

# Enseñanza de Acústica Física con Simulaciones en k-wave

## Teaching of Physical Acoustics with Simulations in k-wave

Jorge Petrosino<sup>#1</sup>, Lucas Landini<sup>#</sup>, Georgina Lizaso<sup>#2</sup>, Ian Kuri<sup>#</sup>, Ianina Canalis<sup>#</sup>

<sup>#</sup>Departamento de Humanidades y Artes. Universidad Nacional de Lanús  
29 de Septiembre 3901, Remedios de Escalada, Buenos Aires, Argentina

<sup>1</sup> jorgepetrosino@gmail.com

<sup>2</sup> georgina.lizaso@gmail.com

Recibido: 07/12/18; Aceptado: 11/04/19

**Abstract**— Using k-wave toolbox as teaching aid in subjects related to physical acoustics is proposed. This toolbox enables simple and efficient simulation of time domain wave propagation. It is widely used for simulating the behaviour of ultrasonic waves in biological media. By adding a small number of instructions it is possible to simulate acoustic field propagation for two-dimensional conditions, based on a reference image (BMP file) in which points of different colours represent specific elements within the simulation. This enables experimentation of different situation using the k-wave toolbox with minimal MATLAB/Octave know-how. Any alteration on the images by simple cut, paste, move or rotate operation results in new simulation conditions.

**Keywords:** acoustic teaching; numerical simulations; pseudospectral method.

**Resumen**— Se presenta el uso del toolbox k-wave para apoyar la enseñanza de acústica. Este toolbox permite de modo simple y eficiente la simulación de propagación de ondas en el dominio del tiempo. Es ampliamente utilizado para simular el comportamiento de ondas ultrasónicas en medios biológicos. Mediante el agregado de un limitado número de instrucciones es posible simular la propagación de campos acústicos en dos dimensiones con diversas condiciones de contorno, basadas en una imagen de referencia (archivo BMP) en el cual puntos de diferentes colores representan elementos específicos de la simulación. Esto permite experimentar con diferentes condiciones utilizando el toolbox k-wave con mínimos conocimientos de MATLAB/Octave. Cualquier alteración en la imagen mediante operaciones simples de corte, pegado, desplazamiento o rotación produce nuevas condiciones de simulación.

**Palabras clave:** enseñanza de acústica; simulación numérica; método pseudoespectral.

### I. INTRODUCCIÓN

k-wave es un toolbox de código abierto desarrollado para MATLAB [1] que permite realizar simulaciones en el dominio del tiempo sobre la propagación de ondas acústicas en 1D, 2D y 3D. Tiene una gran funcionalidad dado que utiliza modelos numéricos avanzados que pueden dar cuenta tanto de propagación lineal como alineal, con una distribución arbitraria de parámetros heterogéneos del material y con distintos modelos de absorción acústica.

Existe una gran cantidad de métodos numéricos para resolver las ecuaciones diferenciales que gobiernan la propagación de ondas mecánicas. Este toolbox utiliza el método pseudoespectral del espacio k [2]. Soluciona las derivadas espaciales obteniendo series de Fourier mediante el método de colocación [3] y las combina con un proceso de propagación temporal expresado en el dominio espacial de frecuencias (el espacio k). Este método, a diferencia de los métodos de diferencias finitas o elementos finitos, requiere el uso de grillas con puntos igualmente espaciados que representan la zona en la cual se propagarán las ondas.

Al calcular la distribución espacial de ondas numéricamente mediante FFT (Fast Fourier Transform) se obtienen soluciones en las que el campo ondulatorio resultante es periódico. Esto provocaría que una onda que se propagase hacia la derecha saliendo de la zona de análisis volviese a aparecer por la izquierda. Para evitar este tipo de comportamiento el k-wave agrega una pequeña franja absorbente en la zona de trabajo, sobre el borde externo (PFM, por Perfectly Matched Layer)

El toolbox k-wave fue desarrollado inicialmente por el Photoacoustic Imaging Group de la University College London (UCL). Existen numerosas publicaciones con reportes de análisis del modelo de simulación y con aplicaciones a diferentes campos. Ha tenido un desarrollo particular en la simulación de espejos de inversión temporal trabajando con ultrasonido para aplicaciones biomédicas [4] [5]. El código, manual de instalación, publicaciones y otra información útil se encuentra en [www.k-wave.org](http://www.k-wave.org).

