLAB served as the primary tool for the development and simulation of the algorithm. Additionally, a comparative evaluation of the proposed algorithm was conducted against other available classification methods in the literature, showcasing the efficiency of our proposal. This algorithm streamlines and facilitates the supervision and analysis of data related to wave modes in plates, significantly contributing to the field of non-destructive testing and the structural safety of plates and tubes in various engineering applications.

1. INTRODUCCIÓN

A pesar de que el método de Inspección no Destructivo por Ondas Guiadas (NDT-GW) ha demostrado ser una técnica efectiva para la detección de discontinuidades en estructuras de gran tamaño, su implementación resulta compleja y costosa debido a la necesidad de análisis detallados de las curvas de dispersión para la identificación de modos de propagación de las ondas. Actualmente, la clasificación de estos modos se realiza a través de diversos algoritmos que no son totalmente precisos lo que aumenta el riesgo de errores y la falta de confiabilidad en los resultados. Por lo tanto, existe la necesidad de desarrollar un algoritmo que permita la identificación automatizada y precisa de los modos de propagación de ondas guiadas en placas.

Este proyecto tiene como fin agilizar y optimizar el análisis de datos que se obtienen de las diferentes pruebas que se realizan a placas

El número MAC del inglés Modal Assurance Criterion desempeña un papel crítico al establecer la similitud entre los perfiles de onda de las estructuras, permitiendo la validación y comparación de modelos numéricos y experimentales. Su importancia radica en su capacidad para cuantificar el grado de correspondencia entre los perfiles de onda, lo que a su vez facilita la identificación de modos redundantes y la mejora de la precisión de los modelos.

En este contexto, la correlación del algoritmo propuesto con el número MAC resulta fundamental para la correcta clasificación de los modos de propagación de ondas guiadas. Esta correlación permitirá una identificación más precisa de los perfiles de onda en las estructuras analizadas, lo que a su vez contribuirá a la comprensión del comportamiento dinámico de dichas estructuras. Como resultado, se podrán tomar decisiones informadas en cuanto a la seguridad, rendimiento y durabilidad de las estructuras en diversos sectores industriales.

2. JUSTIFICACIÓN

En la era de la información, la velocidad y precisión en la recopilación y procesamiento de datos se ha vuelto fundamental para tomar decisiones en diferentes áreas de la ingeniería. En el campo de los ensayos no destructivos (END), esto es especialmente importante. La integridad y seguridad de las infraestructuras y materiales dependen en gran medida de nuestra capacidad para detectar y diagnosticar defectos y anomalías a tiempo. Sin embargo, los avances en técnicas de recolección de datos han generado grandes cantidades de información, por lo que ahora el desafío radica en cómo procesar, interpretar y utilizar esa información eficientemente.

Aquí es donde se vuelve crucial optimizar el tratamiento de datos. Con la implementación de algoritmos eficientes como el propuesto en este proyecto, buscamos agilizar la distribución de datos relacionados con los modos de ondas. Estos algoritmos permitirán simplificar el análisis y supervisión de los datos, convirtiendo lo que antes eran grandes volúmenes de información en conjuntos manejables y comprensibles que se puedan utilizar rápidamente para tomar decisiones. El Número MAC (Modal Assurance Criterion) juega un papel fundamental en este proceso. Al ofrecer una medida cuantitativa de la similitud entre formas modales, el MAC permite identificar y validar modos propios con mayor precisión. Una adecuada optimización de la recolección y tratamiento de datos con respecto al MAC puede llevar a una identificación más precisa de posibles defectos o degradaciones en un sistema, lo que a su vez puede traducirse en ahorros significativos en términos de tiempo y costos asociados a mantenimientos o reparaciones no previstos.

El valor real de este proyecto radica en la intersección entre la necesidad industrial de realizar inspecciones rápidas y precisas, y la capacidad tecnológica de procesar grandes volúmenes de datos de manera eficiente. Al desarrollar un algoritmo que facilite la interpretación y análisis de los modos de ondas, estamos eliminando barreras entre la recolección de datos y la acción,

asegurando que las decisiones se basen en información confiable y actualizada.

En conclusión, el presente proyecto no solo busca optimizar un proceso esencial dentro de los END, sino que también se erige como una respuesta a la creciente demanda de herramientas capaces de manejar y simplificar grandes conjuntos de datos. Al focalizarse en la precisión, este trabajo se posiciona en la vanguardia de la intersección entre tecnología y mantenimiento estructural, buscando siempre la seguridad, durabilidad y optimización de recursos en el ámbito de la ingeniería.

3. OBJETIVOS

3.1. Objetivo general

Desarrollar un algoritmo que permita agrupar los modos de ondas en placas a partir de la información de las curvas de dispersión.

3.2. Objetivos Específicos

- Definir y calcular las propiedades mecánicas y geométricas del material a estudiar, estableciendo los parámetros iniciales necesarios para el desarrollo del modelo computacional en MATLAB.
- Implementar un modelo computacional en MATLAB para la obtención de perfiles de onda y curvas de dispersión de ondas guiadas.
- Clasificar y visualizar los modos de propagación de ondas guiadas, generando curvas de dispersión que sean claras, precisas y visualmente coherentes.

4. MARCO TEÓRICO

En este capítulo se ha recopilado información sobre diversas investigaciones que abordan conceptos claves relacionados de como diferentes autores abordan conceptos clave relacionados con la clasificación de modos de ondas guiadas y el análisis de curvas de dispersión en el contexto de los ensayos no destructivos. En este sentido, se exploran los fundamentos teóricos de las ondas guiadas, las propiedades de los modos de propagación y la importancia de las curvas de dispersión como herramienta para el análisis de estructuras

4.1. Antecedente

En el ámbito del análisis de modos de propagación de ondas guiadas, el desarrollo de un algoritmo preciso y eficiente es de gran importancia. Gracias al avance en el conocimiento de la física de las ondas guiadas y los métodos de análisis, se han sentado las bases para explorar nuevas posibilidades en la clasificación de estos modos. Sin embargo, es fundamental contar con antecedentes y estudios previos que nos brinden referencias sobre cómo se ha abordado esta temática en el pasado. Estos antecedentes nos permitirán establecer un punto de partida sólido para nuestro propio algoritmo, conocer lo que se ha logrado hasta el momento y explorar las oportunidades que se presentan en este campo de investigación.

En la industria existen diversos métodos de inspección mediante ondas guiadas, como es el caso de QUALICONTROL el cual es un sistema de inspección de tuberías mediante ondas guiadas y es así como “Mediante esta técnica se obtienen datos confiables los cuales ponen en evidencia aquellos sitios donde existen discontinuidades,” (QUALICONTROL, año desconocido), se puede complementar mediante otras herramientas para dimensionar fácilmente las discontinuidades encontradas, y con esto tener un margen de comparación sobre el uso de las ondas guiadas con el proyecto.

En *Ultrasonic Waves in Solid Media* (Rose, J. L. 1999) se examina el empleo de ondas guiadas por ultrasonido para la identificación de fallos en tuberías de gas en Corea. Se indica que estas ondas se propagan siguiendo la geometría de la tubería y comprenden tanto ondas longitudinales como transversales. Además, se enfatiza que las ondas guiadas presentan características notables en comparación con las ondas ultrasónicas generales, como la existencia de un modo de onda infinito en un amplio espectro de frecuencias y una velocidad de propagación dependiente de la frecuencia y el grosor de la tubería. "Las curvas de dispersión para las tuberías de gas fueron calculadas por un programa de Matlab para desarrollar una tecnología de prueba no destructiva de largo alcance para detectar fallas en tuberías de gas de la ciudad." (Rose, J. L. 1999).

4.2. Referentes teóricos

El desarrollo de este proyecto requiere una comprensión clara de los conceptos fundamentales involucrados. En este sentido, es crucial establecer una base teórica sólida desde la cual se abordará el problema. Dado que estamos desarrollando un algoritmo para clasificar los modos de propagación de ondas guiadas, es esencial tener un entendimiento profundo de la física subyacente de las ondas acústicas y su interacción con diversos medios. Esto implica explorar las propiedades y comportamientos de los distintos modos de onda, tanto longitudinales como transversales, y cómo estos se ven influenciados por factores como la geometría del medio, las características del material y las condiciones de frontera. Además, se requiere una revisión detallada de las metodologías existentes en el análisis de ondas, incluyendo técnicas avanzadas como la Transformada de Fourier y la Transformada Wavelet, que son fundamentales para descomponer y analizar señales complejas. Este conocimiento teórico no solo proporcionará la base para el desarrollo del algoritmo, sino que también permitirá comprender y mitigar los

desafíos inherentes a la clasificación precisa de modos de onda en prácticas de inspección no destructiva y otros campos aplicados.

4.2.1. Curvas de dispersión

Las curvas de dispersión representan la relación entre la frecuencia de una onda y su número de onda (o velocidad de fase) en estructuras delimitadas, como tuberías o placas. Estas curvas son esenciales en el estudio de ondas guiadas, ya que describen cómo varía la velocidad de propagación de una onda específica con respecto a su frecuencia. La dispersión ocurre cuando diferentes frecuencias de una onda viajan a diferentes velocidades, lo que puede ser influenciado por características como el grosor o material de la estructura. Idzi y boris dicen que " las curvas de dispersión permiten describir, dada una excitación aplicada sobre la estructura en análisis, en qué ondas modales dicha excitación se va a descomponer"

4.2.2. Perfil de ondas

Los perfiles de onda se refieren a las características específicas de las ondas guiadas que se propagan a través de un medio delimitado, como una tubería o una placa. A diferencia de los modos de onda, que describen patrones de desplazamiento y deformación, los perfiles de onda se enfocan en cómo varían parámetros como la velocidad y la amplitud a lo largo del medio de propagación.

Las ondas guiadas utilizan las fronteras del medio como guía, lo que permite que estas ondas viajen distancias largas con una pérdida mínima de energía. Este fenómeno es particularmente útil en la inspección de estructuras como tuberías y placas, donde la

capacidad de detectar defectos a largas distancias y en áreas inaccesibles es crucial (Seco et al., 2002; Gravenkamp, Bause, & Song, 2014).

Un perfil de onda puede incluir diversas características como la velocidad de fase y la velocidad de grupo, las cuales dependen no solo del material del medio, sino también de su geometría, como el grosor de una placa o el diámetro de una tubería. La relación entre estas velocidades y las frecuencias de la onda se representa en las curvas de dispersión, que son esenciales para el análisis de la propagación de ondas guiadas (Gravenkamp, Bause, & Song, 2014).

La complejidad de los perfiles de onda radica en la existencia de múltiples modos de propagación en una estructura determinada. Estos modos pueden ser torsionales, longitudinales, cada uno con características específicas de propagación y sensibilidad a defectos. Por ejemplo, los modos torsionales son comúnmente utilizados en la inspección de tuberías debido a su capacidad de propagarse axialmente con una alta sensibilidad a los cambios en la sección transversal de la tubería (Seco et al., 2002).

Figura 1 Diferentes perfiles de onda

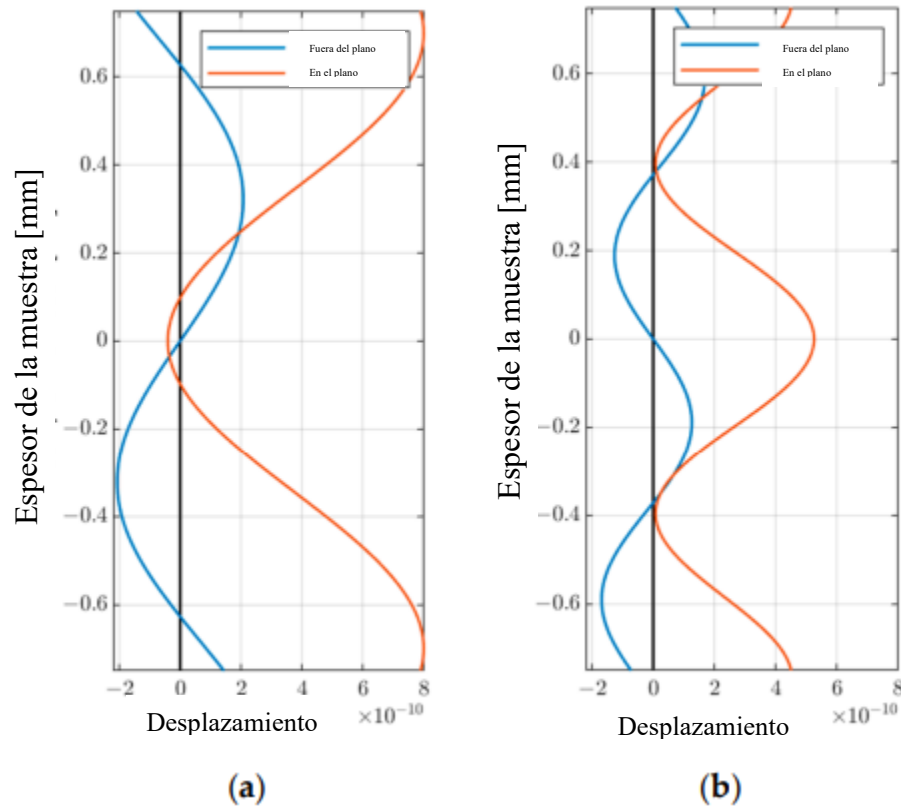


Figura Obtenida de Rekuvienė, R., Samaitis, V., Jankauskas, A., Sadaghiani, A. K., Saeidiharzand, S., & Kořar, A. (2024).

4.2.3. Ensayos no destructivos (END)

Son tćnicas de inspecci3n que permiten examinar y evaluar la integridad y propiedades de un material, componente o estructura sin causar dańo o alterar su funcionalidad. Estos ensayos son esenciales en industrias como la aeron3utica, nuclear, petroquímica y de construcci3n, ya que facilitan la detecci3n temprana de defectos o anomalías, garantizando la seguridad y prolongando la vida 3til de los componentes. Los mćtodos de END incluyen tćnicas como ultrasonido, radiografía, termografía, corrientes inducidas, entre otros.

4.2.4. An3lisis Modal

Es una tćcnica utilizada para determinar las frecuencias naturales, modos de perfiles de

onda y factores de amortiguamiento de una estructura o sistema. En el ámbito de la inspección no destructiva y la dinámica de estructuras, el análisis modal es esencial para comprender cómo se comportará un sistema bajo diferentes condiciones de excitación. La identificación precisa de los modos de perfiles de onda permite diseñar, optimizar y diagnosticar sistemas, garantizando su funcionamiento seguro y eficiente. Específicamente en contextos como el estudio de ondas guiadas, el análisis modal puede ayudar a discernir y clasificar distintos modos de propagación.

4.2.5. Generación y propagación de señales ultrasónicas

Seco y Jiménez (2012) exploran la generación y propagación de señales ultrasónicas en guías de onda cilíndricas, como tuberías, varillas y cables. Estas estructuras son esenciales en diversas aplicaciones científicas y de ingeniería, incluyendo la evaluación no destructiva (NDE) de la salud estructural de elementos de ingeniería civil, sensores de desplazamiento lineal y la evaluación de propiedades de materiales. Para abordar estas aplicaciones, es fundamental contar con modelos cuantitativos precisos que permitan entender los procesos de generación, propagación y recepción de señales ultrasónicas en las guías de onda.

La principal contribución de su trabajo radica en la simulación numérica del comportamiento ultrasónico de guías de onda cilíndricas, basada en la teoría de Pochhammer-Chree. Este enfoque incluye el ensamblaje de la matriz de descripción de la guía de onda, la trazabilidad de las curvas de frecuencia-número de onda, el cálculo de las formas de los modos y el análisis modal para determinar la respuesta de la guía de onda a excitaciones externas y la propagación dispersiva de señales. Como resultado, desarrollaron el software PCDISP, escrito en Matlab, que facilita la solución de las ecuaciones de Pochhammer-Chree y el modelado de la propagación de ondas elásticas en

guías de onda cilíndricas.

Figura 2 Ilustración del algoritmo de rastreo de raíces utilizado por PCDISP.

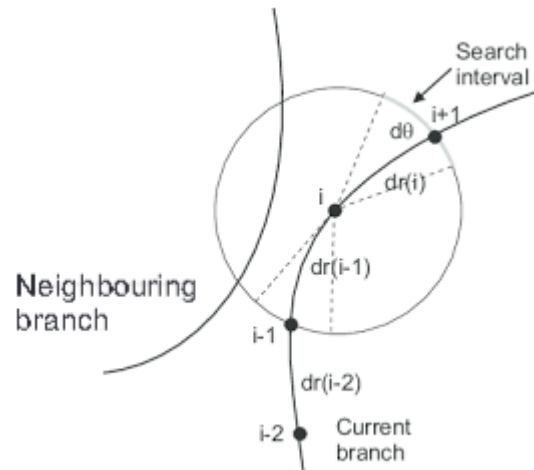


Figura obtenida de Seco, F., & Jiménez, A. R. (2012). Modelling the Generation and Propagation of Ultrasonic Signals in Cylindrical Waveguides

Para clasificar las soluciones, Seco y Jiménez utilizan un algoritmo de rastreo de raíces, ilustrado en la Figura 2. Este algoritmo se basa en la distancia entre modos y la tendencia de las curvas en puntos previos. La figura muestra cómo el algoritmo identifica y sigue las raíces de las ecuaciones características, ajustando el intervalo de búsqueda para asegurar la precisión y continuidad de las curvas de dispersión, evitando cruces entre modos adyacentes.

En la Figura 3(a) y 3(b) se presentan las velocidades de fase y grupo de los primeros modos axisimétricos de la guía de ondas. Estas figuras demuestran cómo el método permite clasificar y seguir de manera precisa los modos de propagación en función de la frecuencia, proporcionando información detallada sobre la dispersión y las velocidades a las que se propagan las señales ultrasónicas.

Figura 3(a) Fase y (b) velocidades de grupo de los primeros modos axisimétricos de la guía de ondas.

