

# Scattering Múltiple

Beñat Abad, Víctor García

20 de abril de 2025

## Resumen

Utilizando MATLAB, se simula el comportamiento de ondas flexurales puntuales y planas al atravesar una placa delgada sobre la cual se sitúan diferentes redes de resonadores. Variando los tipos de redes, se estudia el comportamiento de estas ondas y se visualiza el scattering múltiple que se produce. Comparando con modelos teóricos y simulaciones similares, se estudia la exactitud de la simulación, buscando características físicas que deberían aparecer, como el efecto borde. Se comentan además los problemas computacionales hallados y las limitaciones del modelo empleado, entre otras la falta de unas condiciones de contorno definidas para la red de resonadores. Además, se trata brevemente el *problema de molinos de viento* en scattering múltiple.

## 1. Introducción

El estudio del *scattering* múltiple en medios periódicos es un tema de gran interés en física de ondas, con aplicaciones que van desde la óptica hasta la mecánica de sólidos. En particular, las redes periódicas de dispersores (o *scatterers*) permiten observar fenómenos característicos como la aparición de *band gaps* (o bandas prohibidas), localización de modos o estados topológicos. Entre estas estructuras, destacan las redes hexagonales o tipo *honeycomb*, que reproducen propiedades análogas a las del grafeno, como la aparición de *efectos borde* [1].

El objetivo de este proyecto es la simulación de distintas redes de resonadores mediante la construcción de un modelo numérico, que permita simular el efecto que tendría una onda flexural incidente sobre ellas. A través de simulaciones en MATLAB, se estudia el fenómeno de *scattering* múltiple y se analizan propiedades emergentes como el efecto borde. Además, queremos comprobar la importancia de redes simétricas para simular el scattering, comprobando la aparición del problema de molinos de viento para un caso de red de scatterers en posiciones aleatorias.

Durante el desarrollo del trabajo, se discuten también distintas dificultades computacionales, incluyendo la composición de condiciones de contorno adecuadas y la estabilidad numérica del algoritmo. En conjunto, el estudio permite evaluar la eficacia del modelo implementado y la relevancia de la geometría de la red en la aparición de fenómenos físicos significativos.

## 2. Marco teórico y descripción del modelo físico

El *scattering* es un proceso físico por el cual una onda, al incidir en una partícula o estructura, se desvía de su trayectoria original. En casos simples (una onda incidiendo en una sola partícula) este proceso no puede ser predicho con exactitud, puesto que la posición relativa del centro de scattering con respecto de la onda es prácticamente imposible de medir. En cambio, cuando la onda en cuestión incide en sistemas de numerosas partículas, la aleatoriedad de las interacciones acaba promediándose y cancelándose, haciendo que sea posible predecir una distribución de intensidad tras el scattering [2]. Esto es el *scattering múltiple*, y es el principal problema a abordar en este proyecto.

### 2.1. Modelo físico

Puesto que para que esta simulación de la interacción de los scatterers sea posible, se requiere de una gran cantidad de partículas, el problema resulta demasiado complejo como para resolverse sin ayuda computacional. Nótese antes de proseguir que este proceso de scattering no es único para un solo tipo de ondas, sino que cualquier onda puede ser desviada por algún tipo de partícula o sistema mecánico y sufrir scattering. Para los propósitos de este proyecto, trabajaremos con una onda flexural que atraviesa una placa con una cantidad finita de resonadores masa-resorte organizados de manera periódica.

En el modelo implementado no se definen condiciones de contorno físicas en los límites del dominio. Esto se debe a que la formulación empleada, basada en la función de Green, supone un entorno abierto o infinito en el cual las ondas pueden propagarse libremente.

Esta simplificación resulta adecuada para el objetivo del modelo: simular *scattering* múltiple en un medio que se comporta como si fuera infinito, al menos lejos de los dispersores.

## 2.2. Resolución teórica del scattering múltiple

Para resolver un sistema de scattering múltiple, necesitamos primero explicar dos componentes importantes: la función de Green, una herramienta matemática que se emplea para calcular la dispersión de la onda al pasar por un scatterer; y la ecuación de movimiento para la placa donde se sitúa la red de scatterers. En este proyecto, suponemos una placa elástica con una distribución periódica de resonadores resorte-masa, con constantes de fuerza  $k_R$  y masa  $m_R$ , y además suponemos un número finito de resonadores en posiciones  $R_\alpha$ . La ecuación de movimiento para la placa de resonadores [1] será

$$\left(\nabla^4 - \frac{\omega^2 \rho h}{D}\right) W_1(\vec{r}) = \sum_{\alpha} t_{\alpha} W_1(\vec{R}_{\alpha}) \delta(\vec{r} - \vec{R}_{\alpha}). \quad (1)$$

donde  $\nabla^4$  es el operador bilaplaciano o biarmónico,  $D = Eh^3/12(1 - \nu^2)$  es la rigidez de la placa,  $\rho$  es la densidad de la placa y  $h$  es el grosor. Además, por definición

$$t_{\alpha} = \frac{m}{D} \frac{\omega_R^2 \omega^2}{\omega_R^2 - \omega^2} \quad (2)$$

es la fuerza del resonador. Para resolver la ecuación (1), necesitamos encontrar una función de Green que satisfaga

$$(\nabla^4 - k^4) G_0(\vec{r}) = \delta(\vec{r}), \quad (3)$$

en donde el número de onda  $k$  también satisfaga que

$$k^4 = \omega^2 \frac{\rho h}{D} \quad (4)$$

Una función de Green es tal que resuelve una ecuación diferencial del tipo  $LG(r) = \delta(r)$ , donde  $L$  es un operador lineal diferencial y  $\delta(r)$  es la delta de Dirac. Al ser  $(\nabla^4 - k^4)$  un operador lineal diferencial, debe por tanto existir una función de Green que cumpla (3). Las soluciones a (4) son  $k = \pm \sqrt{\omega}(\rho h/D)^{1/4}$ ;  $k = \pm i\sqrt{\omega}(\rho h/D)^{1/4}$ , que son ondas propagantes hacia atrás y hacia adelante y ondas evanescentes. Estas ondas son descritas generalmente por funciones de Hankel y Bessel (modificadas en el caso de las evanescentes), y el argumento de estas funciones tendrán como módulo  $|k| = \sqrt{\omega}(\rho h/D)^{1/4}$ . Tomando esto, la función de Green solución de (3) será

$$G_0(\vec{r}) = \frac{i}{8k^2} [H_0(kr) + \frac{2i}{\pi} K_0(kr)] \quad (5)$$

siendo  $H_0$  la función de Hankel de orden cero y  $K_0$  la función de Bessel modificada del segundo tipo y orden cero. La solución del campo  $W_1(r)$  tras una excitación  $\psi_0$  se saca de la ecuación

$$W_1(\vec{r}) = \psi_0(\vec{r}) + \sum_{\alpha} T_{\alpha} \psi_e(\vec{R}_{\alpha}) G_0(\vec{r} - \vec{R}_{\alpha}), \quad (6)$$

en la que  $\psi_e(R_{\alpha})$  es el campo incidente en el scatterer  $\alpha$ .