Uno de los modos establecidos en el toolbox para definir los distintos parámetros que intervendrán en la simulación (densidad del medio, velocidad de propagación, ubicación de fuentes y de sensores, entre otros) es mediante matrices que tienen la misma dimensión que la grilla de base (llamada kgrid). De esta manera se dispondrá de una grilla de base que especificará cuestiones generales como el tamaño del espacio de simulación y su resolución, otra representará la ubicación de las fuentes, una tercera indicará la ubicación de los sensores que registrarán variables en función del tiempo, a lo que podrán agregarse grillas que especifiquen la densidad del medio en cada punto. La Fig. 1 muestra un esquema de esta estructura de funcionamiento.

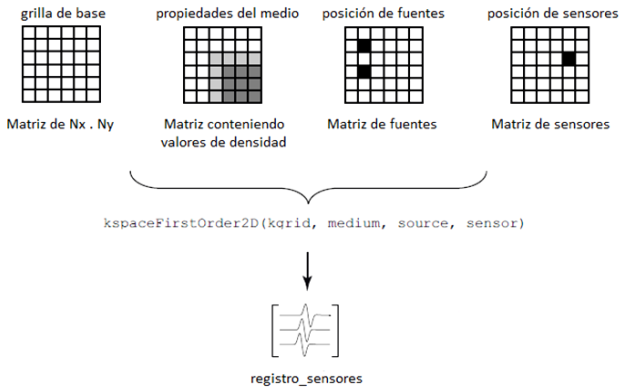


Fig. 1. Esquema de definición de parámetros y resultados de simulación..

La función *k-spaceFirstOrder2D()* incluida en *k-wave* devuelve un arreglo con tantas filas como sensores y tantas columnas como instantes de tiempo de simulación se hayan definido.

Podríamos resumir su funcionamiento en sucesivas etapas. En el primer instante el sistema toma como referencia las características del medio y las fuentes para hallar la distribución espacial de ondas compatible con dichas condiciones. Terminado este proceso grafica la solución espacial en pantalla y guarda la información de los puntos especificados en los sensores para ir construyendo las señales temporales de salida. Luego aplica el proceso de propagación hallando la distribución de presiones y velocidades correspondientes al siguiente instante, grafica este resultado en pantalla y guarda el registro de los sensores. La función repite este proceso para todos los instantes hasta completar el tiempo especificado.

Durante el proceso de resolución, el sistema muestra una animación en pantalla del modo en que se propagan las ondas a medida que se va aplicando la solución. Al terminar genera registros temporales de los sensores utilizados.

El toolbox *k-wave* suele ser utilizado principalmente para estudios de propagación de ondas mecánicas en medios heterogéneos en aplicaciones biomédicas, pero su adaptación para obtener soluciones de ondas mecánicas en el rango audible que se propagan por el aire resulta sencilla.

La propuesta que se presenta describe el uso de una función específicamente desarrollada para que estudiantes con poca o nula experiencia en programación o uso de MATLAB, puedan simular distintas situaciones acústicas con *k-wave*

## II. ADAPTACIÓN PARA SU USO EN ENSEÑANZA

La función desarrollada permite al usuario el diseño de la estructura de simulación trabajando sobre una imagen en forma gráfica, codificando los distintos elementos a utilizar mediante colores preestablecidos. Las partes blancas de la imagen corresponden a un medio homogéneo (aire) en el que se podrán colocar fuentes de presión sonora (variables en el tiempo), objetos de reflexión total (obstáculos o paredes) y sensores que registren presión en función del tiempo. La Fig. 2 muestra un ejemplo con una fuente de presión representada por el punto rojo, un obstáculo (negro) y cuatro sensores (verde), colocados con la intención de explorar los efectos de la difracción frente a obstáculos.