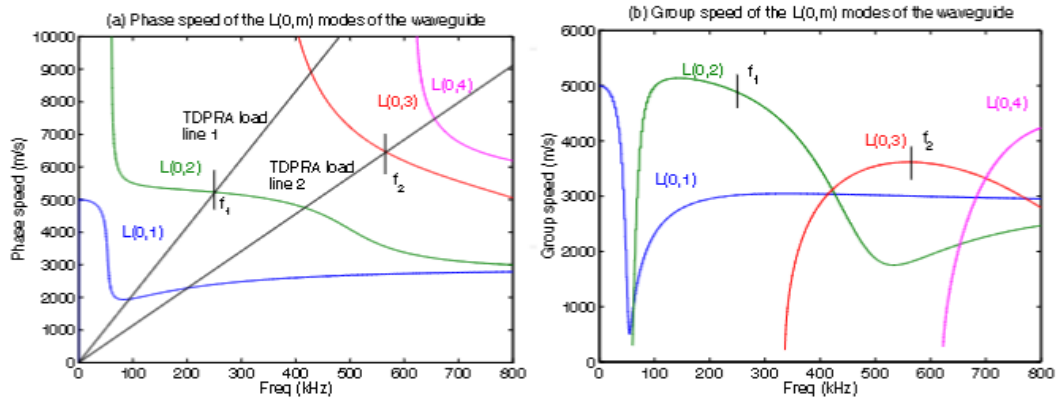


Figura obtenida de Seco, F., & Jiménez, A. R. (2012). Modelling the Generation and Propagation of Ultrasonic Signals in Cylindrical Waveguides

El estudio de Seco y Jiménez (2012) proporciona una base sólida para comprender y modelar las dinámicas complejas involucradas en la generación y propagación de señales ultrasónicas en estructuras cilíndricas. El uso del algoritmo de rastreo de raíces y la representación gráfica de las velocidades de fase y grupo permiten una clasificación precisa de los modos, facilitando un análisis detallado de la propagación de ondas en guías de onda cilíndricas, aspecto esencial en la ingeniería mecánica y la evaluación no destructiva.

4.2.6. Método Scaled Boundary Finite Element Method (SBFEM)

El método Scaled Boundary Finite Element Method (SBFEM) combina las ventajas de dos técnicas conocidas: el método de elementos finitos (FEM) y el método de elementos de contorno (BEM). A diferencia de otros métodos, el SBFEM solo necesita dividir en partes el contorno del área a estudiar, lo que simplifica y agiliza los cálculos. Este método es especialmente útil para analizar cómo se propagan las ondas y cómo se comportan las estructuras en situaciones donde se extienden al infinito, como en el análisis de terremotos. El SBFEM es muy eficiente para modelar la aparición y crecimiento de grietas en

materiales como el concreto cuando están sometidos a cargas repetitivas. Utiliza leyes cohesivas para simular cómo se distribuyen las tensiones alrededor de las grietas, permitiendo predecir con precisión cómo y hacia dónde se propagarán. Esto es útil para diseñar estructuras más seguras y duraderas. Estudios recientes han demostrado que este método proporciona resultados que coinciden bien con experimentos reales, validando su efectividad y precisión (Alrayes, Könke, Ooi, & Hamdia, 2023).

Una de las características clave del SBFEM es su capacidad para calcular con precisión las tensiones en la punta de la grieta utilizando ecuaciones específicas. Por ejemplo, el campo de tensiones cerca de la punta de una grieta puede expresarse como:

$$\sigma(\xi, \eta) = \sum_{i=1}^N \xi^{-\lambda_i} (\lambda_i [B_1(\eta)] + [B_2(\eta)]) \{\phi_i\}$$

donde ξ y η son coordenadas locales, λ_i son los exponentes asociados a los modos de desplazamiento y ϕ_i son los vectores de los modos propios. Esta ecuación permite determinar la distribución de tensiones y, por tanto, prever la propagación de las grietas bajo diferentes condiciones de carga (Wolf & Song, 1998).

En estudios recientes, el SBFEM ha sido utilizado con éxito para calcular curvas de dispersión de guías de ondas elásticas, demostrando su precisión en la caracterización de materiales y en la evaluación de estructuras complejas (Gravenkamp, Bause, & Song, 2014). La capacidad del SBFEM para manejar modos de propagación de ondas en guías cilíndricas y la solución eficiente de problemas de valores propios lo hacen una herramienta valiosa en diversas aplicaciones de la ingeniería.

5. DISEÑO METODOLÓGICO

En este capítulo se describe de forma general, la metodología utilizada para el desarrollo y

ejecución de este trabajo de investigación, con énfasis en las estrategias.

El diseño metodológico se estableció con base en el plan para el diseño sistemático de proyectos propuesto por John K. Gershenson, Ph.D. Profesor de ingeniería mecánica en la universidad tecnológica de Michigan. El plan consta de 5 diferentes fases de diseño, pre-fase A, donde se analizará la propuesta de diseño, la fase A, en la que se realizara el diseño conceptual del sistema en este caso del algoritmo, la fase B, donde establecerán los parámetros que serán necesarios que cumpla el algoritmo, la fase C, donde se optimizaran los previos parámetros y se realizara un diseño en detalle, y por último la fase D que estará compuesta por el desarrollo y prueba del algoritmo.

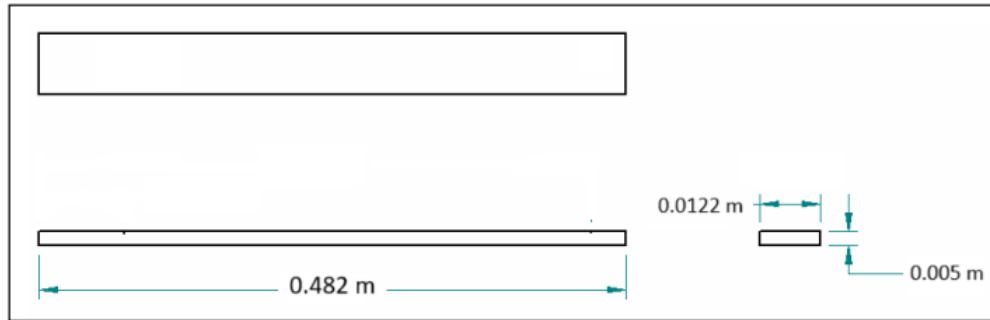
Teniendo esto en cuenta la forma de realizar el proyecto será el siguiente:

5.1. Pre-fase A: Análisis de la Propuesta de Diseño:

Para desarrollar una propuesta de diseño es crucial tener presente el problema central de este estudio, que es mejorar la precisión en la identificación de modos de propagación de ondas guiadas en materiales compuestos. Este desafío es especialmente relevante en campos como la ingeniería estructural y el monitoreo de la salud de infraestructuras, donde las técnicas existentes a menudo fallan en proporcionar detección temprana de fallos bajo condiciones operativas adversas.

Los objetivos del proyecto guían las decisiones metodológicas de esta fase, centradas en el desarrollo de un algoritmo que mejore la precisión en la clasificación de estos modos.

Figura 4 Dimensiones de la placa de acero 1020



El análisis de las propiedades mecánicas y geométricas del acero AISI 1020 es fundamental para el desarrollo preciso de modelos computacionales que simulan fenómenos físicos complejos. Este proyecto se enfoca en una placa de acero AISI 1020, un material comúnmente utilizado en la industria debido a su combinación de resistencia y ductilidad. La correcta definición y cálculo de propiedades como el módulo de elasticidad, la densidad y el coeficiente de Poisson son esenciales para establecer los parámetros iniciales del modelo. Estos parámetros proporcionan la base necesaria para simular con precisión el comportamiento dinámico del material bajo diversas condiciones de carga.

Tabla 1 Propiedades acero 1020

Propiedad	Símbolo	Magnitud	Unidades	Descripción
Módulo de Elasticidad	E	186×10^9	Pa	Indica la rigidez del material.
Coefficiente de Poisson	ν	0.29	-	Relación entre la deformación lateral y axial.
Densidad	ρ	7870	kg/m^3	Influye en la masa de la placa y su respuesta dinámica.

Espesor de la Placa	D	1	mm	Grosor de la placa utilizado en el análisis modal.
Módulo de Cortante	G	72×10^9	Pa	Determina la respuesta del material a esfuerzos cortantes.
Módulo de Volumen	K	154.3×10^9	Pa	Relaciona el cambio de volumen del material bajo presión.
Frecuencias Naturales	-	Variable	Hz	Determinadas por análisis modal.
perfiles de onda	-	Variable	-	Modos fundamental y superiores.

La caracterización comienza con la definición de parámetros fundamentales del material:

- E representa el módulo de elasticidad, que caracteriza la rigidez del acero 1020 bajo esfuerzos de tensión.
- ν es el coeficiente de Poisson que describe la relación entre las deformaciones transversales y longitudinales ante una carga.
- ρ define la densidad del material, lo que es crucial para entender la propagación de la masa y las ondas a través de él.

La función $prop(E, \nu, \rho)$ calcula propiedades mecánicas vitales como el módulo Bulk (K), los parámetros de Láme (L y G), y las velocidades de ondas P y S (V_p y V_s), que son determinantes

para predecir cómo las ondas se propagarán a través del acero. El tensor de elasticidad (C) es esencial para modelar las respuestas del material bajo cargas elásticas diversas.

5.1.1. Condiciones de Trabajo del Algoritmo:

El rendimiento del algoritmo se contextualiza a través de las condiciones de trabajo preestablecidas:

- FM1 establece la frecuencia máxima analizada, un parámetro clave para limitar el espectro de estudio de las ondas.
- NP1 define el número de puntos o divisiones en el análisis, influenciando la granularidad de las curvas de dispersión y la precisión del modelo.
- TP1 corresponde al período de muestreo asociado a la frecuencia máxima y el número de puntos, ajustando el algoritmo a las oscilaciones esperadas del material.
- O1 y J1 son parámetros técnicos del algoritmo que posiblemente relacionan el orden de los polinomios usados en aproximaciones y el factor de escala geométrico asociado al espesor del material, respectivamente.

Con estas configuraciones, el algoritmo está preparado para realizar simulaciones que reflejen con precisión el comportamiento del acero 1020 bajo cargas dinámicas.

Tabla 2 Condiciones de trabajo

Parámetro	Símbolo	Magnitud	Unidades	Descripción
Frecuencia Máxima Analizada	<i>FM</i>	Variable	Hz	Límite superior del rango de frecuencias analizadas.

Número de Puntos Analizados	NP	Variable	-	Cantidad de divisiones en el análisis, influye en la granularidad del modelo.
Período de Muestreo	TP	Variable	s	Intervalo de tiempo entre mediciones sucesivas.
Orden de Polinomios	O	Variable	-	Nivel de detalle en aproximaciones matemáticas.
Factor de Escala Geométrica	J	Variable	-	Influencia del espesor de la placa en el análisis.

5.2. Fase A: Diseño Conceptual del Algoritmo:

En esta fase del diseño metodológico, nos centramos en la conceptualización y estructuración inicial del algoritmo para la clasificación de modos de propagación de ondas guiadas. Optamos por utilizar MATLAB debido a su entorno altamente especializado en la simulación numérica y en la ejecución de algoritmos intensivos de procesamiento de señales. Este entorno es particularmente adecuado para manejar las operaciones intensivas de cálculo que nuestro algoritmo requiere, especialmente en lo que respecta a la manipulación y análisis de grandes volúmenes de datos de ondas guiadas.

5.2.1. Matrices

5.2.1.1. Cálculo de Matrices de Rigidez

La evaluación de la respuesta dinámica del acero 1020, y en particular la propagación de ondas guiadas en este material, requiere la construcción de matrices de rigidez y masa. Estas matrices son componentes esenciales del modelo de elementos finitos utilizado para simular el comportamiento de las ondas dentro del material.

Figura 5 Matriz de rigidez

Matrices de rigidez:

$$[E0, E1, E2, EM] = k_eig(P, J1, O1, CC);$$

Mediante la llamada a la función `k_eig(P, J1, O1, CC)`, se calculan cuatro matrices distintas:

- E0: La matriz de rigidez en el plano, que relaciona los desplazamientos en el elemento con las fuerzas aplicadas. Esta matriz es central para entender cómo los desplazamientos internos del elemento afectan su estado de tensión y deformación.
- E1: La matriz de acoplamiento, que interconecta los desplazamientos y las rotaciones del elemento. Esta matriz es crucial cuando los modos de onda involucran tanto movimientos traslacionales como rotacionales del material.
- E2: La matriz de rigidez de rotación, vinculada directamente con los momentos rotacionales en el elemento finito. Aporta al análisis un componente adicional que es necesario para capturar la totalidad del espectro de comportamientos mecánicos del material bajo estudio.
- EM: La matriz de masa, que representa la inercia del material y es fundamental en la formulación de los problemas de ondas guiadas. Esta matriz permite evaluar cómo la densidad y la geometría del material influyen en la propagación

de ondas y en perfiles de onda.

5.2.1.2. Matriz T de Transformación

Posteriormente, el algoritmo establece una matriz de transformación utilizando la función $mT(s1)$, donde $s1$ es el tamaño de la matriz $E0$. Esta matriz de transformación M es crucial para redefinir las matrices de rigidez en un nuevo sistema de coordenadas, facilitando la solución del problema de autovalores que surge en la búsqueda de perfiles de onda y frecuencias naturales del material.

Figura 6 Matriz T de Transformación

Matriz T de transformación:

```
s1 = size(E0);  
M = mT(s1);  
E0T=(M'*E0*M);  
E1T=(M'*E1*M);  
E2T=(M'*E2*M);  
EMT=(M'*EM*M);
```

Las operaciones subsiguientes realizan transformaciones de similaridad en las matrices $E0$, $E1$, $E2$ y EM , con el fin de alinearlas con el marco más propicio para el análisis. El resultado son las matrices transformadas $E0T$, $E1T$, $E2T$, y EMT , las cuales representan la reconfiguración de las matrices de rigidez originales a través de la matriz M . Este paso es instrumental en el análisis de ondas guiadas, ya que simplifica el cálculo de los modos de propagación de las ondas y su interacción con las propiedades del material. En el contexto de la ingeniería estructural y los ensayos no destructivos, estas matrices transformadas facilitan la identificación de fenómenos como la dispersión de ondas en el material, un aspecto crítico en la evaluación de la integridad estructural y la detección

de defectos. La precisión de este enfoque metodológico permite simular de manera precisa cómo las ondas se propagan a través del acero 1020, contribuyendo significativamente al diseño y análisis basado en el rendimiento.

5.2.2. Autovalores

El análisis dinámico de estructuras es una disciplina central en la ingeniería que requiere la identificación precisa de las frecuencias naturales y los perfiles de onda. El código presentado aborda este desafío al resolver un problema de valores propios generalizado, donde se busca determinar los modos naturales de la estructura de acero 1020 a través de la relación $Ax = \lambda Bx$.

Figura 7 Autovalores

Autovalores - Ax=yBx:

```
[v1,M7,Vp]=lapiz_eig(TP1,FM1,E0T,E1T,E2T,EMT);  
  
Nf = size(M7,2);  
Nm = size(M7,1); % Magnitud compleja de los valores propios
```

Mediante la función $[v1, M7] = \text{lapiz_eig}(TP1, FM1, E0T, E1T, E2T, EMT)$, el algoritmo calcula estos autovalores (λ) (lambda) en este contexto, está relacionado con las frecuencias naturales de la estructura y los autovectores (x), que representa los perfiles de onda de la estructura. En la matriz A están codificadas las propiedades físicas del material y la geometría de la estructura, mientras que la matriz B integra aspectos de la masa y la rigidez rotacional del sistema.

- v1: Este vector resultante encapsula las frecuencias angulares ω (omega), las cuales son el producto de 2π por las frecuencias naturales (f). Estas frecuencias son críticas para el análisis, ya que representan las condiciones bajo las cuales la estructura puede entrar en

resonancia.

- $M7$: Esta matriz contiene los vectores propios correspondientes a cada frecuencia angular calculada. Cada columna de $M7$ representa un perfil de onda posible para la estructura bajo estudio.

El uso de `lapiz_eig` ilustra una aplicación especializada de algoritmos matemáticos avanzados para resolver problemas de autovalores en MATLAB. El nombre tiene una relación con el "método del lápiz", una técnica analítica que transforma problemas de autovalores generalizados a una forma estándar más manejable. Este enfoque es particularmente útil en sistemas complejos donde se requiere una gran precisión, como es el caso en la caracterización de las propiedades dinámicas de las ondas guiadas en el material en estudio.

5.2.3. Filtrado de perfiles de onda.

La matriz ' $M7$ ', que contiene los vectores propios resultantes del cálculo de autovalores, es sometida a un proceso de filtrado que distingue entre modos atenuados y no atenuados como se muestra en la figura.

Figura 8 Filtro de perfiles de onda

Filtro:

```
[s1,s2]=size(M7);           % Dimensiones de la matriz [r,c]
mr0 = zeros(s1,s2);
mr1 = zeros(s1,s2);
mr2 = zeros(s1,s2);
for j=1:s2
    for i=1:s1
        if (imag(M7(i,j))==0) && (real(M7(i,j))>0)
            %if (imag(M7(i,j))==0)
                mr0(i,j) = real(M7(i,j));           % Identifica valores reales
            elseif (imag(M7(i,j))~=0) && (real(M7(i,j))>0) % Identifica valores complejos
                mr1(i,j) = real(M7(i,j));           % Identifica parte real
                mr2(i,j) = imag(M7(i,j));           % Identifica parte imaginaria
            end
        end
    end
end
mr0(mr0==0)=nan;           % Si valor es igual a 0 --> valor = NaN
freq = 2*pi*v1;
```

- `[s1, s2] = size(M7)`: Se establecen las dimensiones de la matriz de perfiles de onda, donde `s1` indica la cantidad de modos y `s2` el número de frecuencias analizadas.
- Se inicializan tres matrices, `mr0`, `mr1` y `mr2`, cada una con las mismas dimensiones que `M7`. Estas matrices se utilizarán para clasificar los modos según la presencia o ausencia de atenuación.

