

MEDIOS ACTIVOS

Resolución numérica del problema de resonancia mediante Matlab

1. Introducción

La ecuación que define las resonancias del sistema de tres capas con medios a , b y c , donde sólo b es un medio activo, está dada por:

$$\rho = e^{i4\pi\eta_{bx}d'} \quad (1)$$

donde ρ se define como:

$$\begin{aligned} \rho_s &= \frac{(\eta_{bx} + \eta_{ax})(\eta_{bx} + \eta_{cx})}{(\eta_{bx} - \eta_{ax})(\eta_{bx} - \eta_{cx})} \\ \rho_p &= \frac{(\eta_{bx} + N_a\eta_{ax})(\eta_{bx} + N_c\eta_{cx})}{(\eta_{bx} - N_a\eta_{ax})(\eta_{bx} - N_c\eta_{cx})} \end{aligned} \quad (2)$$

para el modo S y P respectivamente. Los vectores $\boldsymbol{\eta}_a, \boldsymbol{\eta}_b, \boldsymbol{\eta}_c$ se corresponden con los vectores \mathbf{k} adimensionalizados mediante el factor $2\pi/\lambda_o$. Todos ellos poseen la misma componente η_z en la dirección z (dirección tangente a las interfaces), debido a las condiciones de contorno en las interfaces. Por lo tanto, las proyecciones de los vectores $\boldsymbol{\eta}$ en la dirección x pueden calcularse a partir del módulo de $\boldsymbol{\eta}$ y de η_z como:

$$\begin{aligned} \eta_{ax} &= \sqrt{\eta_a^2 - \eta_z^2} \\ \eta_{bx} &= \sqrt{(\eta_{br} - i\eta_{bi})^2 - \eta_z^2} \\ \eta_{cx} &= \sqrt{\eta_c^2 - \eta_z^2} \end{aligned} \quad (3)$$

donde η_{br} se corresponde con la parte real y $\eta_{bi} > 0$ con la parte imaginaria (definida negativa explícitamente) del índice de refracción generalizado. Los parámetros N_a y N_c están dados por:

$$\begin{aligned} N_a &= \frac{(\eta_{br} - i\eta_{bi})^2}{\eta_a^2} \\ N_c &= \frac{(\eta_{br} - i\eta_{bi})^2}{\eta_c^2} \end{aligned} \quad (4)$$

Se desea resolver la ecuación (1) tomando como datos conocidos los valores de: $\eta_a, \eta_{br}, \eta_c, d'$ y, como incógnitas, a η_{bi} y η_z .

2. Solución numérica

Para la resolución numérica de la ecuación (1) se construye el sistema de dos ecuaciones que surge de la igualdad de módulos y de fases al aplicar la función log en ambos miembros de (1):

$$\begin{aligned} f_1(\eta_a, \eta_{br}, \eta_c, d', \eta_z, \eta_{bi}) &= \log |\rho| + 4\pi \operatorname{Im}(\eta_{bx})d' = 0 \\ f_2(\eta_a, \eta_{br}, \eta_c, d', \eta_z, \eta_{bi}, m) &= \arg |\rho| - 4\pi \operatorname{Re}(\eta_{bx})d' - 2\pi m = 0, \quad m \in Z \end{aligned} \quad (5)$$

donde surge una incógnita más, m , a partir de la ecuación de fases. Las incógnitas del sistema de ecuaciones se marcan en rojo y los parámetros seleccionados a priori, en azul. Para cada valor fijo de d' , pueden encontrarse múltiples valores de m, η_{bi}, η_z para los cuales se satisface el sistema. Algunas consideraciones a tener en cuenta son:

- $0 = \eta_{z0} < \eta_z < \eta_{z1} = \min(\eta_a, \eta_{br}, \eta_c)$
- $0 < \eta_{bi} < \eta_{br}$
- Se desea resolver (5) para distintos valores de d' (en el rango $1 < d' < 1 \times 10^4$).