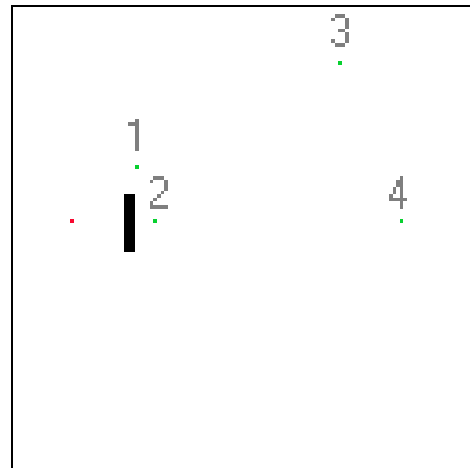


Fig. 2. Imagen de base para ilustrar el fenómeno de difracción

La función denominada *simulaImagen256()* lee el archivo de imagen, construye en base a este archivo las distintas grillas que requiere el *k-wave* y desarrolla la simulación haciendo visible la animación que puede ser grabada en formato video en forma simultánea. Las variables de salida de esta función son los registros temporales de los sensores. Cualquier sector de la imagen en color gris resulta ignorado por el simulador, lo que permite incorporar información que puede facilitar la interpretación del proceso. Los números de sensores de la Fig. 2 fueron escritos en la imagen de base en color gris, de manera que aparecen en la simulación pero son transparentes a efectos de la propagación de ondas.

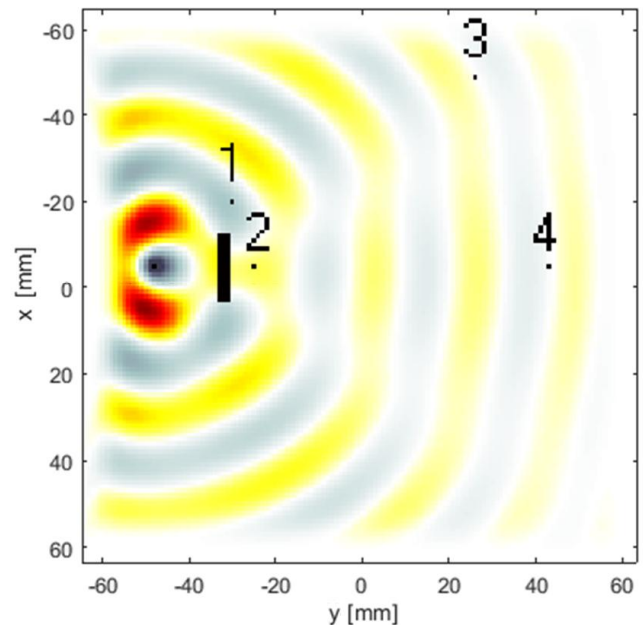


Fig. 3. Simulación de la disposición de fuentes, sensores y obstáculos utilizando *k-wave* con Octave

La Fig. 3 muestra la distribución espacial de frentes de ondas para  $t = 317 \mu s$ . La Fig. 4 corresponde al registro temporal de los sensores. Resulta interesante notar que tanto los sensores 1 y 2, como los sensores 3 y 4 están a distancia similar de la fuente. La diferencia de nivel detectado entre cada par de sensores se debe a la existencia del obstáculo en el camino que los une con la fuente. Puede observarse que el sensor 2 que se ubica justo detrás del obstáculo registra

una señal con un nivel significativamente menor que el sensor 1, mientras que el sensor 3 y el 4 tienen menor diferencia de nivel entre ellos. Con la longitud de onda utilizada la sombra acústica provocada a esa distancia tiene menor impacto.

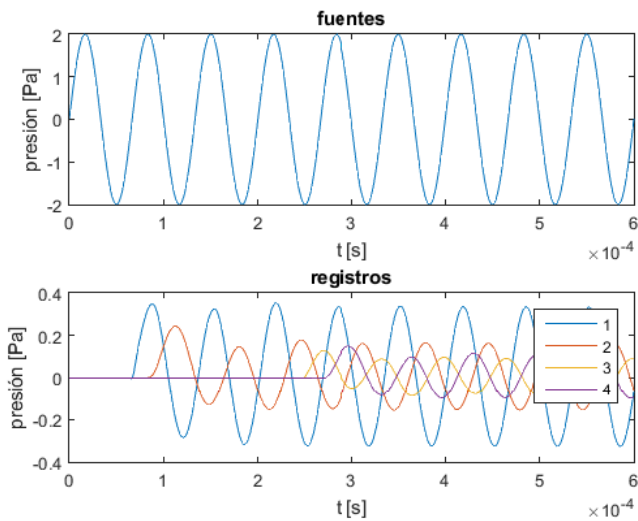


Fig. 4. Registros temporales de presión sonora en los tres sensores

Con esta herramienta de simulación de gráficos, un estudiante principiante puede probar distintas configuraciones simplemente modificando la imagen con cualquier editor. Al guardar la imagen y ejecutar *simulaImagen256()* obtendrá los resultados de la nueva simulación. En los ejemplos que presentamos aquí hemos utilizado el Paint de amplia disponibilidad en las computadoras que utilizan Windows. Por su parte, los estudiantes intermedios o avanzados podrán modificar el código de la función *simulaImagen256()* para incorporar otras características al medio antes de la simulación.