Tras operar suponiendo un único scatterer y resolviendo la ecuación de ondas [? ], llegamos a la siguiente expresión.

$$T_{\alpha} = \frac{t_{\alpha}}{1 - it/(8k^2)} \quad (7)$$

Nótese que para esta expresión se ha utilizado la función de Green  $G_0(0) = i/(8k^2)$ . Tras esto, podemos plantear el siguiente sistema de ecuaciones

$$\sum_{\beta} [\delta_{\alpha\beta} - (1 - \delta_{\alpha\beta}) T_{\beta} G_0(\vec{R}_{\alpha} - \vec{R}_{\beta})] \psi_e(\vec{R}_{\beta}) = \psi_0(\vec{R}_{\alpha}). \quad (8)$$

La solución a este sistema de ecuaciones es lo que buscamos. Para ello, aplicamos la propiedad distributiva

$$\sum_{\beta} \delta_{\alpha\beta} \psi_e(\vec{R}_{\beta}) - \sum_{\beta} (1 - \delta_{\alpha\beta}) T_{\beta} G_0(\vec{R}_{\alpha} - \vec{R}_{\beta}) \psi_e(\vec{R}_{\beta}) = \psi_0(\vec{R}_{\alpha}) \quad (9)$$

y tomando

$$M_{\alpha\beta} = T(1 - \delta_{\alpha\beta}) G_0(\vec{R}_{\alpha} - \vec{R}_{\beta}) \quad (10)$$

podemos reducir la expresión a

$$(I - M) \psi_e = \psi_0. \quad (11)$$

Por último, tomando

$$|\vec{R}_{\alpha} - \vec{R}_{\beta}| = r \quad (12)$$

podemos resolver la función de Green (5) con las funciones de Hankel y Bessel (apartado 3.4). Esto nos permite además resolver analíticamente (10), y por tanto simular el scattering múltiple.

### 2.3. Tipos de onda

Hace falta distinguir además entre dos tipos de ondas flexurales: las ondas flexurales planas (aquellas que inciden en un lateral de la red de resonadores) y las ondas flexurales puntuales (aquellas que son fruto de una perturbación a cierta frecuencia de uno de los resonadores). La *onda puntual* representa una fuente localizada en una posición específica  $r_s$ , que emite ondas en todas direcciones. Este tipo de onda es útil para estudiar la respuesta del sistema ante una excitación localizada. Por otra parte, la *onda plana* es una onda con una dirección de propagación definida y una fase uniforme a lo largo de los planos perpendiculares a esa dirección. tiene la forma:

$$\psi_0(\mathbf{r}) = e^{i\mathbf{k} \cdot \mathbf{r}}, \quad (13)$$

donde  $k$  es el vector de onda. Este tipo de excitación es especialmente útil para estudiar propiedades globales del sistema.

La estructura de la red de resonadores juega un papel clave en el scattering de la onda, así como el lugar de incisión inicial de esta (en caso de tratarse de una onda plana) o el resonador que se altera (en caso de tratarse de una onda puntual).

### 2.4. Problema de molinos de viento. Coherencia de fase

Antes de seguir, es importante plantearse una pregunta: ¿podemos asegurarnos que la fase entre los diferentes caminos que pueda tomar la onda se mantiene bien definida para que la interferencia sea coherente? En el caso de un sistema en el que algún scatterer actuase de manera impredecible, como por ejemplo rotando, la fase de la onda se perdería, y por tanto las interferencias de los scatterers dejarían de ser válidas para nuestra simulación. Esto es lo que se denomina *el problema de molinos de viento*, y en caso de no resolverlo, todo el método empleado para la simulación de scattering múltiple perdería su validez, dado que éste se basa en que la interferencia de las interacciones resulta en un promedio que puede ser simulado. Por lo tanto, hará falta verificar en el código creado que la fase de la onda se mantenga coherente.

## 3. Desarrollo del modelo computacional

En esta sección se describe cómo se ha implementado numéricamente el modelo de *scattering* múltiple sobre una red de dispersores, utilizando el entorno de programación MATLAB. El objetivo es simular la propagación de ondas y su interacción con múltiples resonadores situados sobre una placa.

El desarrollo del modelo computacional se divide en distintas etapas: la generación de la red de dispersores, la construcción de la matriz de interacción, la definición de la onda incidente, la resolución del sistema lineal y el cálculo del campo total. Cada una de estas fases se detalla en los subapartados siguientes, prestando especial atención a los aspectos numéricos y a las decisiones de implementación adoptadas durante el desarrollo del código.

Para la estructura y escritura del código, en este proyecto, se ha elegido un modelo modular que implemente todas estas etapas mediante funciones auxiliares en MATLAB. Estas serán luego controladas en un código principal, donde se decidirán los parámetros más generales de la simulación. Este aspecto es meramente organizativo.

### 3.1. Generación de la red de scatterers

La generación de la red de scatterers constituye la base física sobre la cual se propaga la onda incidente y donde tiene lugar el fenómeno de scattering múltiple. Esta red se construye mediante un conjunto de funciones auxiliares que permiten generar configuraciones geométricas de interés físico, como redes cuadradas o hexagonales. Estas configuraciones permiten el estudio de fenómenos como los *estados borde*.

En esta sección del código se definen también los parámetros físicos de la placa en estudio [I], tales como su espesor, distancia entre scatterers y propiedades elásticas del material.

Desde el punto de vista computacional, la red se representa mediante una matriz de posiciones  $\mathbf{R}$ , que almacena las coordenadas de cada scatterer. Mientras que las coordenadas de redes simples como las cuadradas o lineales pueden generarse directamente mediante bucles, la construcción de redes más complejas —como aquellas inspiradas en estructuras cristalinas reales— requiere un enfoque más elaborado.