El bucle for que sigue realiza la clasificación:

- `if (imag(M7(i,j))==0)&(real(M7(i,j))>0)`: Se verifica si el perfil de onda es no atenuado, esto es, si tiene un componente real positivo y un componente imaginario nulo. Estos modos son almacenados en `mr0`.
- `elseif (imag(M7(i,j))~=0)&(real(M7(i,j))>0)`: Si el modo tiene tanto un componente real positivo como un componente imaginario no nulo, indicando atenuación, estos son registrados en `mr1` para la parte real y `mr2` para la parte imaginaria.

Después de iterar a través de la matriz `M7`, los elementos de `mr0` que son cero se reemplazan por NaN para evitar confusión en el análisis posterior, ya que los ceros podrían interpretarse erróneamente como datos válidos.

Cálculo de Frecuencias Físicas:

$freq = 2 * \pi * v1;$: Esta expresión transforma las frecuencias angulares almacenadas en `v1` a frecuencias físicas en hercios, que son más intuitivas y utilizadas comúnmente en la interpretación y el análisis de los resultados.

Las matrices resultantes `mr0`, `mr1` y `mr2` ahora contienen los perfiles de onda clasificados que servirán como base para el siguiente paso en el análisis, como puede ser la visualización de las curvas de dispersión de ondas. Este filtrado asegura que los modos irrelevantes o que no contribuyen al análisis deseado sean excluidos de las fases subsecuentes, mejorando así la calidad y la precisión de los resultados finales del estudio.

5.2.4. Pre-ordenamiento

El preordenamiento de modos es una etapa crucial en el análisis modal, ya que permite organizar los datos del perfil de onda de manera coherente para facilitar el análisis posterior. A continuación, se describe el proceso de preprocesamiento y ordenamiento de los perfiles de onda utilizando el código mejorado. Este proceso asegura que las frecuencias y modos estén correctamente alineados y preparados para un análisis detallado.

Preprocesamiento de perfiles de onda

El primer paso en el preordenamiento de los perfiles de onda consiste en determinar las dimensiones de la matriz `mr0`, que contiene los valores reales de los modos tras el filtrado. Se extrae la última columna de esta matriz, que representa los perfiles de onda a la frecuencia más alta muestreada. Luego, se identifican los valores `NaN` en esta columna para distinguir

entre los modos válidos y los datos faltantes.

Figura 9 Preordenamiento de modos

Preordenamiento de modos:

```
[s1,s2] = size(mr0);           % Tamaño de matriz de valores reales
M_end=mr0(:,end);           % Columna final de la matriz
n_logic=isnan(M_end);       % Identifica valores NaN
[n_cuentas,valor]=groupcounts(n_logic); % Identifica número de valores NaN y valores no NaN
n_modos=n_cuentas(1);       % Determina número de valores no NaN
```

Se inicializan varias matrices que se utilizarán para almacenar los modos organizados y otros parámetros calculados, como `v`, `Vp_aux`, `Vp_n2`, `MAC`, `Dif`, `Dif_n` y `Crit`.

Figura 10 Inicialización de Matrices y Vectores para Análisis Modal en MATLAB

```
v=zeros(n_modos,s2);
Vp_aux = zeros(n_modos,s1,s2);
Vp_n2 = zeros(n_modos,s1,s2+1);
MAC = zeros(n_modos,n_modos,s2-1);
Dif = zeros(n_modos,n_modos,s2-1);
Dif_n = zeros(n_modos,n_modos,s2-1);
Crit = zeros(n_modos,n_modos,s2-1);
v = zeros(n_modos,s2);
ord = zeros(n_modos,s2);
```

Para cada columna de `mr0`, se identifican y ordenan los valores no `NaN` en orden descendente. Este proceso asegura que los modos válidos se agrupen y ordenen correctamente.

Figura 11 Bucle de Preprocesamiento y Ordenamiento de perfiles de onda en MATLAB

```
for i = 1:1:s2 % Contador en reversa
    v_aux1=mr0(:,i);
    v_logic1=isnan(v_aux1);
    indices = find(v_logic1 == 0);
    Vp_aux(indices,:,i) = Vp(indices,1:s1,i);
    Vp_aux2 = Vp_aux(v_logic1==0,:,i);
    lvp = size(Vp_aux2,1);
    Vp_n(1:lvp,:,i) = Vp_aux2(1:lvp,:);

    [n_count,n_value]=groupcounts(v_logic1);
    if n_count(1)==n_modos
        [v(:,i),ord(:,i)]=sort(v_aux1(v_logic1==0),'descend'); % Ordena los modos reales en orden descendente
    else
        [v_aux2,ord2]=sort(v_aux1(v_logic1==0),'descend'); % Ordena los modos
        l1=length(v_aux2);
        v(1:l1,i)=v_aux2(1:l1);
        ord(1:l1,i) = ord2(1:l1);
    end
    l2 = ord(ord(:,i)~=0,i);
    l3 = length(l2);
    Vp_n2(1:l3,:,i) = Vp_n(ord(1:l3,i),:,i);
    %v(v==0) = 1e7;
end
```

Se ajusta la matriz `v` para incluir una columna adicional, preparando los datos para el ajuste de modos. Luego, en un bucle adicional, se asegura que los modos estén alineados correctamente entre frecuencias adyacentes. Este ajuste es crucial para mantener la coherencia de los datos a lo largo de las frecuencias.

Figura 12 Ajuste y Ordenamiento de perfiles de onda a través de Frecuencias en MATLAB

```
v(:,Nf+1) = v(:,Nf);
for i = 1:1:s2
    nzj = nnz(v(:,i)); % Número de modos diferentes de 0
    nzk = nnz(v(:,i+1)); % Números de modos diferentes de 0
    if i > 1
        nzi = nnz(v(:,i-1));
    end
    dist = zeros(nzk,1);
    dist2 = zeros(nzk,1);
    dif = zeros(2,1);
    Pos_n = zeros(nzj,1);
    for j = 1:1:nzj
        for k = 1:1:nzk
            dist(k) = v(j,i) - v(k,i+1);
        end
        [~,min_4] = sort(abs(dist));
        for k = 1:1:2
            m = min_4(k);
            if i > 1
                dist2(m) = v(j,i-1) - v(j,i);
            else
                dist2(m) = 0;
            end
            dif(k) = dist(m) - dist2(m);
        end
        [~,Pos_p] = sort(abs(dif));
        Pos_n(j) = min_4(Pos_p(1));
        if j > 1 && ismember(Pos_n(j), Pos_n(1:j-1))
            Pos_n(j) = min_4(Pos_p(2));
        end
    end
    df = nzk - nzj;
    Pj = 1:1:nzk;
    Pd = setdiff(Pj,Pos_n);
    if nzj < nzk
        Pos_n(nzj+1:nzk) = Pd;
    end
    if nzj > nzk
        Pos_n(nzk+1:nzj) = [];
        Pd = setdiff(Pj,Pos_n);
    end
end
```

```

    if nzj > nzk && nzi < nzj
        if isempty(Pd)
            else
                Pos_n(nzi+1:nzj) = Pd;
            end
        end
        v(1:nzk,i+1:end) = v(Pos_n,i+1:end);
        Vp_n2(1:nzk, :, i+1:end) = Vp_n2(Pos_n, :, i+1:end);
    end
    v(v==1e7) = 0;
    v(:,Nf+1) = [];
    v(v==0)=nan;

```

Finalmente, se realiza una limpieza de la matriz `v`, sustituyendo valores específicos por `NaN` para mantener la coherencia de los datos.

Este preordenamiento de modos es crucial para alinear los datos de modo que se facilite su análisis comparativo. Asegura que las frecuencias más bajas se sitúen a la izquierda y las más altas a la derecha de la matriz `v`, lo que facilita la visualización y análisis posterior, como la determinación de curvas de dispersión y el cálculo de velocidades de fase y grupo.

5.2.5. Visualización grafica de los resultados de los perfiles de onda.

Tras el preordenamiento de los perfiles de onda, se lleva a cabo la fase de visualización, un componente crítico que permite la interpretación gráfica de la relación entre frecuencias y números de onda, así como las velocidades de fase y grupo. Este paso convierte los datos numéricos en representaciones visuales que son más fáciles de analizar y entender.

Curvas de Dispersión:

- **f vs k:** El primer gráfico creado con `plot(freq, v, '.')` dibuja un mapa de dispersión donde el eje horizontal representa las frecuencias (f) y el eje vertical los números de onda (K). Los límites de los ejes se ajustan de manera precisa con `xlim tight` y `ylim tight`, proporcionando una visión clara de la distribución de los datos sin espacios vacíos

excesivos que podrían distraer o disminuir la claridad del análisis.

Figura 13 Código de Visualización para Gráfico de 'f vs k'

```
plot(freq,v, '.')
xlim tight
ylim tight
xlabel('f')
ylabel('K')
title('f vs k')
```

- f vs C_p : El siguiente gráfico `plot(freq, v1./v, '.')` ofrece una visualización de la relación entre las frecuencias y las velocidades de fase (C_p). Aquí, los ejes se ajustan con `xlim tight` y un rango específico en `ylim` para enfocar el análisis en el intervalo de velocidades entre 0 y 8000, abarcando las velocidades de interés práctico.

Figura 14 Código de Visualización para Gráfico de 'f vs Cp'

```
plot(freq,v1./v, '.')
xlim tight
ylim ([0 8000])
xlabel('f')
ylabel('Cp')
title('f vs Cp')
```

- f vs C_g : Por último, el gráfico `plot(dfreq, dw./dk, '.')` ilustra cómo la velocidad de grupo (C_g) varía con la frecuencia. Antes de graficar, se calcula la diferencia en las frecuencias angulares y los números de onda para determinar el cambio en la velocidad de grupo. Se aplican ajustes similares en los ejes X e Y para mantener una coherencia visual y comparativa con el gráfico de velocidad de fase.

Figura 15 Código de Visualización para Gráfico de 'f vs Cg'

```
dfreq=freq(2:end);  
dw=diff(v1);  
dk=diff(v,1,2);  
plot(dfreq,dw./dk, '.')  
xlim tight  
ylim ([0 8000])  
xlabel('f')  
ylabel('Cg')  
title('f vs Cg')
```

5.2.6. Definición de funciones y resolución de valores propios

El algoritmo de clasificación de modos de propagación de ondas guiadas en placas de acero 1020 incorpora una serie de funciones especializadas en MATLAB que son fundamentales para calcular las propiedades mecánicas de los materiales y para abordar problemas de valores propios, los cuales son esenciales en la comprensión de la dinámica de ondas en estructuras.

- Cálculo de Propiedades Mecánicas del Material

La función prop es esencial para definir las propiedades elásticas fundamentales del material. Calcula el módulo Bulk (K), que representa la resistencia del material a los cambios de volumen bajo presión, y el primer parámetro de Láme (L), crucial para la descripción del comportamiento elástico. El módulo de Corte (G) es determinante en la respuesta del material frente a las fuerzas de cizallamiento. Estas propiedades se complementan con la velocidad de las ondas de compresión (P) y de cizallamiento (S), denotadas por V_p y V_s , respectivamente, y son críticas para el análisis de la propagación de ondas mecánicas.

Figura 16 Cálculo de Propiedades Elásticas y Velocidades de Onda (prop)

```
function [K,L,G,M,C,Vp,Vs] = prop(E,V,P)
K = E/(3*(1-2*V)); % módulo masivo [Pa]
L = E*V/((1+V)*(1-2*V)); % primer parámetro de Láme [Pa]
G = E/(2*(1+V)); % Módulo cortante [Pa]
M = E*(1-V)/((1+V)*(1-2*V)); % Módulo onda P [Pa]
C = E/((1+V)*(1-2*V))*[1-V V 0;V 1-V 0;0 0 (1-2*V)/2]; % Tensor de elasticidad [Pa]
Vp = sqrt(M/P); % velocidad de ondas presivas [m/s]
Vs = sqrt(G/P); % velocidad de ondas cortantes [m/s]
end
```

- Propiedades Elásticas Complejas

Mediante la función `prop_c`, se considera la atenuación de las ondas, un fenómeno físico crucial en materiales reales. Las velocidades de onda P y S ajustadas por la atenuación y los parámetros elásticos complejos reflejan la capacidad del material para disipar energía, lo que afecta directamente la interpretación de la respuesta de la estructura bajo carga dinámica.

Figura 17 Función para Propiedades Complejas de Onda (prop_c)

```
function [VpC,VsC,ErC,VrC,CC] = prop_c(P,Vp,Vs,La,Sa)
VpC = Vp/(1+1i*(La/(2*pi)));
VsC = Vs/(1+1i*(Sa/(2*pi)));
ErC = P*(VsC^2)*((3*VpC^2-4*VsC^2)/(VpC^2-VsC^2));
VrC = (1/2)*((VpC^2-2*VsC^2)/(VpC^2-VsC^2));
CC = ErC/((1+VrC)*(1-2*VrC))*[1-VrC VrC 0;VrC 1-VrC 0;0 0 (1-2*VrC)/2];
end
```

- Resolución de Problemas de Valores Propios

La función `k_eig` calcula las matrices de rigidez y masa del sistema. Estas matrices son utilizadas posteriormente en la función `lapiz_eig`, que resuelve el problema de valores propios generalizado $Ax = \lambda Bx$. Aquí, λ representa las frecuencias naturales de la estructura y x son los perfiles de onda. Esta solución es vital para predecir el comportamiento vibracional de la estructura y para determinar cómo las ondas se propagarán a través del acero 1020.

Figura 18 Función para Cálculo de Autovalores de Rigidez (k_{eig})

```
function [E0,E1,E2,EM] = k_eig(D,J,O,C)
syms x
b1=[0 0;0 1;1 0];
b2=[1 0;0 0;0 1];

[nn,ww]=GLL(O);
N=SF(nn);
D_N=(1/J)*diff(N);

sE0=0;
sE1=0;
sE2=0;
sEM=0;

B1=b1*N;
B2=b2*D_N;

fE0=B1'*C*B1*J;
fE1=B2'*C*B1*J;
fE2=B2'*C*B2*J;
fEM=D*N'*N*J;
for i=1:1:length(nn)
    sE0=sE0+(ww(i)*(subs(fE0,x,nn(i))));
    sE1=sE1+(ww(i)*(subs(fE1,x,nn(i))));
    sE2=sE2+(ww(i)*(subs(fE2,x,nn(i))));
    sEM=sEM+(ww(i)*(subs(fEM,x,nn(i))));
end

E0=double(sE0);
E1=double(sE1);
E2=double(sE2);
EM=double(sEM);
end
```

- Herramientas de Elementos Finitos

Las funciones SF, shfunc, y GLL proporcionan herramientas necesarias para la implementación del método de elementos finitos, una técnica ampliamente utilizada en simulaciones de física de ondas. SF genera una matriz de funciones de forma, shfunc define funciones de forma individuales, y GLL establece nodos y pesos para la integración numérica precisa dentro de este método.

Figura 19 Generador de Matriz de Forma (SF)

```
function M = SF(nodes)
for i=1:1:length(nodes)
    M1(:,i)=shfunc(i,nodes);
end
M2=M1(1)*eye(2);
for j=2:1:length(nodes)
    M3=M1(j)*eye(2);M2=[M2 M3];
end
M=M2;
end
```

- Matriz de Transformación

Por último, la función mT crea una matriz de transformación que se utiliza para simplificar el problema de valores propios, posiblemente convirtiendo el sistema a una base donde el cálculo de los perfiles de onda es más directo.

Cada una de estas funciones meticulosamente definidas en el código es un componente integral del análisis de ondas guiadas. Son herramientas matemáticas y físicas que, cuando se combinan, proporcionan un entendimiento profundo de las propiedades dinámicas de las ondas en las placas de acero, lo que es imprescindible en la ingeniería de materiales y en técnicas avanzadas de ensayo no destructivo. La capacidad de caracterizar con precisión la respuesta de un material a las ondas mecánicas y predecir su comportamiento bajo cargas dinámicas tiene implicaciones directas en la seguridad y la fiabilidad de las estructuras ingenieriles.

Figura 20 Función para Crear Matriz Transpuesta de Identidad (mT)

```
function M = mT(s1)
M=eye(s1(1));
for i = 1:2:s1(1)
    M(i,i)=1i;
end
end
```

5.3. Fase B: Definición de Parámetros del Algoritmo

En la Fase B del diseño metodológico, se definen los parámetros del algoritmo para optimizar su rendimiento y asegurar su aplicabilidad práctica. Cada parámetro fue cuidadosamente seleccionado para alinearse con los objetivos de la investigación y satisfacer las necesidades del campo de ensayos no destructivos.

- Tipo de Datos: Se optó por valores numéricos asociados a curvas de dispersión de ondas guiadas debido a su relevancia directa en el análisis de la propagación de ondas en materiales estructurales.
- Formatos de Entrada: La elección de formatos .csv, .txt, y .mat refleja la necesidad de versatilidad y compatibilidad con distintas fuentes de datos, asegurando una integración fluida en diversos flujos de trabajo de investigación y desarrollo.
- Compatibilidad de Sistema: Garantizar la funcionalidad en sistemas Windows y Linux y las últimas versiones de MATLAB permite que el algoritmo se mantenga relevante y accesible para una amplia base de usuarios.
- Salida del Algoritmo: La clasificación de modos y las visualizaciones gráficas forman el núcleo de la salida del algoritmo, proporcionando insights críticos para el análisis de integridad estructural y la identificación de defectos.
- Nivel de Precisión: Un umbral de precisión superior al 95% asegura que el algoritmo

sea confiable y sus resultados sean comparables o superiores a los métodos de referencia.