### III. DESCRIPCIÓN DE PROCESOS DE LA FUNCIÓN *SIMULAIMAGEN256()*

La función recibe como argumentos el nombre del archivo de imagen, un factor de escala (lado del cuadrado mínimo de la grilla en metros), la duración de la simulación en segundos, un valor lógico para indicar si se desea registrar la simulación en video, el valor de la velocidad del sonido a utilizar y la fuente.

Se utilizan imágenes de 256 colores. Los píxeles de la imagen de color rojo (valor 79) se interpretan como fuentes, los píxeles de color verde (valor 113) corresponden a sensores y los píxeles de color negro (valor 0) representan elementos perfectamente reflectantes.

Es posible utilizar múltiples fuentes y múltiples sensores simultáneamente sin que esto tenga relación directa con el tiempo de cómputo.

La variable de entrada "fuente" es una estructura que contiene varios campos (tipo, amplitud, frecuencia, cantidad de ciclos, duración). La manera más directa de especificar el desarrollo temporal de las fuentes consiste en escribir directamente la ecuación a evaluar utilizando t como la variable que representa al tiempo. Esto se realiza igualando el campo "fuente.tipo" a una cadena de caracteres con el texto de la ecuación a evaluar. La Ec. (1) muestra un ejemplo para generar una fuente que emita una señal

compuesta por la suma de una sinusoidal de amplitud 3 pascales y frecuencia 2 kHz más una sinusoidal de amplitud 1 pascal y frecuencia 10 kHz.

$$fuente.tipo = '3*\sin(2*\pi*2000*t) ... + 10*\sin(2*\pi*10000*t)'; \quad (1)$$

Se han implementado además tres tipos de fuentes que pueden ser especificadas mediante su denominación. Estas corresponden a un impulso, ruido blanco y una ráfaga de n ciclos de la frecuencia especificada. Cada una de ellas requiere parámetros en algún otro campo de la variable fuente. Una fuente impulso requiere especificar su amplitud asignando un valor al campo "fuente.amplitud", tal como se muestra en la Ec. (2)

$$fuente.tipo = 'impulso'; \quad (2) \\ fuente.amplitud=2;$$

Al utilizar 'ruidoBlanco' es necesario especificar una amplitud y una duración del ruido que debe ser menor a la duración total de la simulación (en segundos), pensada para ver lo que sucede al apagar la fuente, tal como se muestra en la Ec. (3)

$$fuente.tipo = 'ruidoBlanco'; \quad (3) \\ fuente.duración=1e-4; \\ fuente.amplitud=2;$$

Si se decide utilizar el tipo de fuente 'nCiclos', será necesario además definir la cantidad de ciclos y la frecuencia en sendos campos adicionales, tal como muestra la Ec. (4)

$$fuente.tipo = 'nCiclos'; \quad (4) \\ fuente.amplitud=2; \\ fuente.n = 3; \\ fuente.f0=4000;$$

Todas las fuentes requieren un campo más para su especificación correspondiente a "fuente.modos", que puede tener dos valores: 'additive' o 'dirichlet'. Una fuente de presión de tipo aditiva provocará un incremento de presión en el punto en que se encuentre por encima del valor que corresponda al desarrollo de la simulación, mientras que una fuente tipo 'dirichlet' obligará a que en ese punto exista el valor de presión especificada. En cierto tipo de experimentos puede requerirse utilizar una u otra. El principal hecho a tener en cuenta es que si durante su propagación una onda se enfrenta a una fuente tipo 'dirichlet' se producirán reflexiones. Se recomienda utilizar el modo 'additive' para las simulaciones generales de uso educativo, reservando el modo 'dirichet' para casos específicos en que se pretenda comprobar las diferencias al especificar las condiciones de contorno de esta manera.

El toolbox k-wave permite utilizar múltiples fuentes, y que cada una de ellas tenga una definición temporal independiente. La adaptación que hemos hecho permite también múltiples fuentes, pero todas ellas tendrán el mismo desarrollo temporal por una cuestión de simplicidad de uso. Esto significa que todas las fuentes emiten en fase.