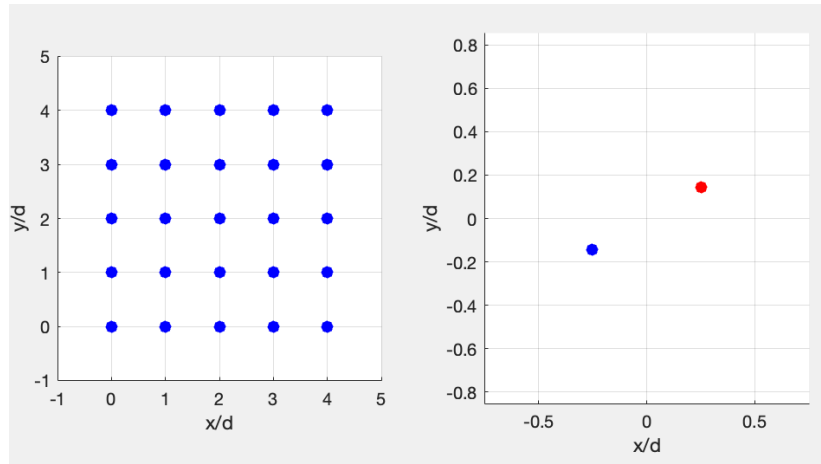


Figura 1: Posiciones de los scatterers sobre el plano. A la izquierda se muestra la generación de la red cuadrada; a la derecha, las posiciones de los dispersores  $\alpha$  y  $\beta$  en una celda unitaria, descritas por las ecuaciones 14 y 15.

En este trabajo, empleamos principalmente redes basadas en la estructura del grafeno, conocidas como redes tipo *honeycomb* o en panal de abeja, debido a su rica simetría y su relevancia en la propagación de ondas elásticas. Estas redes permiten explorar fenómenos tales como los comentados anteriormente.

Para describir este modelo numéricamente, se parte de una celda unitaria definida por dos vectores de red:

$$\vec{a}_1 = \begin{bmatrix} d \\ 0 \end{bmatrix}, \quad \vec{a}_2 = \begin{bmatrix} d \cos\left(\frac{\pi}{3}\right) \\ d \sin\left(\frac{\pi}{3}\right) \end{bmatrix} \quad (14)$$

donde  $d$  es el parámetro de red, es decir, la distancia entre scatterers adyacentes.

Cada celda contiene dos scatterers (análogos a los átomos de carbono en el grafeno), situados en:

$$\vec{R}_\alpha = \frac{d}{2\sqrt{3}} \begin{bmatrix} \cos\left(\frac{\pi}{6}\right) \\ \sin\left(\frac{\pi}{6}\right) \end{bmatrix}, \quad \vec{R}_\beta = -\vec{R}_\alpha \quad (15)$$

Repeticiones traslacionales de esta celda, mostrada en 1, permiten construir la red completa de la Figura 2 de forma controlada. Esto permite estudiar estructuras con bordes definidos, como los bordes zigzagueantes, sin necesidad de implementar condiciones de contorno explícitas.

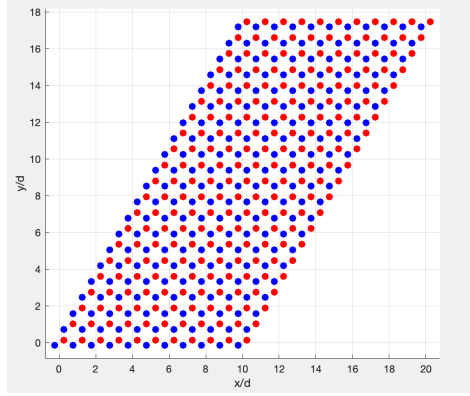


Figura 2: Red ordenada completa de tipo *honeycomb*, que simula la disposición de átomos de carbono en grafeno.

Por otra parte, también contamos con generación de redes *caóticas* o desordenadas en esta simulación, como la que se muestra en la Figura 3 pues serán de gran utilidad a la hora de estudiar el *problema de los molinos de viento*.

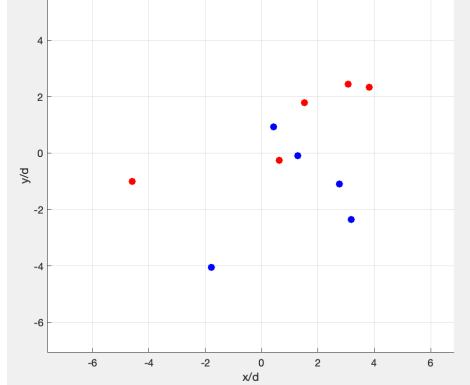


Figura 3: Ejemplo de generación de red desordenada. Las posiciones de los scatterers se generan aleatoriamente siguiendo el sistema definido en el código [II].

### 3.2. Construcción de la matriz del sistema

Volviendo a la ecuación (11), el objetivo de este apartado del código es crear una matriz

$$A = (I - M) \quad (16)$$

que cumpla (11). Con esto, tendríamos un sistema de ecuaciones que puede resolverse sin mayor dificultad con Matlab. Juntando (11) y (16)

$$A\psi_e = \psi_0. \quad (17)$$

Para ello, debemos primero calcular el número de resonadores, que es sencillamente el número de columnas de la matriz  $R$ , definida anteriormente como la red de scatterers. Apoyándose de una función

auxiliar, se calculan luego  $t_\alpha$  (2) y  $T_\alpha$  (7) simplemente sustituyendo en sus debidas fórmulas los parámetros de nuestra red [III].  $T_\alpha$ , por tanto, resultará ser una matriz  $1 \times N$ , siendo  $N$  el número de resonadores. El siguiente paso será crear una matriz de distancias entre resonadores, lo cual resultará en una matriz *dist* de tamaño  $N \times N$  [IVa], y cuya diagonal principal estará compuesta únicamente por ceros, dado que la distancia entre un resonador y sí mismo debe ser necesariamente cero. Para comprobar esto, se muestra en pantalla la suma de toda su diagonal, que, en caso de no haber ningún error, debe ser necesariamente cero. [IVb]

Una vez calculada la matriz de distancias, se puede proceder con el cálculo de la función de Green. Volviendo a (5), y teniendo ya la matriz de distancias entre resonadores *dist* y el número de onda  $k$  podemos calcular fácilmente la función de Green, que resultará ser nuevamente una matriz de tamaño  $N \times N$ , con la única sutileza que para su diagonal principal hemos de definir todos sus elementos como 0, debido a que en estos puntos la distancia es 0 (como se explicó previamente) y la función de Green diverge si no le das unas condiciones de contorno adecuada [V]. Para comprobar que esto se ha realizado adecuadamente, nuevamente se suma la diagonal y se comprueba que esta suma da cero [VI]. Antes de pasar a los últimos pasos, debemos comprobar que la matriz  $T_\alpha$  sea, efectivamente, una matriz fila y las operaciones que siguen sean correctas. Para esto, se comprueba con un bucle *for* que  $T_\alpha$  sea una matriz fila, y, en caso de que sea una matriz columna, se transpone para que sus dimensiones sean correctas [VII].