- **Visualizaciones:** Las gráficas detalladas son fundamentales para la interpretación de los resultados, permitiendo a los usuarios analizar visualmente las curvas de dispersión y los modos clasificados.
- **Robustez:** La habilidad para manejar datos imperfectos es esencial en aplicaciones del mundo real, donde los datos pueden estar sujetos a ruido y errores.
- **Adherencia a Estándares:** La conformidad con estándares industriales, aunque opcional, es un testamento de la validez del algoritmo en entornos profesionales y asegura que los resultados sean aceptados y valorados por la comunidad de ensayos no destructivos.

Cada parámetro está diseñado para consolidar el algoritmo como una herramienta robusta y confiable en la detección de defectos y la evaluación de la integridad estructural, demostrando su potencial para aportar valor significativo en la investigación y la práctica ingenieril.

Tabla 3 Requerimientos

Requerimiento	Descripción
Tipo de Datos	Numéricos relacionados con curvas de dispersión de ondas guiadas.
Formatos de Entrada	.csv, .txt, .mat
Compatibilidad	Windows, Linux y últimas tres versiones de MATLAB.
Salida	Clasificación de modos de ondas guiadas y visualizaciones gráficas.
Eficiencia	Procesamiento de conjuntos de datos estándar en < 8 minutos.
Precisión	>95% en comparación con benchmarks establecidos.
Visualizaciones	Gráficas de las curvas de dispersión y clasificación resultante.
Robustez	Manejo adecuado de datos con ruido, faltantes o inconsistencias.
Estándares	(Opcional) Cumplimiento de estándares de la industria para ensayos no destructivos.

5.4. Fase C: Optimización de Parámetros y Diseño Detallado

En la fase C del diseño metodológico, nos enfocamos en optimizar los parámetros y en el refinamiento del diseño del algoritmo. Cada aspecto del código se ha examinado y ajustado para mejorar la precisión y la robustez del sistema. A continuación, se presentan las optimizaciones clave implementadas, divididas en subsecciones según las partes relevantes del código.

5.4.1. Métodos de cálculo optimizados

- Cálculo de Propiedades Mecánicas (prop y prop_c):

Los cálculos de las propiedades mecánicas y complejas se realizan mediante expresiones cerradas, lo que garantiza que la computación sea directa y evita algoritmos iterativos que serían más lentos

- Cálculo de Matrices de Rigidez (k_eig):

La implementación de la integración numérica y el uso de la cuadratura Gauss-Lobatto-Legendre (GLL) en `k_eig` están aumentando la exactitud.

5.4.2. Optimización de la resolución de valores propios

Función de Transformación (mT) y Problema de Valores Propios (`lapiz_eig`):

La matriz de transformación utilizada en mT y el algoritmo para resolver el problema de valores propios en `lapiz_eig`. Esto incluye técnicas de transformación que simplifican el sistema de ecuaciones y permiten una solución más rápida de los valores propios.

Figura 21 Función para Cálculo de Eigenvalores Laplacianos (`lapiz_eig`)

```
function [v1,M7]=lapiz_eig(Tp,F,E0,E1,E2,EM)
n=round(F/Tp);
s0=size(E1);
M0=zeros(s0);
E1S=(E1'-E1)/(-1i);
M7 = zeros(s0(1)*2,n);
for i=1:1:n
    w=2*pi*(i*Tp);
    v1(1,i)=w;
    A=[M0,E2-(w^2)*EM;E2-(w^2)*EM,E1S];
    B=[E2-(w^2)*EM,M0;M0,-E0];
    [~,M4]=eig(A,B);
    v2=diag(M4);
    M7(:,i)=v2;
end
end
```

5.4.3. Visualización y representación de datos

Generación de Gráficos:

Las funciones de graficación proporcionan visualizaciones claras y detalladas de las curvas de dispersión. Las rutinas utilizan ajustes automáticos de escala y limitación de ejes para presentar los datos de manera efectiva, permitiendo a los usuarios una interpretación rápida y precisa.

5.4.4. Cumplimiento y adaptabilidad

Adherencia a estándares y formatos de datos, la capacidad del algoritmo para procesar múltiples formatos de datos y su conformidad con los estándares de la industria no destructiva garantizan que el sistema sea no solo adaptable sino también aplicable en entornos profesionales.

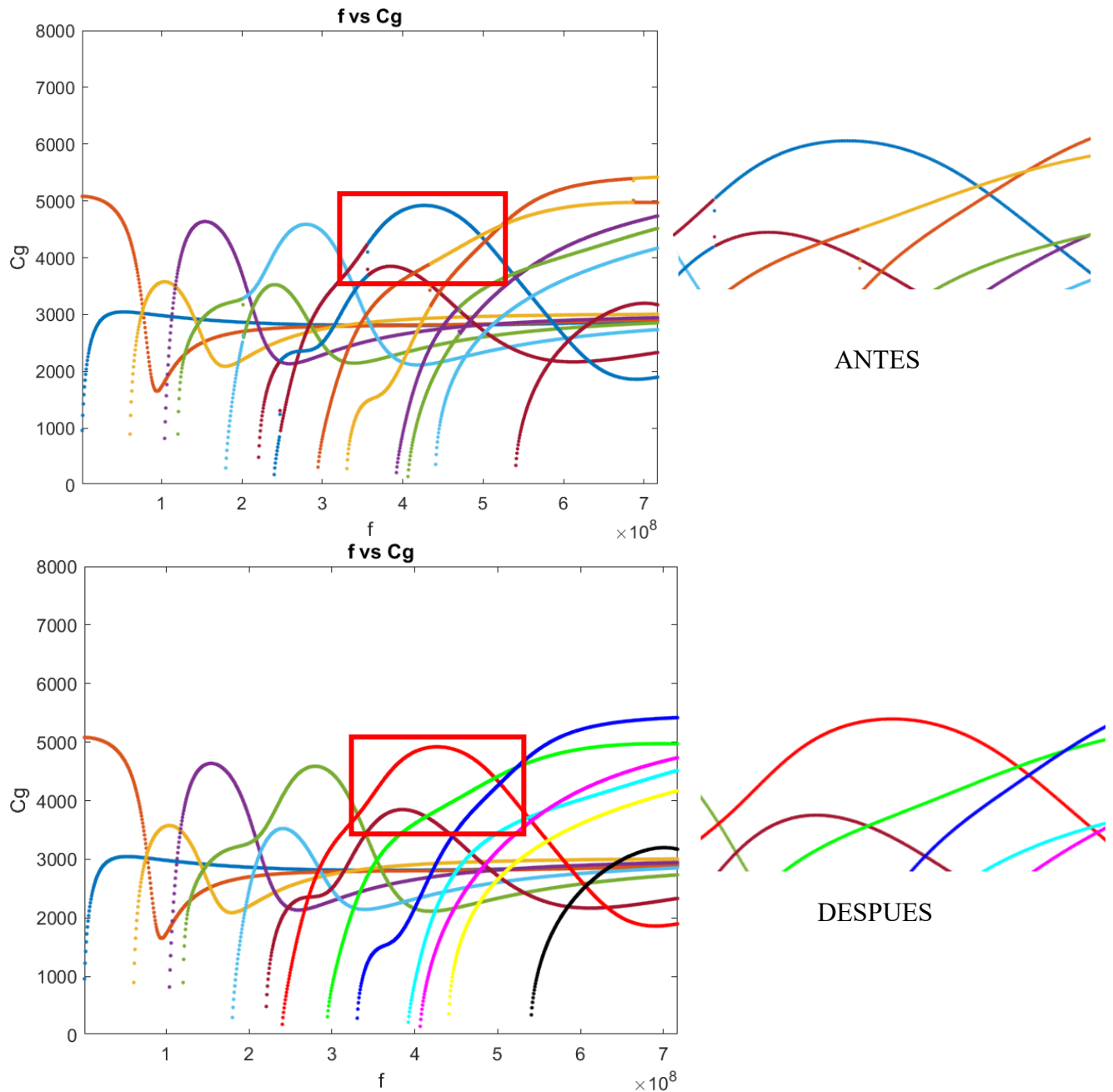
5.5. Fase D: Desarrollo y Pruebas del Algoritmo

En la Fase D, el enfoque se dirige hacia el desarrollo práctico del algoritmo y la realización de pruebas para validar su funcionamiento. A través de un análisis comparativo detallado, se demuestra cómo los ajustes implementados mejoran la precisión del algoritmo en el procesamiento y visualización de datos de ondas guiadas en estructuras laminares. Este proceso de validación es esencial para garantizar que el algoritmo cumple con los requisitos establecidos y proporciona resultados fiables y precisos en aplicaciones reales.

5.5.1. Pruebas del algoritmo

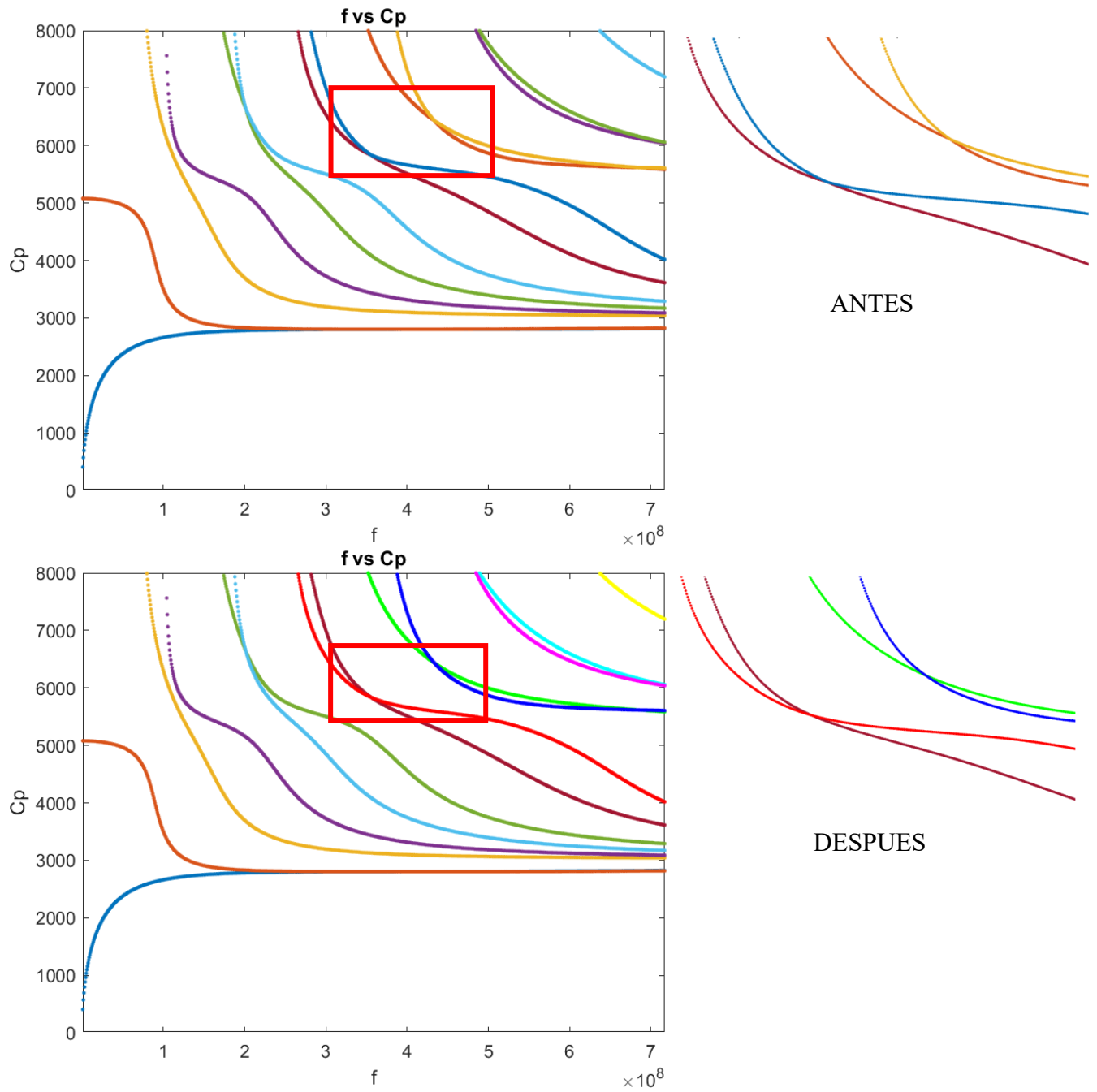
Enfocada en las pruebas del algoritmo, se evaluó la precisión en comparación con estándares de referencia temporal en el procesamiento de datos. La validación de resultados revela un avance en el rendimiento del algoritmo. A través de las pruebas, se confirma que las mejoras aplicadas han resultado en una clasificación y una mejor representación de los modos de ondas guiadas.

Figura 22 Comparativa de Velocidad de Grupo: Código Antiguo vs. Optimizado



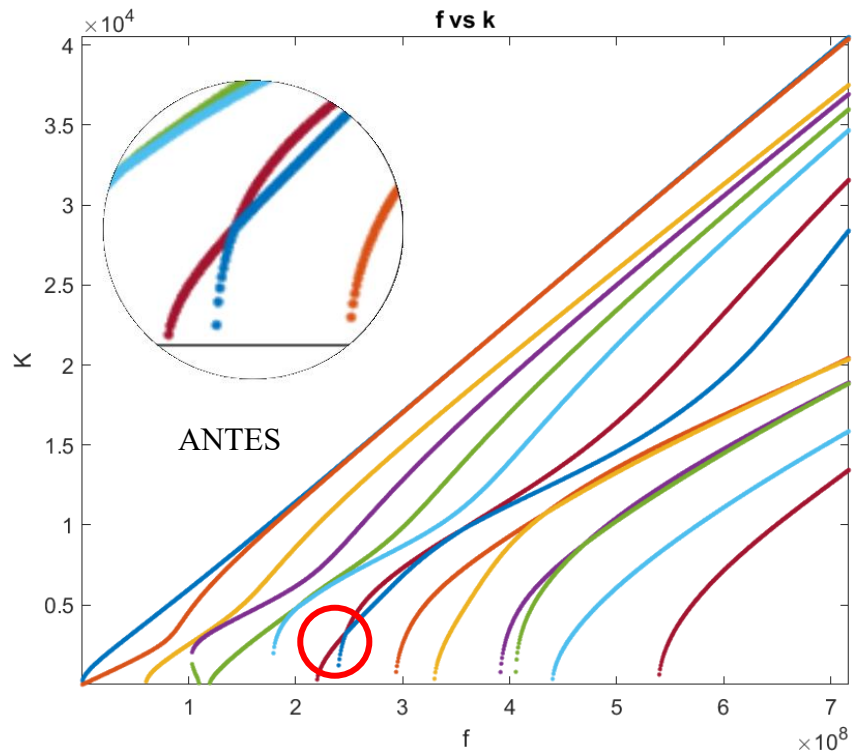
Los resultados como se pueden ver en la figura (22, 23 y 24) evidencian un progreso notable en la clasificación de los modos sobre el análisis modal de la placa de acero 1020 con un espesor de 1 mm. Como se ilustra en la figura (22), las optimizaciones implementadas han perfeccionado la capacidad de discernir y organizar los datos, lo que se refleja en una definición más exacta de las curvas de dispersión, facilitando el trabajo analítico posterior.

Figura 23 Evolución de Velocidad de Fase: Antes y Después de la Optimización

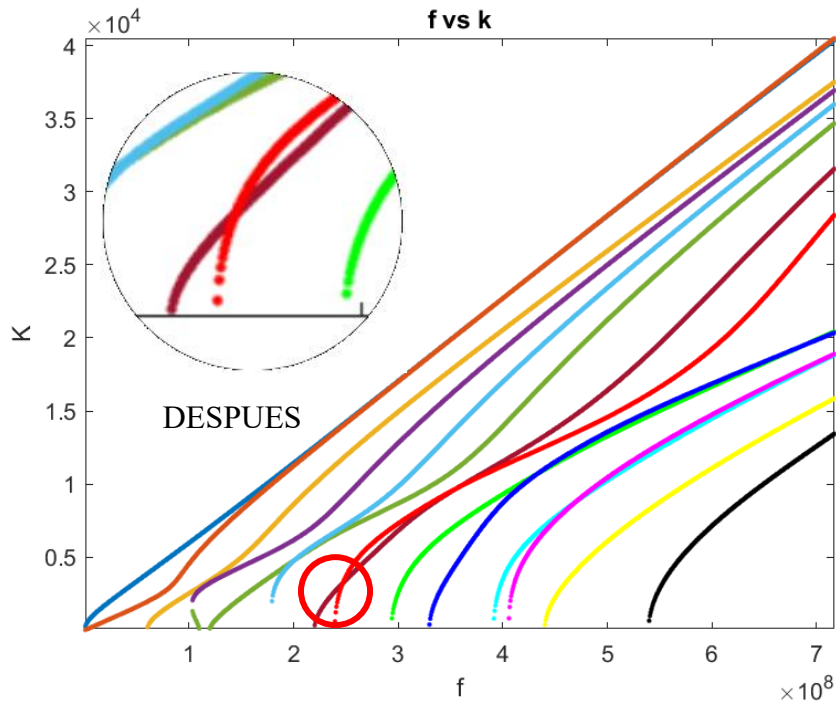


La comparativa entre las imágenes previas y las actuales subraya la evolución positiva del algoritmo y refuerza su validez para aplicaciones prácticas en el ámbito de los ensayos no destructivos y la evaluación de la integridad estructural.

Figura 24 Gráfico de Frecuencia vs Constante elástica (f vs k)



Las gráficas de dispersión actualizadas, diferenciadas por colores vivos, muestran con claridad la segregación de los modos de onda. Esta discriminación cromática facilita la identificación de los distintos modos de onda, lo cual es esencial para el análisis detallado requerido en investigaciones de materiales y estudios de ingeniería.