Luego de ejecutada la simulación, la función *simulaImagen256()* devuelve diversas variables:

- *datos\_sensor*, que es un arreglo con tantas filas como sensores y tantas columnas como instantes de tiempo en la simulación,
- *t*, que es un vector correspondiente a la variable tiempo,
- *dt*, que es el intervalo mínimo de discretización temporal,
- *lx*, que es la longitud en metros en el eje x del espacio simulado,
- *ly*, que es la longitud en el eje y del espacio simulado.

Estas variables pueden resultar útiles para graficar o procesar los resultados obtenidos durante el proceso.

La función *simulaImagen256()* fue diseñada para operar con archivos de imagen BMP de 256 colores. Esto se debe a que se necesita trabajar con imágenes sin compresión y se pretende utilizar un formato muy sencillo y de amplia cobertura en diferentes editores. Tanto los procesos de compresión como los de modificación del tamaño de la imagen pueden alterar la codificación de los colores, dando por resultado comportamientos no deseados en la simulación.

#### IV. EJEMPLOS DE USO CON ONDAS EN TUBOS

Como ejemplo de la sencillez de uso de la herramienta se simuló la propagación de ondas en tubos, con diferentes condiciones de borde. Para ello partimos de una imagen de base como la que se muestra en la Fig. 5.

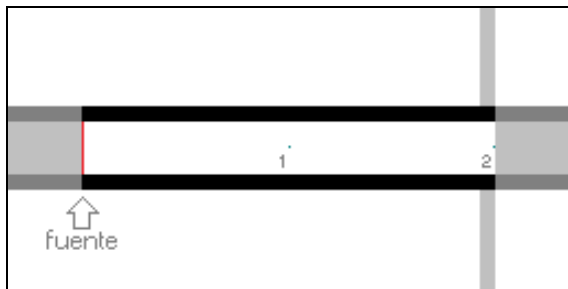


Fig. 5. Imagen propuesta como base para las simulaciones de tubos

La fuente es la línea roja ubicada hacia la izquierda de la Fig. 5. Se agregan dos sensores (verde), uno en el centro y otro contra el extremo derecho. Las zonas negras representan paredes perfectamente reflectantes. Los grises que se observan a los costados sólo están allí para poder "pintarlos" antes de iniciar la simulación con color negro o color blanco permitiendo realizar cambios rápidos de situaciones que representen extremo cerrado, extremo abierto, extremo abierto con pestaña o tubo de longitud infinita.

Los grises son ignorados para la simulación por la función *simulaImagen256()*, aunque sí aparecen en las imágenes de fondo de la animación, por lo que es posible colocar allí números o palabras que faciliten la interpretación de la animación y los gráficos posteriores.

Como ya se mencionó previamente k-wave tiene en sus bordes una condición de absorción *PML*, por lo cual si el tubo llega a tocar los extremos de la zona definida para la simulación será interpretado como un tubo de longitud infinita en esa dirección.

#### A. Propagación en un Tubo Infinito con Extremo Cerrado

Pintando de modo conveniente las zonas grises de la imagen de base (blanco para aire, negro para paredes) pueden generarse las condiciones de un tubo infinito con un extremo cerrado. La Fig. 6 muestra un instante de la simulación posterior a la reflexión en el extremo cerrado. Los colores cálidos representan niveles de compresión y los fríos de rarefacción (presión sonora positiva y negativa respectivamente).

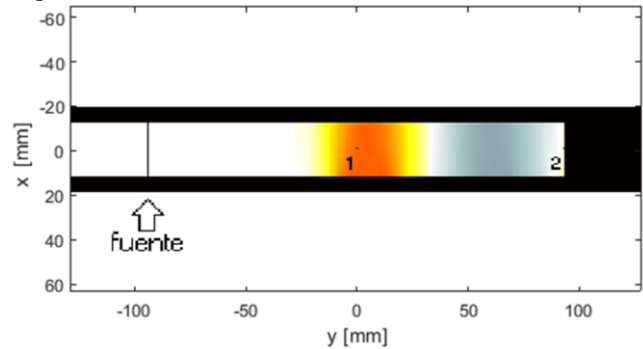


Fig. 6. Propagación de ondas en un tubo infinito con extremo cerrado

La Fig. 7 muestra el desarrollo temporal de los dos sensores. El trazo azul corresponde al sensor 1 colocado en mitad de la imagen, mientras que el trazo rojo corresponde al sensor 2 ubicado junto al extremo cerrado.