Habiendo calculado todo lo anterior, la ecuación (11) se vuelve resoluble. Basta con multiplicar en Matlab  $T$  con la delta de Kronecker, que en Matlab se puede expresar como la resta entre una matriz de unos de tamaño  $N \times N$  y una matriz identidad de tamaño  $N \times N$ ; y luego multiplicarlo por nuestra función de Green *hayada* previamente. La matriz  $M_{\alpha\beta}$  será, por tanto, una matriz con diagonal nula. Habiendo resuelto todo lo anterior, hallar la matriz  $A$  (16) que buscamos y que es clave en la resolución de (11) es trivial [VIII]. La única complicación que puede resultar de esto es que nuestra matriz identidad  $I$  empleada en (16) tenga un tamaño distinto a  $N \times N$ , o que la matriz  $M$  arrastre algún número no nulo en su diagonal. Para comprobar que esto no ha sucedido, se muestra la dimensión de  $A$  y la multiplicación de su diagonal principal en la pantalla, cuyos resultados deberían ser  $N \times N$  y 1, respectivamente [IX].

### 3.3. Modelado de la onda incidente

En el formalismo del método de scattering múltiple, el vector  $\psi_0$  contiene los coeficientes de la onda incidente sobre cada scatterer. La elección de la onda incidente es crucial para definir las condiciones iniciales del sistema. Existen dos enfoques típicos para modelar esta onda: como una *onda puntual* o como una *onda plana*. Estas ya han sido discutidas en el marco teórico de este trabajo.

Para modelar una onda puntual, se considera únicamente un scatterer de la red, siendo este excitado inicialmente. En la práctica, esto se implementa construyendo el vector  $\psi_0$  como un vector de ceros, salvo en la componente correspondiente al scatterer excitado, donde se asigna el valor 1, lo que equivale a una fuente localizada que emite ondas desde una única posición. Más adelante se discutirá como este tipo de ondas es útil para evitar problemas de simetría en estructuras regulares. La implementación es directa: basta con conocer el índice del scatterer excitado [X].

Para simular una onda plana incidente desde el infinito, se considera una función del tipo 13 con  $r$  como la posición del scatterer, y  $k = k(\cos\theta, \sin\theta)$  como el vector de onda definido por la dirección de incidencia  $\theta$ . El vector  $\psi_0$  se construye evaluando esta expresión en cada posición de scatterer [XI]. Este tipo de excitación permite estudiar la respuesta del sistema ante una onda con dirección bien definida.

Al utilizar ondas planas en estructura con cierta simetría, puede aparecer el llamado *problema de los molinos de viento*. Desde el punto de vista computacional, esto se traduce en un vector  $\psi_0$  cuyas componentes con casi linealmente dependientes o nulas, lo que deteriora la calidad de la solución del sistema lineal asociado al modelo de scattering.

Para evitar este problema, una estrategia eficaz ha sido utilizar una onda puntual, que rompe la simetría de forma natural. Otra alternativa sería introducir una ligera perturbación en el ángulo de la onda plana para desalinearla de las direcciones de simetría más críticas.

### 3.4. Resolución del sistema y evaluación del campo total

Continuando con el código del apartado anterior, el siguiente paso para continuar con la simulación es resolver el sistema lineal planteado anteriormente (17). Resolver este sistema resulta trivial con un código de Matlab [XII]. Para comprobar que no haya habido ningún error ni en la generación de la onda ni en la simulación del scattering, se comprueba que las dimensiones de tanto la onda incidente como la onda externa son las adecuadas,  $N \times 1$ , y se muestran en pantalla [XIII].

Una vez que se tenemos las soluciones que se necesitan, se puede seguir creando una malla donde calcular el campo. Para esto, se definen unos ejes donde se realizará la simulación y una resolución a la que se verá nuestra simulación. Cuando tenemos esto, basta con definir otro bucle for que irá calculando gradualmente el campo  $W_1(\vec{r})$  para cada punto, empleando la ecuación (6) [XIV].

### 3.5. Visualización del resultado

Finalmente, para visualizar los resultados obtenidos de las simulaciones se grafica el módulo del campo complejo  $|W|$  en el plano bidimensional normalizado por la distancia  $d$ . Se genera una superficie tridimensional en la que el eje vertical representa la magnitud del campo y para ver claramente la posición de los resonadores, se superponen marcadores de negro, marcando en rojo el resonador fuente si se trata de una onda puntual [XV]. Rotando esta figura en el programa podemos observar distintas perspectivas de la simulación, siendo a la vez una representación tridimensional del campo y un *mapa de calor* (véase figura 4), facilitando así distintos tipos de análisis del fenómeno.

Mediante ese análisis, esta representación permite identificar visualmente patrones de interferencia, modos localizados y regiones de alta intensidad, lo cual es especialmente útil para estudiar efectos como resonancias o estados ligados al borde.

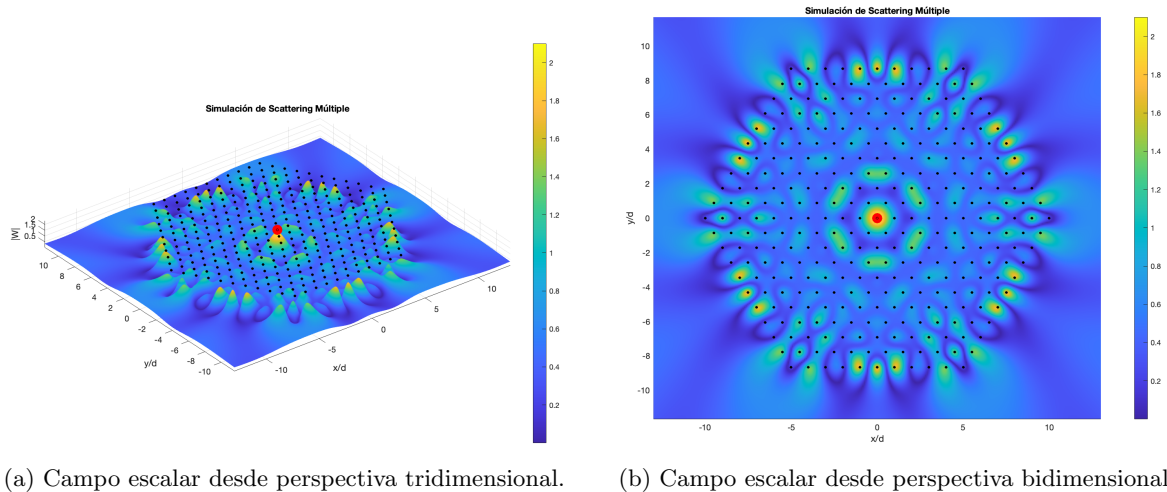


Figura 4: Visualización del módulo del campo  $|W|$  desde dos perspectivas distintas de la simulación. Se simula una red de *scatterers* hexagonal, sin bordes zigzagües, y se señala en rojo el dispersor fuente de la onda puntual.