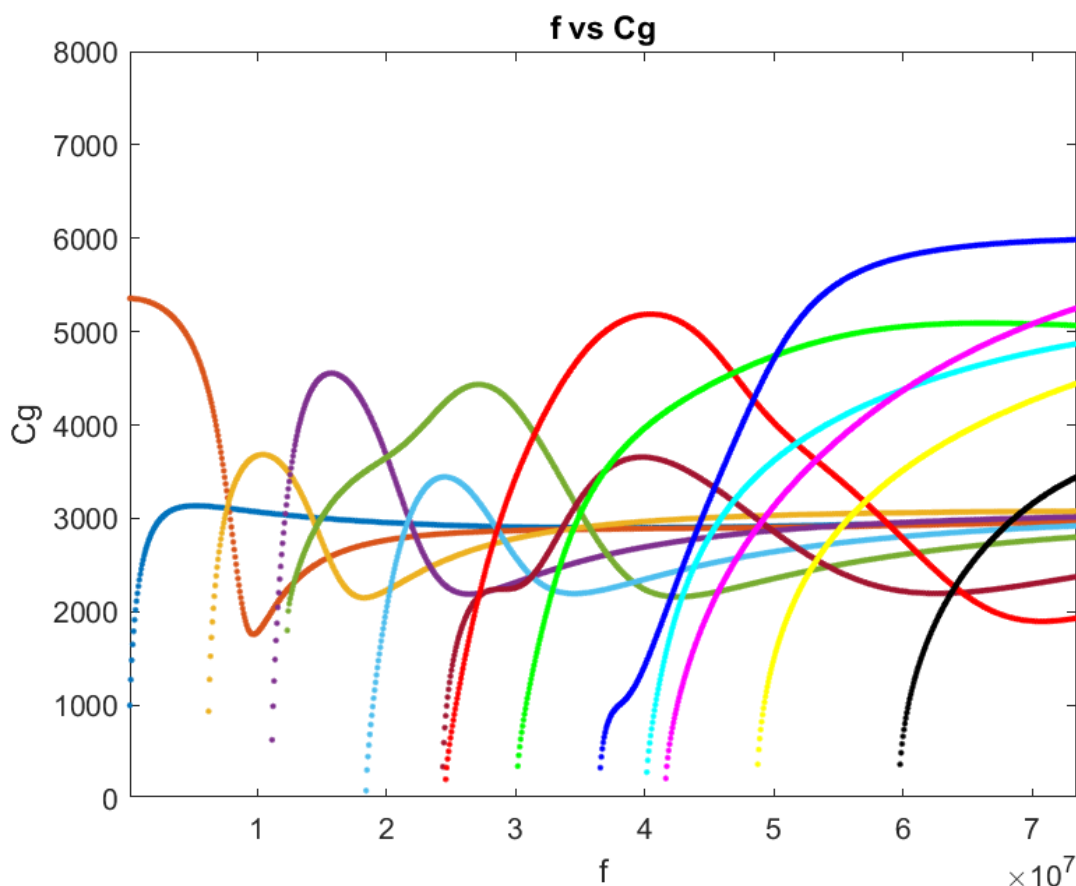
En estas figuras se demuestra una correcta clasificación de las curvas de dispersión, demostrando el funcionamiento del algoritmo.

Se realizaron pruebas cambiando las propiedades del material y el espesor de la placa para evaluar la flexibilidad y eficiencia del algoritmo. Para estas pruebas, se seleccionaron las propiedades del aluminio y del cobre, debido a su prevalencia en la industria. Además, se varió el espesor de la placa en tres valores: 5 mm y 10 mm.

El objetivo de estas pruebas fue demostrar la viabilidad del algoritmo y su practicidad en diferentes materiales y configuraciones geométricas. Los resultados fueron favorables, ya que el tiempo de procesamiento no aumentó significativamente y la clasificación de los modos se efectuó de manera correcta. Estos resultados se ilustran en las siguientes imágenes, mostrando la robustez y adaptabilidad del algoritmo frente a distintas condiciones de material y espesor.

El aluminio, con un módulo de elasticidad de (69×10^9) Pa, un coeficiente de Poisson de 0.33, y una densidad de 2700 kg/m^3 , es conocido por su alta conductividad térmica y eléctrica. En la figura (25), se observa el ordenamiento de los modos de propagación de ondas guiadas con un espesor de 10 mm, en función de la frecuencia (f) y la velocidad de grupo (C_g). La gráfica ilustra la separación de los diferentes modos, lo que facilita su interpretación y análisis detallado. Los modos están representados con diferentes colores, lo que ayuda a identificar las características dinámicas del material y a realizar un seguimiento de cada modo a través del espectro de frecuencias.

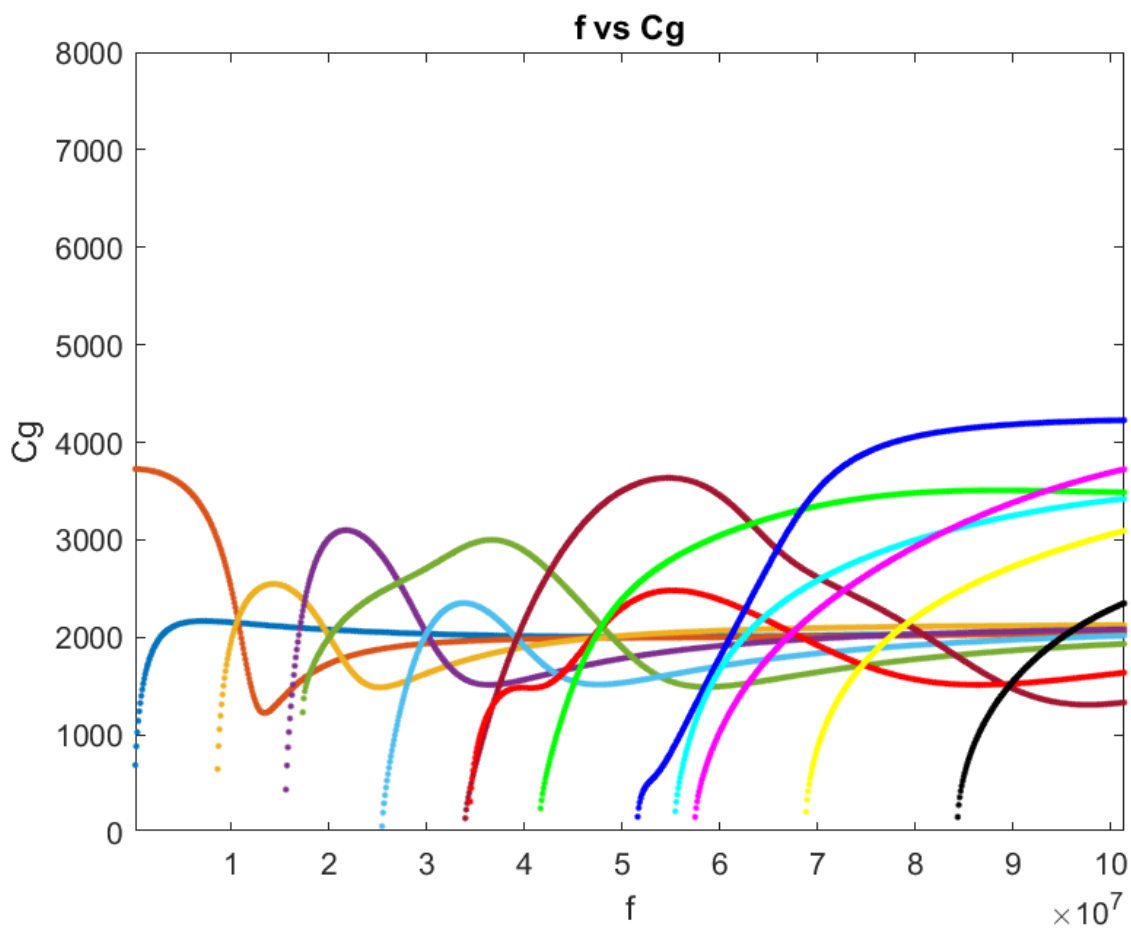
Figura 25 Ordenamiento de Modos de Propagación de Ondas Guiadas en Aluminio



El siguiente material con el que se hicieron las pruebas fue con el cobre, con un módulo de elasticidad de (110×10^9) Pa, un coeficiente de Poisson de 0.34, y una densidad de 8960

kg/m³, es apreciado por su conductividad eléctrica y térmica, así como por su alta resistencia a la corrosión. En la figura (26), se observa el ordenamiento de los modos de propagación de ondas guiadas en cobre con un espesor de 5 mm. La gráfica, que muestra la relación entre la frecuencia (f) y la velocidad de grupo (Cg), se puede ver las diferencias del análisis modal con respecto a los anteriores dos materiales connotando esto que el algoritmo genera las curvas de dispersión propias de cada material y no se basa en repetir o en entregar el mismo análisis para cada material, dando con esto más validez a la generación de valores del algoritmo y a la clasificación de los modos sin importar el material y la geometría de la prueba.

Figura 26 Ordenamiento de Modos de Propagación de Ondas Guiadas en Cobre con Espesor de 5 mm



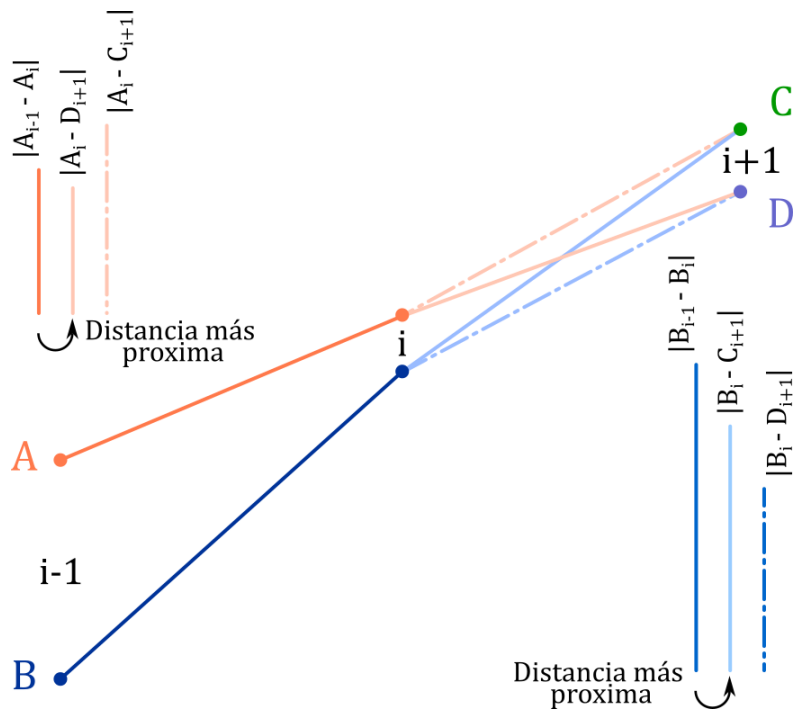
5.5.2. Metodología de Ordenamiento y Seguimiento de Modos en Diagramas de Dispersión de Ondas Guiadas

En el análisis de la propagación de ondas guiadas en estructuras, los diagramas de dispersión son herramientas de gran relevancia. Sin embargo, la presencia de información mezclada o cruzada con respecto a los modos presentes en estos diagramas puede dificultar su interpretación. Para abordar este problema, se propone una metodología de seguimiento de curvas basada en la distancia entre modos y la tendencia de las curvas en puntos previos, aplicada a los números de onda. La metodología propuesta permite ordenar los modos presentes en los diagramas de dispersión, garantizando la continuidad adecuada de las curvas sin cruces de información entre modos adyacentes. Esto facilita la interpretación de los diagramas y proporciona una mejor comprensión de la propagación de ondas guiadas en placas.

La implementación efectiva de la metodología propuesta para el ordenamiento de modos en diagramas de dispersión de ondas guiadas requiere de dos aspectos fundamentales:

1. Cantidad de frecuencias (líneas de resolución): Se recomienda emplear un número significativo de frecuencias para el análisis. Esto permite obtener una mayor resolución en los diagramas de dispersión y facilita la identificación precisa de los modos. Una cantidad insuficiente de frecuencias puede conducir a la pérdida de información crucial y a una interpretación errónea de los resultados.
2. Ordenamiento previo de los modos: Se debe realizar un ordenamiento previo de los modos de mayor a menor antes de aplicar la metodología. Este paso inicial ayuda a optimizar el proceso de seguimiento y ordenamiento, reduciendo el tiempo de procesamiento computacional.

Figura 27 Seguimiento de Modos en Diagramas de Dispersión



En general, los pasos propuestos para el seguimiento y ordenamiento de las curvas de dispersión son los siguientes:

1. Inicialización:
 - a. Se prepara la matriz de valores propios (números de onda), donde las filas representan los modos y las columnas representan las frecuencias de análisis.
 - b. Inicialmente, se añade una columna adicional con los datos de la última columna para facilitar el procesamiento de datos.

2. Iteración sobre frecuencias:

El proceso de ordenamientos de modos sigue un enfoque iterativo sobre cada una de las frecuencias de análisis, desde la primera hasta la penúltima frecuencia o columna de la matriz.

3. Conteo de modos:

Para cada una de las frecuencias, se cuentan los modos no nulos presentes en la columna actual, la siguiente y la anterior (si los hay).

4. Cálculo de distancias:
 - a. Para cada modo de la frecuencia actual, se calculan las distancias con todos los modos de la siguiente frecuencia.
 - b. Se calcula el valor absoluto de las distancias, se ordenan de menor a mayor y se seleccionan los dos valores más pequeños.
 - c. Se guardan las posiciones correspondientes a los dos valores más pequeños de las distancias.
 - d. Para los modos en las dos posiciones previamente establecidas, se calcula la distancia con los modos en las mismas posiciones, en la frecuencia anterior (si los hay).
5. Verificación de tendencia:
 - a. Para los modos en las dos posiciones previamente establecidas, se calcula la diferencia entre las distancias entre la frecuencia actual y las siguientes frecuencias, y las distancias entre la frecuencia actual y las anteriores frecuencias, calculadas previamente.
 - b. Se selecciona el modo con la menor diferencia absoluta y se guarda su posición correspondiente.
 - c. Si un modo ya ha sido asignado previamente, se selecciona la siguiente menor diferencia absoluta disponible.
6. Manejo de desajuste en el número de modos:

- a. Se ajustan las posiciones si el número de modos en la siguiente frecuencia es diferente al de la frecuencia actual.
- b. Se utilizan conjuntos de posiciones predefinidas para identificar modos no asignados y se ajustan las posiciones de los modos según sea necesario.

7. Reorganización de la matriz:

La matriz de valores propios (números de onda) se reorganiza según las nuevas posiciones asignadas para cada frecuencia.

8. Continuidad del proceso iterativo:

Se repite el proceso iterativo desde el paso 3 hasta la penúltima frecuencia o columna de la matriz de valores propios.

9. Limpieza final:

- a. Se elimina la columna adicional agregada inicialmente
- b. Se reemplazan los valores cero en la matriz con valores NaN (Not a Number).

Anteriores pasos

Los gráficos de propagación de ondas guiadas como los diagramas de dispersión son susceptibles de mostrar información mezclada o cruzada con respecto a los modos presentados. Esta particularidad derivada del ordenamiento básico de los modos al solucionar el problema de valores y vectores propios conlleva a análisis erróneos sobre las particularidades de los modos estudiados.

Para evitar esta representación errónea de las curvas de dispersión de los modos, es necesario aplicar metodologías de seguimiento de los modos, que permitan ordenar los

números de onda, velocidades de onda o fases de onda, garantizando la adecuada continuidad de las curvas sin cruces de información entre modos cercanos.

En este trabajo se propone una metodología de seguimiento de curvas, basada en la distancia entre modos y la tendencia de las curvas en puntos previos, aplicada a los números de onda. Para la aplicación de la metodología de seguimiento, es recomendable el uso de una cantidad significativa de frecuencias (líneas de resolución) y el ordenamiento previo de los números de onda en orden descendente, para cada una de las frecuencias establecidas.

Los valores propios se representan en forma matricial como la matriz λ de tamaño $i \times j$, donde las filas i corresponden a los modos y las columnas j a las frecuencias definidas para el análisis y estudio de la propagación de las ondas. Los coeficientes λ_{ij} representan los números de onda k .

1. Para cada uno de los números de onda i, j en la columna 2, calcule el valor absoluto de las diferencias con los números de onda en la columna 3 para todas las combinaciones posibles con las filas i

$$\Delta_a \lambda_i = |\lambda_{i,2} - \lambda_{1:n,3}|$$

n representa el número total de modos.

2. Determine los 2 valores más pequeños de $\Delta \lambda_i$, ordénelos de menor a mayor y guarde sus respectivas posiciones p_i .
3. Para los números de onda en las posiciones correspondientes a los dos valores más pequeños de $\Delta \lambda_i$, calcule el valor absoluto de las diferencias con los valores de número de onda en la frecuencia anterior.

$$\Delta b \lambda_2 = |\lambda_{p(2),1} - \lambda_{p(2),2}|$$

$$\Delta b\lambda_2 = |\lambda_{p(2),1} - \lambda_{p(2),2}|$$

4. Calcule el valor absoluto de la diferencia entre $\Delta a\lambda$ y $\Delta b\lambda$ en las posiciones $p1$ y $p(2)$.
5. Ordene de menor a mayor los valores de Δ y guarde la posición del menor valor qi .
6. Repita los pasos del 1 al 5, para las siguientes columnas o frecuencias.
7. Reemplace las posiciones qi en los valores propios o números de onda λ .

$$\lambda_{1:n} = \lambda_{q(1),3: end}$$

6. PRESUPUESTO

A continuación, se presenta un desglose detallado del presupuesto destinado para la realización del proyecto. Este presupuesto ha sido estimado considerando todos los aspectos clave y recursos necesarios para llevar a cabo la investigación de manera efectiva. Incluye desde gastos asociados con la asesoría y logística, hasta inversiones en herramientas y licencias de software.