Puede observarse allí que la onda de presión se refleja sin inversión en el extremo cerrado. Puede notarse además que el nivel de presión instantánea en el sensor 2 es mayor ya que al encontrarse junto al extremo cerrado hay superposición entre la onda progresiva y la regresiva.

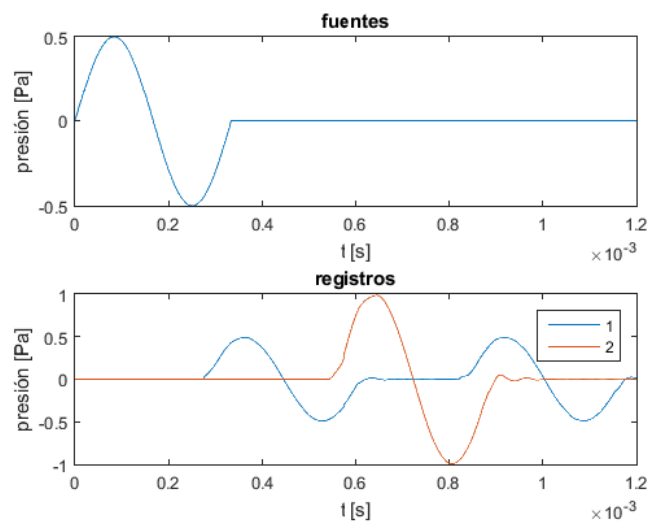


Fig. 7. Diagramas temporales de la fuente y los sensores

#### B. Propagación en un Tubo Infinito con Extremo Abierto

La Fig. 8 muestra la simulación de un tubo infinito con extremo abierto en un instante posterior a la reflexión. Allí puede notarse la diferencia de fase existente en la propagación de ondas que se observan viajando hacia la izquierda en el interior y en el exterior del tubo.



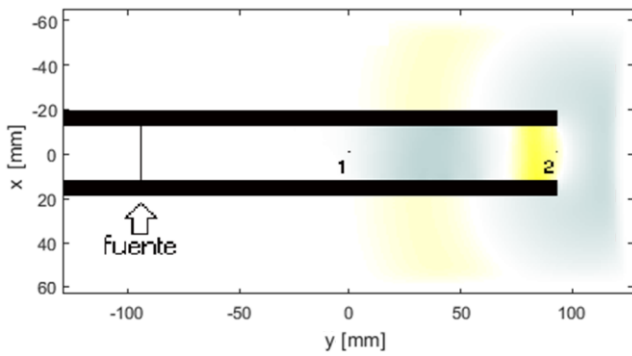


Fig. 8. Propagación de ondas en un tubo infinito con extremo abierto

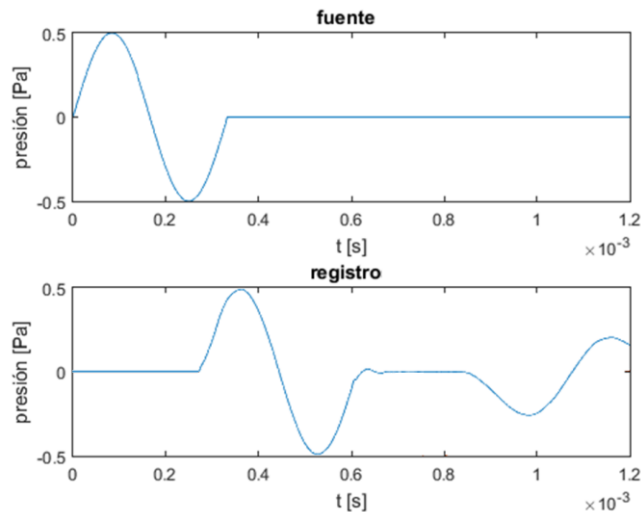


Fig. 9. Diagramas temporales de la fuente (arriba) y el sensor 1 (abajo)

Observando el registro del sensor 1 puede notarse claramente que la onda de presión se refleja con inversión (Fig. 9). Se verifica además que la onda reflejada tiene menos energía que la incidente (debido a la energía que se transmite al exterior del tubo por el extremo abierto).