Para resolver el sistema de ecuaciones (5), teniendo en cuenta estas consideraciones, se implementó en Matlab® el algoritmo descrito a continuación en pseudocódigo:

```
resonancia( $\eta_a$ ,  $\eta_{br}$ ,  $\eta_c$ ,  $ds$ )

  para cada  $d$  en  $ds$ 

     $\eta_{bi0}$  = fzero( $f_1(\eta_a, \eta_{br}, \eta_c, d, \eta_{z0}, \underline{\eta_{bi}})$ )
     $\eta_{bi1}$  = fzero( $f_1(\eta_a, \eta_{br}, \eta_c, d, \eta_{z1}, \underline{\eta_{bi}})$ )

     $m_0$  = fzero( $f_2(\eta_a, \eta_{br}, \eta_c, d, \eta_{z0}, \eta_{bi0}, \underline{m})$ )
     $m_1$  = fzero( $f_2(\eta_a, \eta_{br}, \eta_c, d, \eta_{z1}, \eta_{bi1}, \underline{m})$ )

     $ms$  = lista_enteros_entre( $m_0, m_1$ )

    para cada  $m$  en  $ms$ 

      res_parciales <- fsolve( $f_1(\eta_a, \eta_{br}, \eta_c, d, \underline{\eta_{bi}}, \underline{\eta_z}) = 0$ ,  $f_2(\eta_a, \eta_{br}, \eta_c, d, \underline{\eta_{bi}}, \underline{\eta_z}, m) = 0$ )

    resultados <- res_parciales

  devolver resultados
```

La función **resonancia** toma los valores de los parámetros $\eta_a, \eta_{br}, \eta_c$ y la lista ds de valores para d' .

Luego, para cada valor en la lista ds se buscan los valores η_{bi0} y η_{bi1} que solucionan la ecuación $f_1 = 0$ cuando η_z toma sus valores límite η_{z0} y η_{z1} , respectivamente. Esto se lleva a cabo mediante el uso del método **fzero** de Matlab que halla una raíz de una función, en términos de una única variable. En el pseudocódigo, se indica mediante subrayado la variable a partir de la cual **fzero** busca la raíz de la función (en el código de Matlab esto

se indica anteponiendo a la función el símbolo @ seguido de la variable). El resto de los parámetros de dicha función están fijos. El método **fzero** toma también como parámetro una condición inicial o, en su defecto, el intervalo en el cual se encuentra la raíz que se quiere hallar. En este caso se utilizó el intervalo $(0, \eta_{br})$, de acuerdo con las consideraciones previas.

A continuación, de manera análoga, se obtienen los valores m_0 y m_1 que solucionan la ecuación $f_2 = 0$ al fijar los valores de η_z y η_{bi} en (η_{z0}, η_{bi0}) y (η_{z1}, η_{bi1}) , respectivamente. En este caso se utilizó como condición inicial el valor $m = 0$.

A partir de los valores encontrados m_0 y m_1 se construye la lista ms de números enteros contenidos en el intervalo (m_0, m_1) .

Para cada elemento de la lista ms se resuelve el sistema de ecuaciones $f_1 = 0$ y $f_2 = 0$ en función de las variables η_z y η_{bi} . Esto se realiza mediante el método **fsolve** de Matlab. Este método toma como parámetro una condición inicial para cada incógnita. Inicialmente se emplearon como condición inicial los valores η_{z0} y η_{bi0} , pero luego se encontró que era necesaria una modificación en este punto (Ver la siguiente sección). Los resultados parciales se guardan en una matriz de 3 filas (la primer fila contiene los valores de η_{bi} , la segunda fila, los de η_z y la tercera, los de m) y tantas columnas como soluciones. Al finalizar el ciclo de m , esta matriz que contiene los resultados para cada valor de d' fijo, se vuelca como elemento en una celda de Matlab.

Este algoritmo se ejecuta 2 veces, una para la solución del modo perpendicular o S y otra para el modo paralelo o P , donde lo único que cambia en cada ejecución es la definición de ρ , de acuerdo a la ecuación (2).

3. Evaluación

Para la evaluación de la implementación del algoritmo en Matlab se emplearon las soluciones del problema obtenidas de manera independiente mediante el software MathCAD.