## 4. Resultados y validación

Con el objetivo de evaluar la validez del modelo de scattering múltiple implementado y caracterizar su comportamiento en distintos casos, se han diseñado una serie de simulaciones que permiten analizar configuraciones controladas y escenarios físicamente relevantes. Esta sección se divide en dos



partes principales: por un lado, se presentan resultados obtenidos de situaciones simples, que permiten contrastar la simulación con predicciones teóricas conocidas; por otro, se exploran configuraciones más complejas en las que se manifiestan efectos físicos de interés, como la propagación guiada en estructuras periódicas o el problema de los molinos de viento.

Estos casos no solo permiten validar la implementación numérica desarrollada, sino también poner en evidencia su capacidad para reproducir fenómenos característicos del sistema físico modelado, en particular aquellos asociados a propiedades simétricas susceptibles al *molino de viento*.

#### 4.1. Red lineal

Se considera el caso de dos dispersores alineados, separados por una distancia  $\lambda = \frac{2\pi}{k}$ , con una onda incidente plana propagándose a lo largo del eje  $x$ . Este caso es adecuado para observar la interferencia de las ondas dispersadas en la misma dirección que la onda incidente. Debido a que la separación entre los scatterers es exactamente un múltiplo de la longitud de onda, ambos reciben la onda incidente con la misma fase.

Las ondas dispersadas por cada uno se suman constructivamente en la dirección de propagación, generando un patrón de intensidad reforzada hacia delante. La simulación reproduce este comportamiento, mostrando dos máximos claramente definidos a la derecha de cada dispersor, lo cual es coherente con la interferencia constructiva hacia delante esperada para esta configuración.

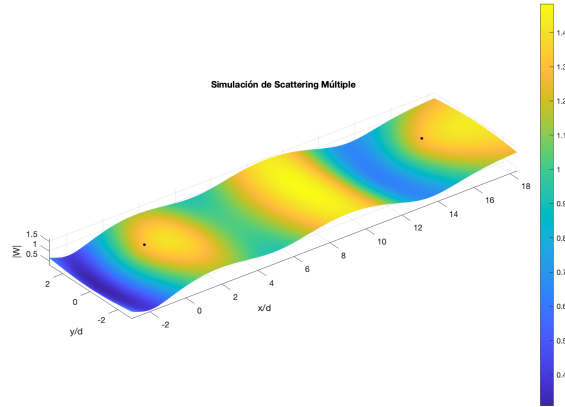


Figura 5: Caso simple de scattering múltiple con alineación lineal y separación resonante.

#### 4.2. Simetría radial

Como siguiente caso de validación y análisis del comportamiento del modelo, se ha implementado una simulación con una fuente puntual situada en el centro de una red cuadrada de dispersores distribuidos regularmente. La fuente emite una onda saliente que se propaga radialmente hasta encontrar los primeros scatterers.

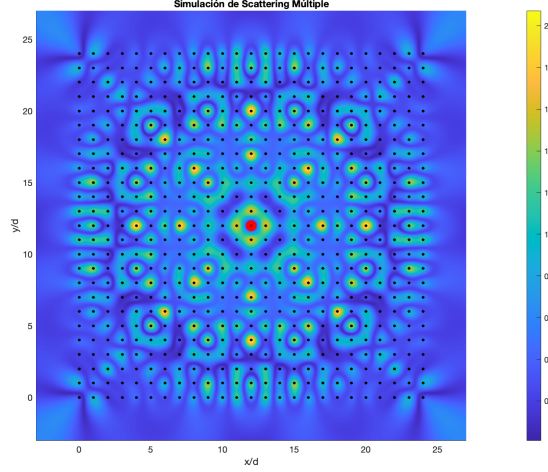


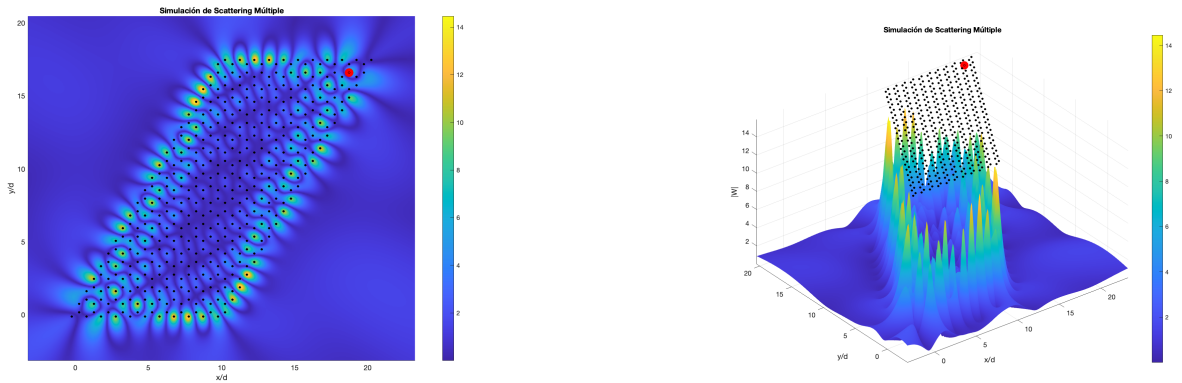
Figura 6: Visualización de scattering múltiple por onda puntual en el centro de una red cuadrada. Se observan patrones de simetría radial en la amplitud del campo.

Este escenario, ilustrado en las Figuras 4 y 6 permite comprobar que el modelo es capaz de manejar adecuadamente una fuente puntual en un entorno no trivial. Además, la simetría del patrón observado en regiones lejanas respecto al centro confirma la consistencia entre la geometría de la red y la propagación resultante validando la implementación desde un punto de vista simétrico.