### C. Comparación de Resultados por Elementos Finitos

Con el fin de comparar estos resultados con otras herramientas de uso más frecuente en acústica, se simularon situaciones similares con tubos mediante el método de elementos finitos utilizando COMSOL Multiphysics. La Fig. 10 muestra el resultado de simular un tubo infinito con un extremo cerrado. El trazo azul corresponde a la fuente y el trazo verde a un sensor separado del extremo. Se observa en color verde el registro de la onda progresiva y de la regresiva con el mismo nivel de amplitud.

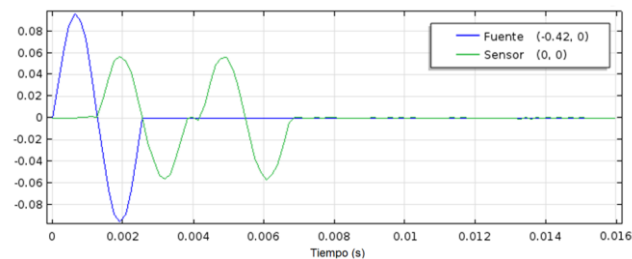


Fig. 10. Resultados de la simulación por elementos finitos de la reflexión en un extremo cerrado.

La simulación de un tubo con un extremo abierto mediante COMSOL puede resolverse de diversas maneras. La Fig. 11 muestra los resultados cuando se intenta modelar el extremo abierto imponiendo como condición de contorno que exista presión cero en el extremo del tubo. En este caso puede notarse que la onda se refleja con inversión, pero sin pérdida de energía. Esta es una solución consistente con la condición de contorno impuesta, pero no representa adecuadamente la situación física de un extremo abierto. Se muestra además el resultado superpuesto obtenido mediante k-wave para el mismo tubo abierto, más adecuada a lo que sucede físicamente en esa situación.

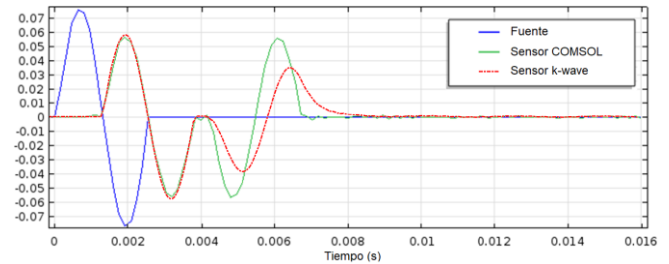


Fig. 11. Resultados comparados entre k-wave y COMSOL utilizando un extremo con presión igual a cero como condición de contorno.

La Fig. 12 representa los resultados de agregar una cámara de aire luego del extremo abierto del tubo. En esta situación la reflexión se invierte con una caída de nivel por la energía radiada hacia el exterior del tubo. Distintos tamaños de cámara de aire producen resultados diferentes, pero más cercanos a la situación física correspondiente a un extremo abierto (semejantes a lo obtenido por k-wave).

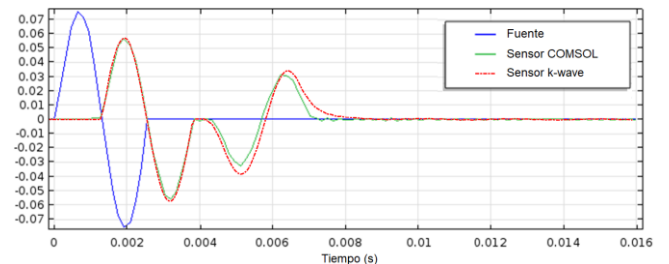


Fig. 12. Resultados comparados entre k-wave y COMSOL utilizando un extremo comunicado con una cámara de aire externa al tubo.

### V. CÓDIGO DE LOS EJEMPLOS PRESENTADOS

El código necesario para realizar una simulación como las reportadas más arriba requiere pocas líneas y es relativamente sencillo. La Fig. 13 muestra el caso de la simulación de un tubo infinito-abierto.

```
%Ejemplo simulación tubo infinito-abierto
nombreArchivoImagen='TuboInfAbierto.bmp';
escala=1e-3;
duracion=1.2e-3;
registraVideo=false;
c0=344;
fuente.tipo='nCiclos';
fuente.amplitud=0.5;
fuente.f0=3000;
fuente.n=1;
fuente.modo='additive';
[datos_sensor, t, dt, ecuacion, lx, ly] = ...
simulaImagen256(nombreArchivoImagen, escala, ...
duracion, registraVideo,c0,fuente);
```

Fig. 13. Código del ejemplo utilizando la función simulaImagen256() para simular un tubo infinito-abierto.