Se había partido de la hipótesis de que para valores fijos de d' y m entero en el intervalo (m_0, m_1) , el sistema de ecuaciones (5) poseía una única solución de η_z y η_{bi} . Durante la evaluación y comparación se encontró que, dado que las ecuaciones son no lineales, esto no es necesariamente cierto. En particular se encontró que para valores de d' pequeño existen casos donde para un mismo m pueden existir dos soluciones distintas. Por ejemplo, para el modo paralelo (P) con $\eta_a = 1.0$, $\eta_{br} = 1.3$, $\eta_c = 1.5$, $d = 2$, se encuentra que el único valor posible para m es $m = -4$. Con este valor de m se encuentran dos soluciones posibles para η_z y η_{bi} :

- $\eta_z = 0.6649$ y $\eta_{bi} = 0.1806$
- $\eta_z = 0.8715$ y $\eta_{bi} = 0.1695$

Ahora, el algoritmo descripto, debido a cómo funciona el método **fsolve** encuentra sólo una de estas dos soluciones. Qué solución encuentre, depende de las condiciones iniciales suministradas. Entonces, es posible hallar diversas soluciones resolviendo el problema

para distintas condiciones iniciales. En principio, si a priori no sabemos la cantidad de soluciones distintas que tiene el problema para los datos de entrada suministrados, no es posible hallar numéricamente todas las soluciones sin recurrir a algoritmos mucho más complejos, ya que no sabemos cuándo hemos hallado todas las soluciones.

Luego de evaluar diversas modificaciones al código se encontró que una manera simple de solucionar este problema es resolver el sistema de ecuaciones (5) mediante **fsolve** empleando condiciones iniciales de la forma: $\eta_z = \eta_{z0} + r(\eta_{z1} - \eta_{z0})$, $\eta_{bi} = \eta_{bi0} + r(\eta_{bi1} - \eta_{bi0})$, donde r es un parámetro a definir. Experimentando con diversos valores para r se observó que resolviendo dos veces el sistema de ecuaciones mediante **fsolve**, empleando $r = 0.4$ y $r = 0.9$ respectivamente, es posible recuperar cada una de las soluciones distintas para el caso expuesto previamente. Además, estos valores no sólo funcionan en este caso, sino que se observó que también hallan las soluciones distintas (para igual valor de m) para todos los sets de parámetros para los que se conocía la solución de manera independiente.

En los casos en que existe una única solución para cada m alguna de estas condiciones iniciales lleva a que **fsolve** no converja a una solución o que la solución no cumpla alguna de las consideraciones detalladas en la sección anterior (por ejemplo, se obtiene una solución con $\eta_z < 0$ o $\eta_z > \min(\eta_a, \eta_{br}, \eta_c)$). Entonces, antes de guardar las soluciones en la matriz de resultados parciales, se constata que la solución cumpla con las propiedades esperadas y, en caso contrario, es descartada.

En el caso de las soluciones con $d' = 200$ no se observó en ningún caso la existencia de más de una solución para cada m . Dado que el costo computacional del algoritmo crece con d' es deseable no agregar demasiado cómputo extra para solucionar el problema de la no unicidad de la solución, ya que impacta negativamente en la performance del algoritmo. Si fuera necesario, podría realizarse una evaluación del problema matemático para elegir algún valor umbral de d' a partir del cual se pueda asumir unicidad de la solución. También habría que ver si para otros valores bajos de d' la solución propuesta efectivamente encuentra siempre todas las soluciones existentes, o si se puede resolver este problema de alguna otra manera.

3.1. Comparación de resultados

A continuación se muestra la comparación de resultados entre las soluciones obtenidas mediante el algoritmo implementado en Matlab y el método de solución alternativo mediante el uso de MathCAD.

En el archivo adjunto `datos_comparacion.xls` se presentan las planillas con los datos obtenidos a partir de cada método. Los nombres de cada planilla se identifican con los valores de los parámetros de la siguiente forma: `na nbr nz modo d`.

Se puede observar que los resultados coinciden dentro del error determinado por el truncamiento de los datos en todos los casos. La única diferencia encontrada es que algoritmo de Matlab encuentra más soluciones que el de MathCAD en la mayoría de los casos para $d' = 200$, que corresponden para los mayores valores de m . Creo que esto se debe al redondeo que emplearon en MathCAD para definir el valor máximo de η_z . (por ejemplo, $\eta_{z\max} = 0.99$ para indicar $\eta_z < 1$, entonces se pierden todas las soluciones entre

(0.99, 1.0)).