### 4.3. Efecto borde

En este caso, se ha simulado el comportamiento de una onda puntual incidente sobre una red tipo *honeycomb*, como las que han sido discutidas a lo largo de este trabajo y la que se ha representado en la Figura 2. La propagación de la onda se ve fuertemente influenciada por la geometría de la red, dando lugar a *estados borde*.

Replicar este efecto en redes físicamente realistas es sumamente complicado por lo que, para poder simularlo, hemos tomado como referencia el trabajo descrito en [1]. En este artículo, centrado en la aplicación de redes ordenadas con el fin de propagar controladamente ondas elásticas, se toman unos parámetros específicos para poder recrear este fenómeno en redes realistas similares al grafeno.



(a) La perturbación inicial se propaga efectivamente por el borde de la red.

(b) La amplitud del campo en el borde es mucho mayor que en el resto del espacio.

Figura 7: Simulación de *scattering* múltiple en una red finita de resonadores de tipo *honeycomb*. La onda incidente puntual se aplica en la esquina superior derecha de la red, lo que genera -a ciertas frecuencias de resonancia- una propagación por los bordes zigzagueantes de la geometría.

A medida que la onda incide en la red, en la Figura 7 se observa que, debido a la estructura y la disposición de los scatterers, la propagación se concentra principalmente a lo largo de los bordes del paralelogramo. Los dispersores situados en las zonas periféricas de la red actúan como barreras que redirigen la onda, produciendo reflexiones y difracciones significativas. En particular, la onda no penetra fácilmente en el centro de la red, lo que refleja el marcado *efecto borde* característico de estas estructuras.

Este comportamiento es un claro ejemplo de cómo los bordes de una red pueden canalizar la propagación de ondas, limitando su difusión en el interior de la red y creando patrones complejos de interferencia en las cercanías de los límites. La simulación demuestra que el modelo es capaz de capturar este fenómeno, reproduciendo un patrón de *scattering* que refuerza la propagación a los bordes y muestra una propagación mínima hacia el centro.

#### 4.4. *Cluster* caótico y problema de los *molinos de viento*

Finalmente, se ha simulado un caso con un conjunto aleatorio de scatterers distribuidos dentro de una región circular, emulando un entorno desordenado similar al problema conocido como el de los *molinos de viento*. Esta red es la que aparece simulada en la Figura 3 (claro que al ser aleatoria tendrá forma diferente en cada simulación). En este caso, se hace incidir una onda plana sobre el *cluster* con el objetivo de observar cómo la presencia de múltiples obstáculos distribuidos aleatoriamente afecta a la propagación.

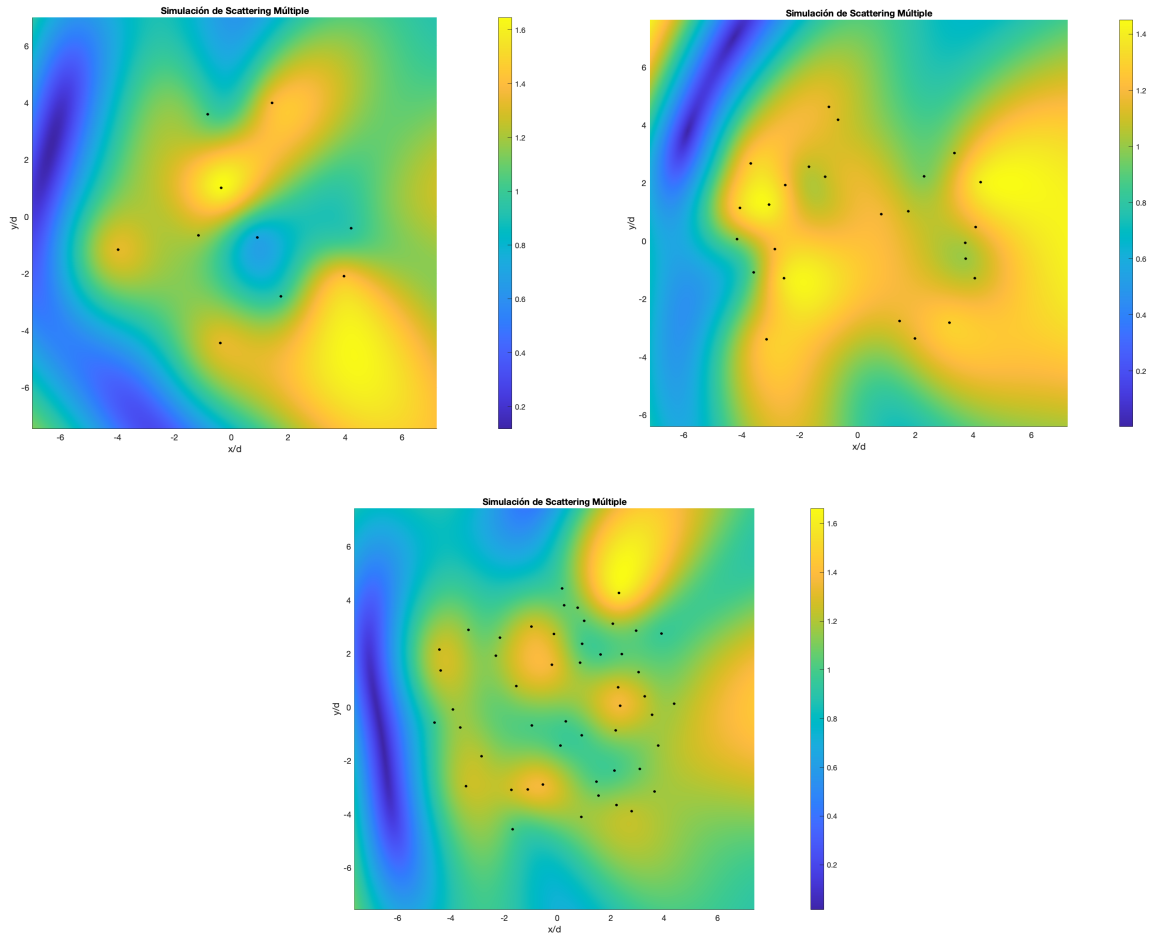


Figura 8: Distintos casos de clusters aleatorios de resonadores, a los que les incide una onda plana que se propaga a lo largo del eje  $x$ .