Se requiere especificar el nombre del archivo de imagen, la escala (m), duración (s), una variable lógica que indica si se desea el registro en video, velocidad del sonido (m/s), las características de la fuente, agregado finalmente una línea que llama a la función *simulImagen256()*. La estructura es la misma para cualquier simulación. Las modificaciones principales corresponden a utilizar otra imagen, o elegir un tipo de fuente diferente.

Los resultados pueden graficarse de forma libre por parte del usuario, ya que la función *simulImagen256()* entrega sus resultados como variables de salida que quedan disponibles en la memoria de trabajo de MATLAB. En la Fig. 14 se muestra el código necesario para obtener el gráfico temporal superpuesto de los dos sensores utilizados en la Fig. 8.

```
%Gráficos de resultados de la simulación
plot(t,datos_sensor);
xlabel('t [s]')
ylabel('presión [Pa]')
legend('1','2')
```

Fig. 14. Código para graficar el desarrollo temporal de los sensores utilizados en la simulación incluyendo leyendas de ejes y referencias.

Cualquier modificación realizada en la imagen respecto de las condiciones de borde del tubo, o de la ubicación de fuentes o sensores sirve de información para que la función *simulImagen256()* genere los parámetros necesarios para una simulación diferente.

## VI. CONCLUSIONES

El trabajo propone el uso del toolbox k-wave como un modo de obtener simulaciones útiles para la enseñanza de acústica física. Con el fin de simplificar la manera de trabajar en estas simulaciones hemos incorporado una función que permite definir la estructura de lo que se desea simular a partir de construir una imagen. Esto permite que los estudiantes, al menos en una primera aproximación, puedan concentrarse en el problema físico que desean estudiar sin necesidad de ocuparse de los algoritmos de la simulación.

Se muestran solamente unos pocos ejemplos relacionados con las reflexiones en distintas condiciones de contorno para tubos. Sin embargo la herramienta resulta muy versátil para explorar diversos fenómenos como interferencia, difracción y difusión.

La comparación con simulaciones realizadas mediante elementos finitos resultan consistentes, aunque es claro que para utilizar estos modelos de elementos finitos se requiere una mayor experiencia que la que podría esperarse de estudiantes iniciales o intermedios de Acústica Física.

El código de la función *simulImagen256()*, junto con ejemplos de uso y archivos de imágenes para probar los ejemplos se encuentran disponibles en [6].

## AGRADECIMIENTOS

Los autores de este trabajo quieren agradecer especialmente al equipo conformado por Andrés Bonino Reta, Damian Andrés Fernández y Nicolás Casais Dassie por sus aportes durante el proceso.

El presente trabajo surgió como resultado de un proyecto de investigación actualmente en desarrollo en la Universidad Nacional de Lanús que requiere utilizar simulaciones de propagación de ondas en k-wave.

## REFERENCIAS

- [1] B. E. Treeby, y B. T. Cox, "k-Wave: MATLAB toolbox for the simulation and reconstruction of photoacoustic wave fields", *Journal of biomedical optics*, vol. 15, n° 2, p. 021314, 2010
- [2] M. Tabei, T. D. Mast, y R. C. Waag, "A k-space method for coupled first-order acoustic propagation equations", *The Journal of the Acoustical Society of America*, vol. 111, n° 1, pp. 53-63, 2002
- [3] J. P. Boyd, *Chebyshev and Fourier spectral methods*. 2da ed., Nueva York, EEUU: Courier Corporation, 2001
- [4] B. T. Cox, S. Kara, S. R. Arridge, y P. C. Beard, "k-space propagation models for acoustically heterogeneous media: Application to biomedical photoacoustics", *The Journal of the Acoustical Society of America*, vol. 121, n° 6, pp. 3453-3464, 2007
- [5] B. T. Cox, y P. C. Beard, *Modeling photoacoustic propagation in tissue using k-space techniques*, 1era. ed., L. V. Wang, Ed. Boca Ratón, EEUU: CRC Press, 2009
- [6] Repositorio correspondiente al usuario GLizaso en [www.github.com](http://www.github.com), accesible mediante el enlace <https://tinyurl.com/y2da8noy>.