Tabla 4 Presupuesto del proyecto

PRESUPUESTO			
CONCEPTO	COSTO UNITARIO	CANTIDAD	COSTO TOTAL
Asesoría			
Asesoría	\$ 120.000	8	\$ 960.000
Gastos de oficina			
Computador HP	\$ 2.399.000	1	\$ 2.399.000
Material bibliográfico	\$ 1.500.000	1	\$ 1.500.000
Licencias de software	\$ 2.100.000	1	\$ 2.100.000
Gastos logísticos			
trabajo intelectual	\$ 1.000.000	1	\$ 1.000.000
viaticos	\$ 641.000	1	\$ 641.000
Bibliografía	\$ 1.449.000	1	\$ 1.449.000
Herramientas y accesorios	\$ 1.000.000	1	\$ 1.000.000
Total sin respaldo			\$ 11.049.000
Respaldo	20%		\$ 2.209.800
Total			\$ 13.258.800

7. CONCLUSIONES

El presente trabajo se ha centrado en la clasificación de modos de propagación de ondas guiadas en placas, destacando su importancia en los ensayos no destructivos y la gestión de datos estructurales. A continuación, se presentan las principales conclusiones derivadas de esta investigación:

- Se logró definir y calcular con precisión las propiedades mecánicas y geométricas del acero 1020, aluminio y cobre para las pruebas del algoritmo, incluyendo el módulo de elasticidad, la densidad y el coeficiente de Poisson. Estos parámetros iniciales fueron fundamentales para desarrollar un modelo computacional robusto en MATLAB, el cual permite simular con el comportamiento dinámico del material.
- Se desarrolló e implementó un modelo computacional en MATLAB que permite obtener perfiles de onda y curvas de dispersión de ondas guiadas con alta precisión. El uso de técnicas avanzadas como la cuadratura Gauss-Lobatto-Legendre (GLL) y la resolución de problemas de valores propios ha permitido mejorar la precisión del algoritmo. Las pruebas realizadas confirmaron la efectividad del modelo, evidenciando su capacidad para minimizar errores y proporcionar resultados consistentes.
- El algoritmo desarrollado clasifica y visualiza de manera efectiva los modos de propagación de ondas guiadas, generando curvas de dispersión que son claras, precisas y visualmente coherentes.
- El uso del Modal Assurance Criterion (MAC) presenta limitaciones en la clasificación de modos de propagación de ondas guiadas debido a su dependencia en la similitud modal, lo que puede llevar a errores en la identificación de modos cercanos. En su lugar, la metodología descrita en la sección 5.5.2, basada en la distancia entre modos y la tendencia

de las curvas en puntos previos, permite un seguimiento y ordenamiento más preciso en los diagramas de dispersión. Este método garantiza la continuidad adecuada de las curvas y evita cruces de información entre modos adyacentes, mejorando la precisión en la clasificación.

8. OBSERVACIONES

Durante el desarrollo y análisis de este proyecto, se hicieron varias observaciones que conviene destacar para comprender mejor el contexto y la importancia del trabajo realizado:

- **Complejidad de los Datos:** Los datos asociados a las curvas de dispersión son intrínsecamente complejos debido a la naturaleza física de las ondas guiadas y los factores externos que pueden afectar su propagación. Esta complejidad requiere métodos avanzados para su procesamiento y análisis, lo cual ha sido abordado mediante la implementación de algoritmos robustos en MATLAB.
- **Importancia de la Precisión:** En el campo de los ensayos no destructivos, la precisión es de suma importancia. Pequeños errores en la identificación o clasificación de las formas de onda pueden dar lugar a interpretaciones incorrectas, afectando la integridad estructural de un material o componente. Este proyecto ha subrayado la necesidad de métodos precisos para garantizar resultados fiables y seguros.
- **Variabilidad de los Datos:** Aunque los modos de propagación tienen propiedades teóricamente definidas, en la práctica pueden variar debido a defectos en el material, condiciones ambientales y otros factores. Esta variabilidad resalta la importancia de desarrollar algoritmos que sean capaces de adaptarse y manejar estas diferencias para proporcionar análisis precisos.
- **Potencial de Mejoras Futuras:** El algoritmo desarrollado ha demostrado su eficacia, pero se ha identificado el potencial de incorporar técnicas avanzadas como el aprendizaje automático. Estas técnicas pueden mejorar aún más la precisión y adaptabilidad del algoritmo a diferentes escenarios, ampliando su aplicabilidad en diversos contextos industriales.
- **Aplicabilidad en el Mundo Real:** Aunque los resultados son prometedores, es necesario realizar más pruebas en el mundo real para validar la eficacia del algoritmo en escenarios más

complejos y variados. Estas pruebas ayudarán a confirmar la robustez y fiabilidad del algoritmo en aplicaciones prácticas.

9. RECOMENDACIONES

Basado en el desarrollo del proyecto, las observaciones realizadas y las conclusiones derivadas, se proponen las siguientes recomendaciones para futuras investigaciones y aplicaciones en el campo de los ensayos no destructivos y la clasificación de modos de propagación de ondas guiadas:

- **Formación Continua:** Es esencial que los profesionales involucrados en el uso y análisis del algoritmo reciban formación continua. Mantenerse actualizado con las últimas técnicas y avances en el campo garantizará una interpretación correcta y precisa de los resultados obtenidos por el algoritmo.
- **Expansión a Otras Plataformas:** Considerar la implementación del algoritmo en otras plataformas o lenguajes de programación computacional, especialmente cuando se manejan grandes conjuntos de datos. Esto podría mejorar significativamente el rendimiento y escalabilidad del algoritmo.
- **Incorporación de Machine Learning:** Investigar la posibilidad de incorporar técnicas de aprendizaje automático para mejorar la precisión y adaptabilidad del algoritmo a diferentes escenarios y conjuntos de datos. El aprendizaje automático puede proporcionar una mayor flexibilidad y capacidad de ajuste en la clasificación de modos de propagación.
- **Revisión y Actualización Periódica:** Dada la naturaleza cambiante de la tecnología y las ciencias aplicadas, es crucial revisar y actualizar el algoritmo periódicamente para asegurar su relevancia y eficacia. Esto permitirá mantener el algoritmo alineado con los últimos desarrollos y requerimientos del campo.
- **Documentación Exhaustiva:** Asegurarse de mantener una documentación detallada y

actualizada del algoritmo, sus aplicaciones y limitaciones. Una buena documentación facilita su uso y adaptación por parte de otros investigadores o profesionales, garantizando una correcta aplicación del algoritmo en diversas situaciones.

- **Feedback de Usuarios:** Establecer canales de comunicación con los usuarios finales del algoritmo para obtener retroalimentación directa. Esta información es invaluable para identificar áreas de mejora y adaptar el algoritmo a las necesidades reales del campo, asegurando su utilidad y efectividad en aplicaciones prácticas.

10. BIBLIOGRACIA

- [1] QUALICONTROL. (Año de publicación, no está disponible). Título de la página [Ondas Guiadas]. Recuperado de <https://qualicontrolsa.com/ondas-guiadas.php>
- [2] Rose, J. L. (1999). *Ultrasonic Waves in Solid Media*. [Libro]. Cambridge University Press.
- Korea Atomic Energy Research Institute. (2006). *Development of Long Range Nondestructive Testing Technology for Detecting Flaws in City Gas Pipelines*.
- [4] ciencias.ubiobio.cl/fisica/wiki/uploads/PedroLabrana/230027b8.pdf
- [5] J. Y. Kim, D. H. Lee, K. S. Park, Y. D. Jo, S. C. Choi, C. H. Lee, S. J. Song, and Y. M. Cheong, "Long Range Inspection of City Gas Pipeline Using Ultrasonic Guided Waves," presented at 12th Asia-Pacific Conference on Non-Destructive Testing, Auckland, New Zealand, 2006
- [6] Idzi, J. L., Rojo Tanz, B. N., Iturrioz, I., Groth, E. B., & Clarke, T. G. (2017). Estudio de la propagación de ondas guiadas sobre un riel ferroviario. In *XXIII Congreso de Métodos Numéricos y sus Aplicaciones (ENIEF)* (La Plata, 7 al 10 de noviembre 2017).
- [7] Pastor, M., Binda, M., & Harčarik, T. (2012). Modal assurance criterion. *Procedia Engineering*, 48, 543-548.
- [8] Groth E.B. Propagação de ondas de tensão em hastes retangulares no intervalo de frequência de (0;100kHz). Tesis de Maestría, Universidade Federal do Rio Grande do Sul (UFRGS), 2016.
- [9] Lowe, M. J. S. (1995). Matrix Techniques for Modeling Ultrasonic Waves in Multilayered Media. *IEEE TRANSACTIONS ON ULTRASONICS, FERROELECTRICS, AND FREQUENCY CONTROL*, 42(4).
- [10] Seco, F., Martín, J. M., Jiménez, A., Pons, J. L., Calderón, L., & Ceres, R. (2002). PCDISP: A tool for the simulation of wave propagation in cylindrical waveguides. Instituto de Automática Industrial.

- [11] Seco, F., & Jiménez, A. R. (2012). Modelling the Generation and Propagation of Ultrasonic Signals in Cylindrical Waveguides. Centro de Automática y Robótica (CAR), Madrid, Spain: Consejo Superior de Investigaciones Científicas (CSIC)-UPM
- [12] Bonafonte, A. (2008). Transformada de Fourier. Departamento de Teoría de la Señal y Comunicaciones, Universitat Politècnica de Catalunya (UPC).
- [13] Rekuvienė, R., Samaitis, V., Jankauskas, A., Sadaghiani, A. K., Saeidiharzand, S., & Koşar, A. (2024). Early-stage ice detection utilizing high-order ultrasonic guided waves. *Sensors*, 24(2850).
- [14] Gravenkamp, H., Bause, F., & Song, C. (2014). On the computation of dispersion curves for axisymmetric elastic waveguides using the Scaled Boundary Finite Element Method. *Computers and Structures*, 131, 46-55
- [15] Alrayes, O., Könke, C., Ooi, E. T., & Hamdia, K. M. (2023). Modeling Cyclic Crack Propagation in Concrete Using the Scaled Boundary Finite Element Method Coupled with the Cumulative Damage-Plasticity Constitutive Law. **Materials**, 16, 863.
- [16] Wolf, J. P., & Song, C. (1998). The Scaled Boundary Finite-Element Method: State of the Art. In T. L. Geers (Ed.), **IUTAM Symposium on Computational Methods for Unbounded Domains. Fluid Mechanics and Its Applications** (Vol. 49). Springer, Dordrecht.

ANEXOS

```
clear
clc
close all
format short

mycolors = [0, 0.4470, 0.7410; % Azul
            0.8500, 0.3250, 0.0980; % Rojo
            0.9290, 0.6940, 0.1250; % Amarillo
            0.4940, 0.1840, 0.5560; % Morado
            0.4660, 0.6740, 0.1880; % Verde
            0.3010, 0.7450, 0.9330; % Celeste
            0.6350, 0.0780, 0.1840; % Granate
            1, 0, 0; % Rojo
            0, 1, 0; % Verde
            0, 0, 1; % Azul
            0, 1, 1; % Cyan
            1, 0, 1; % Magenta
            1, 1, 0; % Amarillo
            0, 0, 0]; % Negro
```

Propiedades y geometría - acero 1020:

```
E = 186*10^9; % Módulo de elasticidad [Pa]
V = 0.29;    % coeficiente de poisson
P = 7870;    % densidad [kg/m^3]
La = 0;     % factor de atenuación longitudinal
Sa = 0;     % factor de atenuación cortante
D = 1e-3;   % espesor [m]
[K,L,G,M,C,Vp,Vs] = prop(E,V,P); % propiedades reales
[VpC,VsC,ErC,VrC,CC] = prop_c(P,Vp,Vs,La,Sa); % propiedades complejas
```

Condiciones de trabajo:

```
FM1 = round(Vs/D)*6;
NP1 = 1000;
TP1 = FM1/NP1;
O1 = 9;
J1 = D/2;
```

Matrices de rigidez:

```
[E0,E1,E2,EM] = k_eig(P,J1,O1,CC);
```

Matriz T de transformación:

```
s1 = size(E0);
M = mT(s1);
E0T=(M'*E0*M);
E1T=(M'*E1*M);
E2T=(M'*E2*M);
EMT=(M'*EM*M);
```

Autovalores - Ax=yBx:

```
[v1,M7,Vp]=lapiz_eig(TP1,FM1,E0T,E1T,E2T,EMT);

Nf = size(M7,2);
Nm = size(M7,1); % Magnitud compleja de los valores
proprios
% % % % M7(:,Nf+1) = zeros(Nm,1);
% % % % Valp = abs(M7);
% % % % Vp(:, :,Nf+1) = zeros(Nm,Nm,1);
% % % % Vecp = abs(Vp);
% % % %
% % % % Dif_vp = zeros(Nm,Nm,Nf);
% % % % MAC_f = zeros(Nm,Nm,Nf);
% % % % Dif_n = zeros(Nm,Nm,Nf);
% % % % Crit = zeros(Nm,Nm,Nf);
% % % % Pos = zeros(Nm,1);
% % % %
% % % % for i = 1:1:Nf
% % % %     for j = 1:1:Nm
% % % %         for k = 1:1:Nm
% % % %             Dif_vp(k,j,i) = abs(Valp(j,i) - Valp(k,i+1));
% % % %             MAC_f(k,j,i) =
abs((abs(Vecp(j,:,i)*Vecp(k,:,i+1)')^2)/(Vecp(j,:,i)*Vecp(j,:,i)'*Vecp(k,:,i+1)*V
ecp(k,:,i+1)'));
% % % %         end
% % % %         Dif_n(:,j,i) = 1 - normalize(Dif_vp(:,j,i),"range");
% % % %         Crit(:,j,i) = (MAC_f(:,j,i) + Dif_n(:,j,i))/2;
% % % %         [~,Pos(j)] = max(Crit(:,j,i));
% % % %     end
% % % %     Valp(:,i+1:Nf) = Valp(Pos,i+1:Nf);
% % % %     Vecp(:, :,i+1:Nf) = Vecp(Pos, :,i+1:Nf);
% % % %     M7(:,i+1:Nf) = M7(:,i+1:Nf);
% % % %     Vp(:, :,i+1:Nf) = Vp(Pos, :,i+1:Nf);
% % % % end
% % % %
% % % % M7(:,Nf+1) = [];
% % % % Vp(:, :,Nf+1) = [];
```

```

%%% for i = 1:1:size(M7,2)
%%%     [M7(:,i),I] = sort(round(M7(:,i),7));
%%%     Vp(:,i) = Vp(I,:,i);
%%%     for j = 1:2:size(M7,1)-1
%%%         V_II = [j;j+1];
%%%         % [~,II] = sort(abs(imag(Vp(j:j+1,1,i))), 'descend');
%%%         % Vp(j:j+1,:,i) = Vp(V_II(II),:,i);
%%%         if i > 1
%%%             Vp_d1 = abs(abs(Vp(j,1,i)) - abs(Vp(j,1,i-1)));
%%%             Vp_d2 = abs(abs(Vp(j,1,i)) - abs(Vp(j+1,1,i-1)));
%%%             [~,II] = sort([Vp_d1; Vp_d2]);
%%%         else
%%%             Vp_d1 = abs(abs(Vp(j,1,i)) - abs(Vp(j,1,i+1)));
%%%             Vp_d2 = abs(abs(Vp(j,1,i)) - abs(Vp(j+1,1,i+1)));
%%%             Vp_d3 = abs(abs(Vp(j+1,1,i)) - abs(Vp(j,1,i+1)));
%%%             Vp_d4 = abs(abs(Vp(j+1,1,i)) - abs(Vp(j+1,1,i+1)));
%%%             [~,II] = sort([Vp_d1; Vp_d2; Vp_d3; Vp_d4]);
%%%             if II(1) || II(2) == 1
%%%                 II = [1;2];
%%%             elseif II(3) || II(4) == 1
%%%                 II = [2;1];
%%%             end
%%%         end
%%%         Vp(j:j+1,:,i) = Vp(V_II(II),:,i);
%%%     end
%%% end

```

Filtro:

```

[s1,s2]=size(M7);           % Dimensiones de la matriz [r,c]
mr0 = zeros(s1,s2);
mr1 = zeros(s1,s2);
mr2 = zeros(s1,s2);
for j=1:s2
    for i=1:s1
        if (imag(M7(i,j))==0) && (real(M7(i,j))>0)
            %if (imag(M7(i,j))==0)
                mr0(i,j) = real(M7(i,j));           % Identifica
valores reales
            elseif (imag(M7(i,j))~=0) && (real(M7(i,j))>0)           % Identifica
valores complejos
                mr1(i,j) = real(M7(i,j));           % Identifica parte
real
                mr2(i,j) = imag(M7(i,j));           % Identifica parte
imaginaria
            end
        end
    end
end

```

```

end
mr0(mr0==0)=nan; % Si valor es igual
a 0 --> valor = NaN
freq = 2*pi*v1;

```

Preordenamiento de modos:

```

[s1,s2] = size(mr0); % Tamaño de matriz
de valores reales
M_end=mr0(:,end); % Columna final de
la matriz
n_logic=isnan(M_end); % Identifica
valores NaN
[n_cuentas,valor]=groupcounts(n_logic); % Identifica número
de valores NaN y valores no NaN
n_modos=n_cuentas(1); % Determina número
de valores no NaN
v=zeros(n_modos,s2);
Vp_aux = zeros(n_modos,s1,s2);
Vp_n2 = zeros(n_modos,s1,s2+1);
MAC = zeros(n_modos,n_modos,s2-1);
Dif = zeros(n_modos,n_modos,s2-1);
Dif_n = zeros(n_modos,n_modos,s2-1);
Crit = zeros(n_modos,n_modos,s2-1);
v = zeros(n_modos,s2);
ord = zeros(n_modos,s2);

Dif_vp = zeros(n_modos,n_modos,Nf); %%%%%%%%%%%
MAC_f = zeros(n_modos,n_modos,Nf);
Dif_n = zeros(n_modos,n_modos,Nf);
Crit = zeros(n_modos,n_modos,Nf);
Pos = zeros(n_modos,1); %%%%%%%%%%%
%v(:,Nf+1) = zeros(n_modos,1);
% v(:,Nf+1) = v(:,Nf);

for i = 1:1:s2 % Contador en reversa
    v_aux1=mr0(:,i);
    v_logic1=isnan(v_aux1);
    indices = find(v_logic1 == 0);
    Vp_aux(indices,:,i) = Vp(indices,1:s1,i);
    Vp_aux2 = Vp_aux(v_logic1==0,:,i);
    lvp = size(Vp_aux2,1);
    Vp_n(1:lvp,:,i) = Vp_aux2(1:lvp,:);