Los resultados muestran que, a diferencia de los casos anteriores con geometrías regular, la onda experimenta una dispersión múltiple compleja, con zonas de interferencia, acumulación de energía y regiones de sombra. En muchos casos la onda es incapaz de atravesar el cluster de forma directa, desviándose lateralmente o siendo reflejada hacia atrás. Esta fuerte alteración del frente de onda es una manifestación característica del fenómeno que se pretende estudiar, donde la presencia de múltiples resonadores bloquea parcialmente la propagación.

Este escenario valida que el modelo puede tratar entornos sin simetría, con scattering múltiple intenso, y reproduce correctamente los efectos esperados del *problema de los molinos de viento*.

## 5. Discusión

Tras haber realizado todas estas simulaciones, podemos pasar a comentar aspectos interesantes de cada una de ellas, así como las limitaciones computacionales que se han encontrado. En primer lugar, resulta interesante observar los patrones de interferencia generados por las redes con alto nivel de simetría, como pueden ser las hexagonales o cuadradas (ver Figuras 4b y 6, respectivamente), que seguirán un patrón claramente visible. Esta simetría se producirá de manera radial, lo cual coincide con nuestro modelo teórico. Podemos afirmar, por tanto, que la simulación ha resultado exitosa en este aspecto.

Cabe recalcar un pequeño matiz con respecto a las condiciones de contorno: en la Figura 6, podemos observar que la onda se termina propagando por los "vértices" del cuadrado que forma la red de resonadores. Esto se debe a que, como se comentó en el apartado 2.1, no hemos definido unas condiciones de contorno estrictas para nuestra simulación, por lo que la onda puede seguir propagándose libremente una vez escapa de la red de resonadores.

Esta aproximación es válida debido a que el modelo se centra en calcular el campo total como la superposición de la onda incidente y las ondas dispersadas por un conjunto discreto de resonadores. Como las posiciones de los dispersores están definidas explícitamente, y el resto del medio se considera homogéneo e infinito, se asume implícitamente que las ondas pueden escapar hacia el infinito sin interacción adicional.

Sin embargo, al trabajar con una red finita de dispersores, aparecen de forma natural ciertos *efectos borde*, ya que la onda no puede seguir propagándose más allá de los límites de la estructura. Estos efectos se manifiestan, por ejemplo, en la aparición de reflexiones, zonas de interferencia y posibles modos localizados en los extremos de la red. Aunque no se imponen condiciones de contorno rígidas o absorbentes, la geometría finita de la red actúa como un contorno efectivo que influye en la propagación. En esta simulación se puede comprobar que la aparición del efecto borde se da bajo determinadas condiciones. Esto se puede ver claramente en la Figura 7a. Bajo este efecto borde, se puede apreciar que la onda se propaga en mucha mayor medida por los bordes de la red que por la parte interior. El interior de la red permanece casi aislado de las ondas. Este efecto es algo que sigue nuestro modelo teórico, por lo que la simulación es un éxito en este otro aspecto.

## 6. Conclusiones

Se ha presentado el marco teórico de la evolución de una onda flexural al atravesar una placa fina cuya superficie está cubierta por una red de resonadores, distinguiendo entre ondas flexurales planas y puntuales, así como entre diferentes estructuras de la red de scatterers. Se ha desarrollado un modelo en MATLAB que generaba estas redes y la onda a estudiar, resolvía los sistemas de ecuaciones planteados en el marco teórico, evaluaba el campo total y representaba el resultado de los cálculos tridimensionalmente. En esta representación se pueden observar las características físicas que esperadas, como el efecto borde o la simetría radial (según el caso), lo que permite concluir que las simulaciones han sido exitosas.

No obstante, es importante señalar que estas simulaciones son en un plano ideal y puramente teórico: no se tienen en cuenta posibles pérdidas de energía, interferencias externas o materiales con propiedades no lineales. Aunque estos aspectos se podrían mejorar para describir de manera más adecuada modelos físicos reales, la simulación presentada es una aproximación suficientemente exacta. Por poner un ejemplo, la estructura de *honeycomb* (Fig. 2, 7a) es análoga a la del grafeno [2].

En definitiva, el estudio del scattering múltiple mediante simulaciones computacionales resulta una aplicación muy valiosa de las herramientas que proporciona MATLAB.

## 7. Agradecimientos

Agradecemos a la Universidad Autónoma de Madrid por la licencia de MATLAB, utilizada para la mayor parte de este proyecto, así como a nuestro profesor José Vicente Álvarez por la ayuda proporcionada en clase a la hora del planteamiento del proyecto y la programación de los códigos.

## 8. Bibliografía

### Referencias

- [1] Daniel Torrent, Didier Mayou, and José Sánchez-Dehesa. Elastic analog of graphene: Dirac cones and edge states for flexural waves in thin plates. *Physical Review B: Condensed Matter and Materials Physics*, 87(11):115143, 2013.
- [2] Antonios Gonis and William H. Butler. *Multiple Scattering in Solids*. Springer Science & Business Media, 2012.

## 9. Anexos

```
params.rho = 2700;      % Densidad de la placa (kg/m³, aluminio)
params.h = 0.01;       % Espesor de la placa (m)
params.E = 70e9;       % Módulo de Young (Pa)
params.nu = 0.33;      % Coeficiente de Poisson
params.kR = 500;       % Rigidez del resorte (N/m)
% Rigidez flexural (D = E·h³ / 12(1-ν²))
params.D = params.E * params.h^3 / (12*(1 - params.nu^2)); % ≈6.54 Pa·m³
%params.D = 6.54;

% --- Parámetros de la red ---
params.d = 1;          % Parámetro de red

% --- Resonadores ---
params.m = params.D * 10; % Masa de cada resonador (kg)
params.gamma = 10;

% --- Frecuencias y número de onda ---
params.Omega = 2.589; % Frecuencia normalizada (adimensional, Ωa = 2.589)
params.omegaR = sqrt(params.Omega * params.D / (params.rho * params.h)); % Frecuencia de resonancia de los scatterers (rad/s)
%params.omegaR = sqrt((params.kR / params.m) * (params.rho * params.d^2 * params.h / params.D));
%params.omegaR = 2.5;
params.k = sqrt(params.Omega) * ((params.rho * params.h / params.D)^(1/4)); % Número de onda
%params.k = sqrt(params.Omega) / sqrt(params.d); % Número de onda normalizado

% --- Parámetros de simulación ---
params.field_resolution = 300; % Resolución de la malla para el campo (puntos por eje)
params.visualize = true; % Visualizar resultados automáticamente
```