Los casos con $d' = 2$ encuentran la misma cantidad de soluciones y su diferencia se encuentra dentro del error de truncamiento.

A continuación se presentan los gráficos obtenidos para los casos con $d' = 200$ donde se puede observar que las soluciones superpuestas de ambos métodos coinciden.

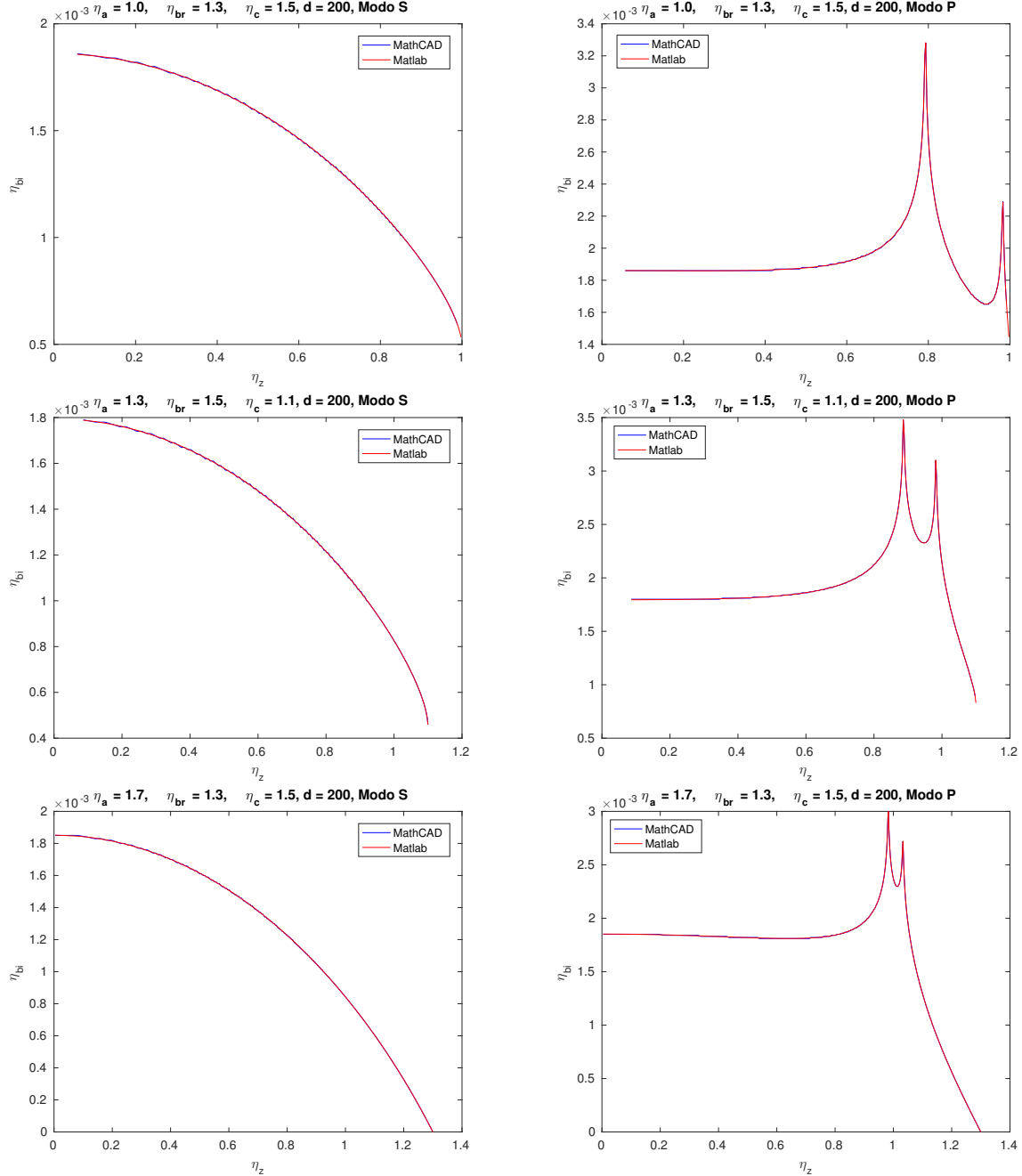


Figura 1: Comparación de soluciones entre los métodos empleando Matlab y MathCAD.