```

```

[n_count,n_value]=groupcounts(v_logic1);

```

```

    if n_count(1)==n_modos
        [v(:,i),ord(:,i)]=sort(v_aux1(v_logic1==0), 'descend');           % Ordena
los modos reales en orden descendente
    else
        [v_aux2,ord2]=sort(v_aux1(v_logic1==0), 'descend');           % Ordena los
modos
        l1=length(v_aux2);
        v(1:l1,i)=v_aux2(1:l1);
        ord(1:l1,i) = ord2(1:l1);
    end
    l2 = ord(ord(:,i)~=0,i);
    l3 = length(l2);
    Vp_n2(1:l3,:,i) = Vp_n(ord(1:l3,i),:,i);
    %v(v==0) = 1e7;
end

v(:,Nf+1) = v(:,Nf);

for i = 1:1:s2
    nzj = nnz(v(:,i)); % Número de modos diferentes de 0
    nzk = nnz(v(:,i+1)); % Números de modos diferentes de 0 en la siguiente
frecuencia
    if i > 1
        nzi = nnz(v(:,i-1));
    end
    dist = zeros(nzk,1);
    dist2 = zeros(nzk,1);
    dif = zeros(2,1);
    Pos_n = zeros(nzj,1);
    for j = 1:1:nzj
        for k = 1:1:nzk
            dist(k) = v(j,i) - v(k,i+1);

            % if i > 1
            %     dist2(k) = v(k,i-1) - v(k,i);
            % else
            %     dist2(k) = 0;
            % end
            % dif(k) = dist(k) - dist2(k);
        end
    end
    [~,min_4] = sort(abs(dist));
    for k = 1:1:2
        m = min_4(k);
        if i > 1
            dist2(m) = v(j,i-1) - v(j,i);
        else

```

```

        dist2(m) = 0;
    end
    dif(k) = dist(m) - dist2(m);
end
% dist_n = normalize(abs(dist),"norm",1);
% dif_n = normalize(abs(dif),"norm",1);
% dd_n = (10*dist_n + dif_n)/11;
[~,Pos_p] = sort(abs(dif));
Pos_n(j) = min_4(Pos_p(1));
if j > 1 && ismember(Pos_n(j), Pos_n(1:j-1))
    Pos_n(j) = min_4(Pos_p(2));
end
% [~,Pos_n(j)] = min(abs(dif));
end

df = nzk - nzj;
Pj = 1:1:nzk;
Pd = setdiff(Pj,Pos_n);
if nzj < nzk
    Pos_n(nzj+1:nzk) = Pd;
end
if nzj > nzk
    Pos_n(nzk+1:nzj) = [];
    Pd = setdiff(Pj,Pos_n);
end
if nzj > nzk && nzi < nzj
    if isempty(Pd)
    else
        Pos_n(nzi+1:nzj) = Pd;
    end
    % Pos_n(nzk+1:nzj) = [];
    % Pd = setdiff(Pj,Pos_n);
end
v(1:nzk,i+1:end) = v(Pos_n,i+1:end);
Vp_n2(1:nzk,:,i+1:end) = Vp_n2(Pos_n,:,i+1:end);
end

v(v==1e7) = 0;
v(:,Nf+1) = [];

v(v==0)=nan;

```

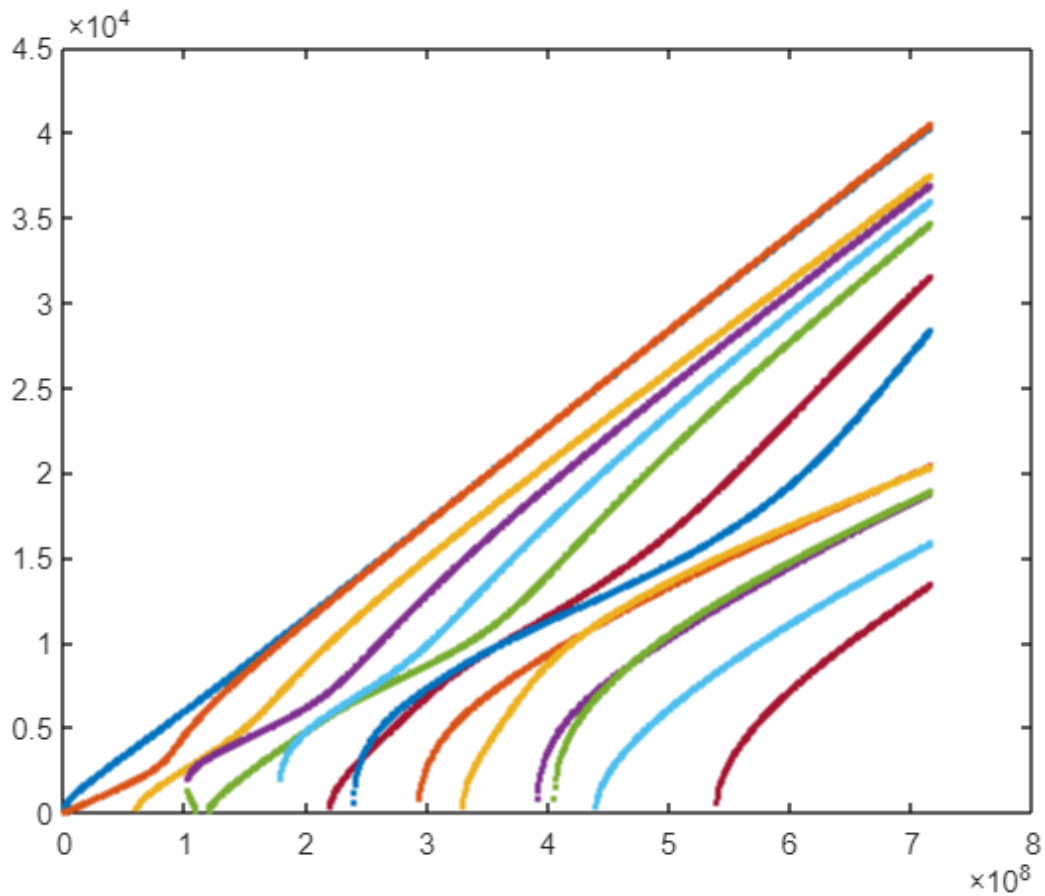
```
[s3,s4]=size(v);
```

```

vt1=zeros(s3,s4);
vt1(vt1==0)=nan;
it=1;
for j = 1:1:s4
    vt1(:,it)=abs(v(:,j));
    it=it+1;
end
figure(1)
plot(freq,vt1(:,:),'.')
hold on

xlim("auto")
ylim("auto")

```

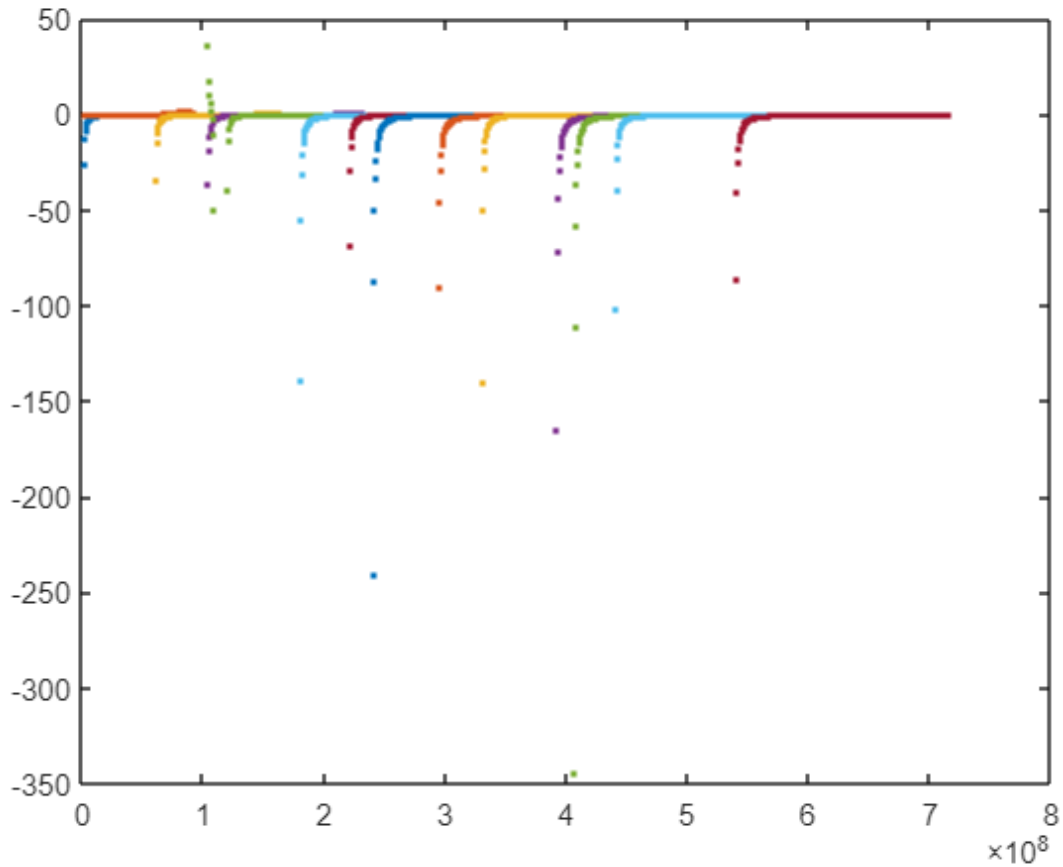


```

%%
vtd21=diff(vt1,2,2);
figure(3)
plot(freq(3:end),vtd21,'.')

```

```
hold on
```



Extraer y organizar los datos numéricos:

```
% Aquí es donde guardas las variables en archivos .mat
```

```
save('frecuencias.mat', 'freq');
```

```
save('wave_numbers.mat', 'v');
```

```
save('phase_velocities.mat', 'v1');
```

```
% Calcula la velocidad de fase para cada modo
```

```
velocidades_fase = v1 ./ v;
```

```
% Calcula la velocidad de grupo para cada modo
```

```
% el vector `freq` tiene el mismo tamaño que `dw` y `dk` después de la  
diferenciación
```

```
dfreq = freq(2:end); % Esto se hace ya en tu código
```

```
dw = diff(v1); % Diferencia de frecuencias angulares
```

```

dk = diff(v,1,2);      % Diferencia de números de onda
velocidades_grupo = dw ./ dk;

% Normaliza cada perfil de modo para la velocidad de fase
for i = 1:size(velocidades_fase, 2)
    velocidades_fase(:, i) = velocidades_fase(:, i) / norm(velocidades_fase(:,
i));
end

% Normaliza cada perfil de modo para la velocidad de grupo
% Nota: Asegúrate de que `velocidades_grupo` no contenga NaN o Inf antes de
normalizar
for i = 1:size(velocidades_grupo, 2)
    if all(~isnan(velocidades_grupo(:, i))) && all(~isinf(velocidades_grupo(:,
i)))
        velocidades_grupo(:, i) = velocidades_grupo(:, i) /
norm(velocidades_grupo(:, i));
    end
end
for i = 1:size(velocidades_grupo, 2)
    perfil_actual = velocidades_grupo(:, i);
    indices_validos = ~isnan(perfil_actual);
    x_validos = dfreq(indices_validos); % Asume que dfreq es el vector de
frecuencias sin el primer elemento
    y_validos = perfil_actual(indices_validos); % Valores válidos de la columna
i de velocidades_grupo

    if sum(indices_validos) > 1 % Si hay al menos dos valores no-NaN para
interpolar
        % Se interpola solo en el rango donde hay valores válidos
        y_interp = interp1(x_validos, y_validos, x_validos, 'linear', 'extrap');
        % Se asigna la salida interpolada solo en los índices válidos
        velocidades_grupo(indices_validos, i) = y_interp;
    else
        % Si solo hay un valor válido o ninguno, se mantiene NaN o se puede
gestionar de otra manera
        velocidades_grupo(:, i) = NaN;
    end
end
end
% Normaliza cada perfil de modo para la velocidad de grupo.
for i = 1:size(velocidades_grupo, 2)
    % Solo normaliza si el perfil no es completamente NaN.
    if all(~isnan(velocidades_grupo(:, i)))
        perfil_actual = velocidades_grupo(:, i);
        perfil_normalizado = perfil_actual / norm(perfil_actual);
    end
end

```

```

        velocidades_grupo(:, i) = perfil_normalizado;
    end
end

% Paso 1: Verificar los datos antes de la normalización y el cálculo del MAC
for i = 1:n_modos
    perfil_modo = velocidades_grupo(:, i);
    if all(perfil_modo == 0)
        disp(['Modo ' num2str(i) ' es todo ceros.']);
    elseif all(perfil_modo == perfil_modo(1))
        disp(['Modo ' num2str(i) ' es constante.']);
    end
end
end

```

Clasificación de modos

```

% Pre-allocate the MAC matrix for speed.
num_modos = size(velocidades_grupo, 2);

% Paso 2: Calcular la matriz MAC
MAC_matrix = zeros(num_modos, num_modos);
for i = 1:num_modos
    for j = i:num_modos
        if any(velocidades_grupo(:, i)) && any(velocidades_grupo(:, j))
            % Calcula el número MAC solo si hay datos no nulos o no cero
            MAC_matrix(i, j) = abs(dot(velocidades_grupo(:, i),
                velocidades_grupo(:, j)))^2 / ...
                (norm(velocidades_grupo(:, i))^2 *
                norm(velocidades_grupo(:, j))^2);
            MAC_matrix(j, i) = MAC_matrix(i, j); % La matriz MAC es simétrica.
        else
            disp(['Modo ' num2str(i) ' o Modo ' num2str(j) ' tiene datos no
                válidos.']);
        end
    end
end
end

% Pre-asignar la matriz para las distancias euclidianas
distancias_euclidianas = zeros(num_modos, num_modos);

% Calcular la distancia euclidiana para cada par de vectores de perfil de modo.
for i = 1:num_modos

```

```

    for j = i+1:num_modos % No es necesario calcular para j <= i, ya que la
matriz es simétrica y la diagonal es cero.
        vector_diferencia = velocidades_grupo(:, i) - velocidades_grupo(:, j);
        distancias_euclidianas(i, j) = norm(vector_diferencia);
        distancias_euclidianas(j, i) = distancias_euclidianas(i, j); % La matriz
es simétrica.
    end
end

% Pre-asignar espacio para los vecinos más cercanos
vecinos_mas_cercanos = zeros(num_modos, 1);

% Identificar el vecino más cercano para cada modo basado en la distancia
euclidiana.
for i = 1:num_modos
    % Excluimos la columna actual asignando NaN antes de encontrar el mínimo.
    distancias_a_otros = distancias_euclidianas(i, :);
    distancias_a_otros(i) = NaN;
    [~, vecino_mas_cercano] = min(distancias_a_otros);
    vecinos_mas_cercanos(i) = vecino_mas_cercano;
end

% Definir un umbral para el valor MAC considerando que dos modos son similares.
umbral_MAC = 0.9 ; % Este es un valor de ejemplo, se debe ajustar según sea
necesario.

% Inicializa el arreglo de grupos
grupos = cell(1, num_modos);
contador_grupos = 0;

% Bucle para agrupar modos basados en el número MAC
for i = 1:num_modos
    % Si el modo actual no ha sido ya incluido en un grupo
    if ~any(cellfun(@(c) ismember(i, c), grupos))
        similitudes = find(MAC_matrix(i, :) >= umbral_MAC);
        if ~isempty(similitudes)
            contador_grupos = contador_grupos + 1;
            grupos{contador_grupos} = similitudes;
            % Marcamos todos los modos similares como ya revisados
            MAC_matrix(similitudes, :) = 0;
            MAC_matrix(:, similitudes) = 0;
        end
    end
end