Anexo I: Código conteniendo todos los parámetros necesarios.

```
% Generación de posiciones
for i = 1:N
    while true
        % Generar posición candidata
        angle = 2*pi*rand();
        radius = cluster_radius*sqrt(rand());
        candidate = radius * [cos(angle); sin(angle)];

        % Verificar distancia mínima
        valid = true;
        for j = 1:i-1
            if norm(candidate - R(:,j)) < min_distance
                valid = false;
                break;
            end
        end

        % Aceptar posición si es válida
        if valid
            R(:,i) = candidate;
            break;
        end

        % Control de intentos máximos
        attempts = attempts + 1;
        if attempts > max_attempts
            warning('No se pudo colocar el resonador %d/%d', i, N);
        end
    end
end
```

Anexo II: Código para crear una red caótica.

```

function T = compute_T_coefficients(R,params)
% Calcula t y T para todos los resonadores
omega = params.Omega;
omegaR = params.omegaR;

t = (params.m / params.D) * (omegaR^2 * omega^2) / (omegaR^2 - omega^2);
T = t ./ (1 - 1i * t / (8 * params.k^2));

T = T * ones(1, size(R, 2));

% Imprimir valores clave
disp('=== Parámetros de Scattering ===');
disp(['t: ', num2str(t)]);
disp(['T: ', num2str(T)]);
disp(['size T: ' num2str(size(T))]);

end

```

Anexo III: Cálculo de  $t$  y  $T_\alpha$ .

```
size distance matrix:462  462
```

(a) Comprobación del tamaño de la matriz de las distancias.

```
diag: 0
```

(b) Comprobación de la diagonal de la matriz de las distancias.

Anexo IV: Comprobaciones adicionales de la matriz de distancias para evitar errores arrastrados.

```
G(r == 0) = 0; % Anular auto-interacción (diagonal)
```

Anexo V: Definición de la diagonal de la función de Green como 0 para evitar divergencia.

```
=== Matriz G ===
```

```
diag: 0
```

Anexo VI: Comprobación de la diagonal de la función de Green.

```

% Asegurar que T sea un vector fila para multiplicación correcta
if iscolumn(T)
    T = T';
end

```

Anexo VII: Comprobación de que  $T$  es una matriz fila.



```

% Para onda incidente plana
theta = 0; % dirección de propagación (0 = eje x)
k = params.k;
kvec = k * [cos(theta); sin(theta)];

% Definir la fuente
% Fuente puntual
% psi0(idx_source) = 1;
% Onda plana incidente
% Evaluar la onda plana en cada resonador
psi0 = exp(1i * (kvec' * R)); % R es 2xN, psi0 es 1xN
% Asegurarse de que psi0 sea columna
psi0 = psi0.'; % ahora psi0 es Nx1
idx_source = []; % no hay una fuente puntual en este caso

```

Anexo XI: Coeficientes de  $\psi_0$  para simulación de onda plana.

```

% Multiplicar cada columna j de G_matrix por T(j)
interaction_matrix = T .* ((ones(num_resonators) - eye(num_resonators))) .* G_matrix;

% Construir matriz A
A = eye(num_resonators) - interaction_matrix;
end

```

Anexo VIII: Cálculo de la matriz  $A$ .

```

=== Matriz A ===
size: 462 462
diag: 1

```

Anexo IX: Comprobación del tamaño y diagonal de la matriz  $A$ .

```

% Posición
% Encontrar el resonador con máxima coordenada x e y
| [~, idx_source] = max(R(1, :) + R(2, :));

% Resonador central para red cuadrada
% idx_source = round(length(R)/2);

% Encontrar un resonador concreto similar al del paper
% coord_obj = [18.75; 16.5988]; % Coordenadas objetivo del resonador
% distancias = vecnorm(R - coord_obj, 2, 1); % Distancia euclídea columna a columna
% [~, idx_source] = min(distancias);

```

Anexo X: Búsqueda del índice de la posición del scatterer para simular la onda puntual.

```

% Resolver sistema lineal
psi_e = A \ psi0;
disp('=== Onda externa ===');
disp(['size: ' num2str(size(psi_e))]);

```

Anexo XII: Resolución del sistema lineal 17.

```

=== Onda incidente===
size: 331    1
=== Onda externa ===
size: 331    1

```

Anexo XIII: Comprobación del tamaño de ambas ondas,  $\psi_e$ ,  $\psi_0$ .

```

% Calcular el campo total en una malla
x = linspace(min(R(1,:)) - 3, max(R(1,:)) + 3, params.field_resolution);
y = linspace(min(R(2,:)) - 3, max(R(2,:)) + 3, params.field_resolution);
[X, Y] = meshgrid(x, y);
W1 = zeros(size(X));

for n = 1:length(R)
    r = sqrt((X - R(1,n)).^2 + (Y - R(2,n)).^2);
    W1 = W1 + psi_e(n) * GreenFunction(params.k, r);
end

% Añadir la onda incidente (fuente puntual)
r_source = sqrt((X - R(1, idx_source)).^2 + (Y - R(2, idx_source)).^2);
W1 = W1 + GreenFunction(params.k, r_source);

```

Anexo XIV: Resolución de  $W_1$  con la ecuación 6.

```

% Calcular coordenadas normalizadas
d = params.d;
Rx = R(1,:)/d;
Ry = R(2,:)/d;

% Crear malla de coordenadas
padding = 3/d;
xvec = linspace(min(Rx)-padding, max(Rx)+padding, params.field_resolution);
yvec = linspace(min(Ry)-padding, max(Ry)+padding, params.field_resolution);

% Visualización 3D del campo
surf(xvec, yvec, abs(W1), 'EdgeColor', 'none');
colormap(ax, p.Results.Colormap);
if p.Results.Colorbar
    colorbar;
end

% Añadir resonadores
hold on;
height = max(abs(W1(:))) * 1.1;
plot3(Rx, Ry, height*ones(size(Rx)), 'ok',...
      'MarkerSize', 3,...
      'MarkerFaceColor', 'k');

if isempty(idx_source) == 0
    % Destacar fuente
    plot3(Rx(idx_source), Ry(idx_source), height, 'or',...
          'MarkerSize', 12,...
          'MarkerFaceColor', 'r');
end

% Ajustes finales
%view(p.Results.ViewAngle);
axis tight equal;
title('Simulación de Scattering Múltiple');
xlabel('x/d'); ylabel('y/d'); zlabel('|W|');
shading interp;
end

```

Anexo XV: Gráfico de la simulación del scattering múltiple.