```

```

        end
    end
end
grupos = grupos(1:contador_grupos); % Elimina espacios no utilizados

% Ajustar 'freq' para que coincida con el número de columnas en
'velocidades_grupo'
freq_ajustado = freq(2:end); % Ahora 'freq_ajustado' tiene 999 elementos

if ~isempty(grupos) && ~isempty(grupos{1})
    % Seleccionar el primer grupo
    primer_grupo = grupos{1};

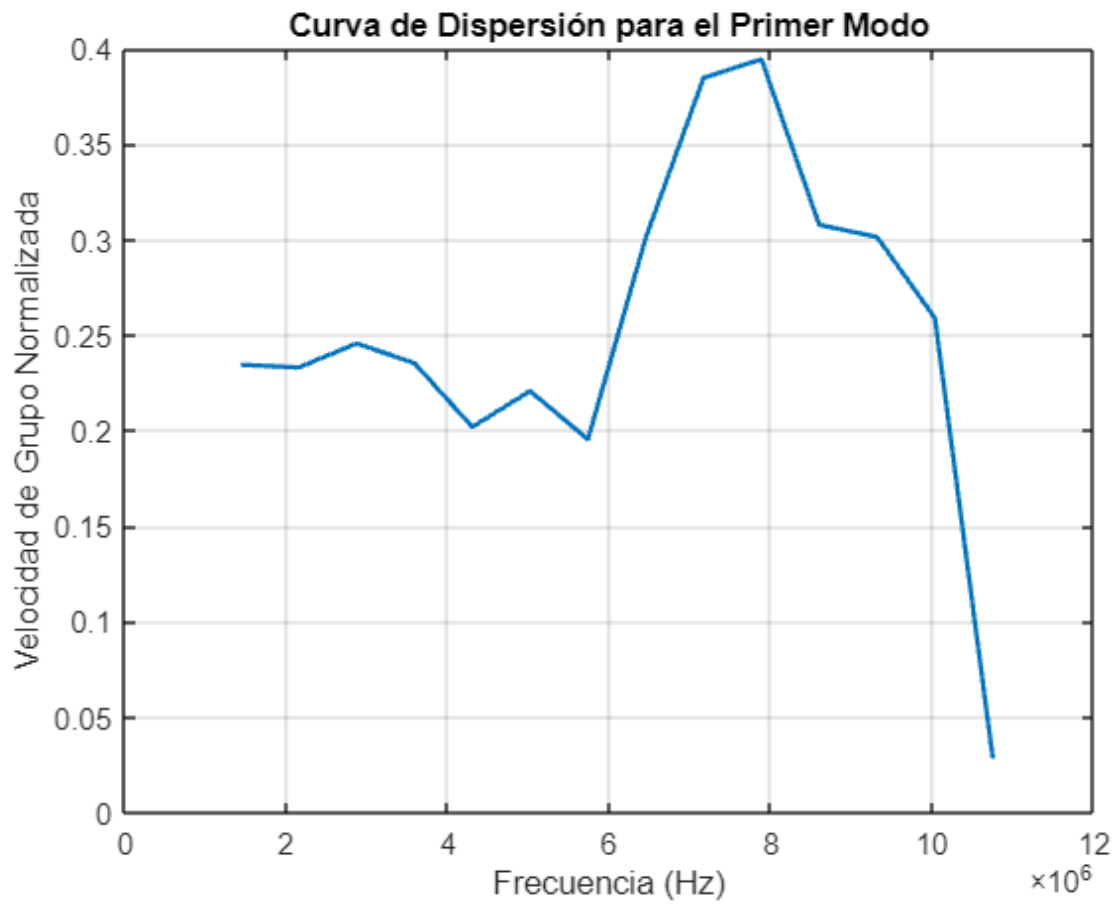
    % Asegurarse de que el primer grupo tiene al menos un elemento
    if ~isempty(primer_grupo)
        % Seleccionar el primer modo del primer grupo
        indice_primer_modo = primer_grupo(1);

        % Asegurarse de que el índice seleccionado no excede el número de modos
        disponibles
        if indice_primer_modo <= size(velocidades_grupo, 2)
            % Obtener el perfil de modo para este modo
            perfil_primer_modo = velocidades_grupo(:, indice_primer_modo);
            indices_no_nan = ~isnan(perfil_primer_modo); % Ignorar los valores
NaN

            % Graficar el primer modo
            figure;
            plot(freq_ajustado(indices_no_nan),
perfil_primer_modo(indices_no_nan), 'LineWidth', 1.5);
            xlabel('Frecuencia(Hz)');
            ylabel('Velocidad de Grupo Normalizada');
            title('Curva de Dispersión para el Primer Modo');
            grid on; % Añadir una cuadrícula para mejor visualización
        else
            error('El índice del primer modo excede el número de modos
disponibles. ');
        end
    else
        error('El primer grupo está vacío. ');
    end
end
else
    error('No se han definido grupos. ');
end

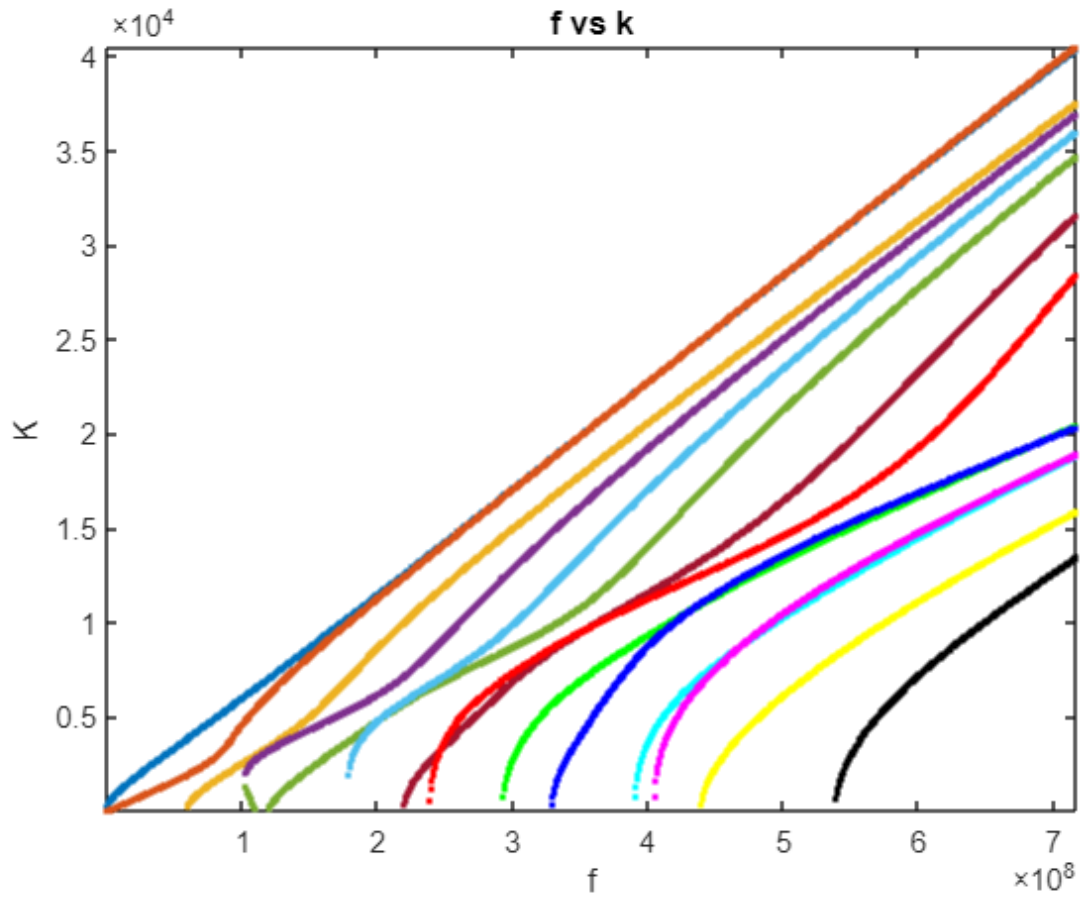
```

end



Gráficos:

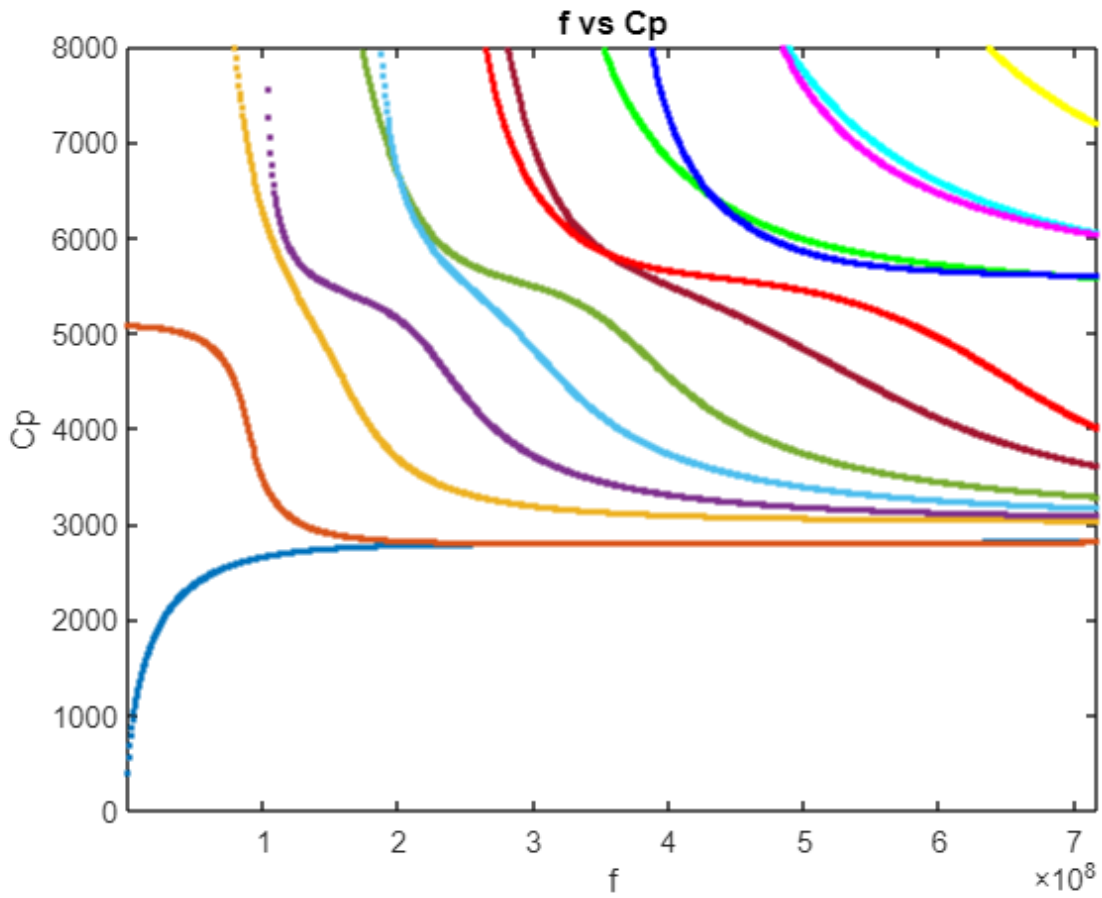
```
plot(freq,v, '.')
xlim tight
ylim tight
xlabel('f')
ylabel('K')
title('f vs k')
ax = gca;
ax.ColorOrder = mycolors;
```



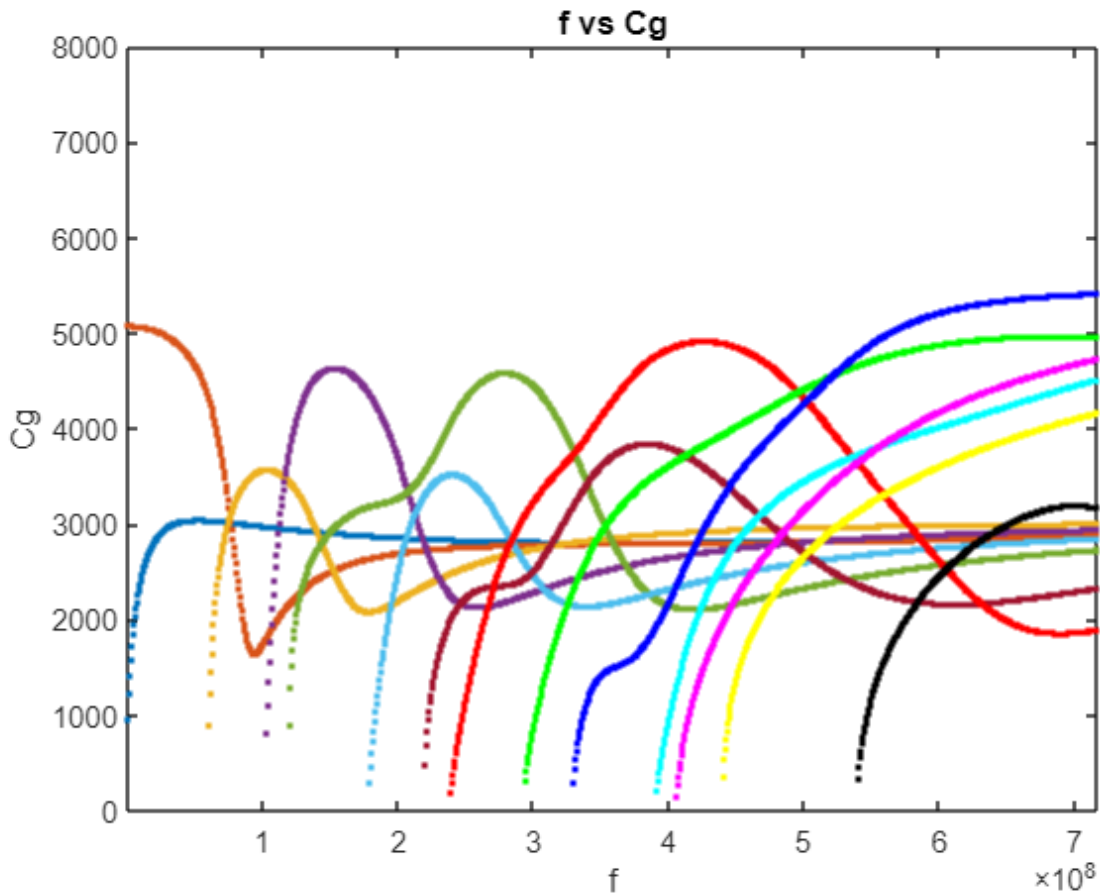
```

plot(freq,v1./v,'.')
xlim tight
ylim ([0 8000])
xlabel('f')
ylabel('Cp')
title('f vs Cp')
ax = gca;
ax.ColorOrder = mycolors;
ax = gca;
ax.ColorOrder = mycolors;

```



```
dfreq=freq(2:end);
dw=diff(v1);
dk=diff(v,1,2);
plot(dfreq,dw./dk, '.')
xlim tight
ylim ([0 8000])
xlabel('f')
ylabel('Cg')
title('f vs Cg')
ax = gca;
ax.ColorOrder = mycolors;
```



```
function [K,L,G,M,C,Vp,Vs] = prop(E,V,P)
K = E/(3*(1-2*V)); % módulo masivo [Pa]
L = E*V/((1+V)*(1-2*V)); % primer parámetro de Láme [Pa]
G = E/(2*(1+V)); % Módulo cortante [Pa]
M = E*(1-V)/((1+V)*(1-2*V)); % Módulo onda P [Pa]
C = E/((1+V)*(1-2*V))*[1-V V 0;V 1-V 0;0 0 (1-2*V)/2]; % Tensor de elasticidad [Pa]
Vp = sqrt(M/P); % velocidad de ondas presivas [m/s]
Vs = sqrt(G/P); % velocidad de ondas cortantes [m/s]
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [VpC,VsC,ErC,VrC,CC] = prop_c(P,Vp,Vs,La,Sa)
VpC = Vp/(1+1i*(La/(2*pi)));
VsC = Vs/(1+1i*(Sa/(2*pi)));
ErC = P*(VsC^2)*((3*VpC^2-4*VsC^2)/(VpC^2-VsC^2));
```

```

VrC = (1/2)*((VpC^2-2*VsC^2)/(VpC^2-VsC^2));
CC = ErC/((1+VrC)*(1-2*VrC))*[1-VrC VrC 0;VrC 1-VrC 0;0 0 (1-2*VrC)/2];
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [E0,E1,E2,EM] = k_eig(D,J,O,C)
syms x
b1=[0 0;0 1;1 0];
b2=[1 0;0 0;0 1];

[nn,ww]=GLL(O);
N=SF(nn);
D_N=(1/J)*diff(N);

sE0=0;
sE1=0;
sE2=0;
sEM=0;

B1=b1*N;
B2=b2*D_N;

fE0=B1'*C*B1*J;
fE1=B2'*C*B1*J;
fE2=B2'*C*B2*J;
fEM=D*N'*N*J;

for i=1:1:length(nn)
    sE0=sE0+(ww(i)*(subs(fE0,x,nn(i))));
    sE1=sE1+(ww(i)*(subs(fE1,x,nn(i))));
    sE2=sE2+(ww(i)*(subs(fE2,x,nn(i))));
    sEM=sEM+(ww(i)*(subs(fEM,x,nn(i))));
end

E0=double(sE0);
E1=double(sE1);
E2=double(sE2);
EM=double(sEM);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function M = SF(nodes)
for i=1:1:length(nodes)

```

```

    M1(:,i)=shfunc(i,nodes);
end
M2=M1(1)*eye(2);
for j=2:1:length(nodes)
    M3=M1(j)*eye(2);M2=[M2 M3];
end
M=M2;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function ni = shfunc(i,xi)
syms x
n=1;m=1;
for j=1:1:length(xi)
    if i~=j
        n=n.*(x-xi(j));m=m.*(xi(i)-xi(j));
    end
end
ni=n./m;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [x,w]=GLL(o)
n1=o+1;
x=cos(pi*(0:1:o)/o)';
p=zeros(n1,n1);
xold=2;
while max(abs(x-xold))>eps
    xold=x;
    p(:,1)=1;
    p(:,2)=x;
    for k= 2:1:o
        p(:,k+1)=((2*k-1)*x.*p(:,k)-(k-1)*p(:,k-1))/k;
    end
    x=xold-(x.*p(:,n1)-p(:,o))./(n1*p(:,n1));
end
w=2./(o*n1*p(:,n1).^2);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function M = mT(s1)
M=eye(s1(1));
for i = 1:2:s1(1)
    M(i,i)=1i;
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [v1,M7,Vp]=lapiz_eig(Tp,F,E0,E1,E2,EM)
n=round(F/Tp);
s0=size(E1);
M0=zeros(s0);

```

```

E1S=(E1'-E1)/(-1i);
M7 = zeros(s0(1)*2,n);
for i=1:1:n
    w=2*pi*(i*Tp);
    v1(1,i)=w;
    A=[M0,E2-(w^2)*EM;E2-(w^2)*EM,E1S];
    B=[E2-(w^2)*EM,M0;M0,-E0];
    [Vpi,M4]=eig(A,B);
    v2=diag(M4);
    M7(:,i)=v2;
    Vp(:, :, i) = Vpi;
end
end

```