

APENDICE : rutina “sonicbands.m” en MATLAB para calcular bandas sonicas

Uso:

```
>> [k,ks,w]=sonicbands(d,a0,rhoh,rhoc,ch,cc,N,M)
```

Calcula la frecuencias propias del sonido en un sistema periodico infinito que consiste en una red cuadrada de cilindros (solidos o liquidos) en un medio fluido.

Parámetros de entrada:

d: diametro de los cilindros (en cm)

a0: parametro de red (en cm)

rhoh,rhoc: densidad del medio exterior y de los cilindros (kg/cm³)

ch,cc: velocidad del sonido en el medio exterior y en los cilindros (m/s)

N+1: numero de ondas planas en cada direccion -> numero total de ondas planas=(N+1)². Advertencia: usar N par!

M: numero de (k,w) que calcula para cada camino -> numero total de (k,w)=3M+1

Variables de salida:

k: matriz de 2x(3M+1). Contiene los vectores k del camino usado en la primera zona de Brillouin

ks: vector de 3M+1 elementos. Contiene la longitud del camino recorrido en el espacio k.

w: matriz de (N+1)²x(3M+1). Contiene las frecuencias propias, en kHz.

Ejemplo tipico: (con los parámetros dados en la Introducción)

```
>> [k,w]=sonicbands(1.9,2.5,1.2,7.8e3,343,5790,22,10)
```

Calcula las bandas de una red de cilindros de acero macizo de 1.9 cm de diámetro, separados 2.5 cm, en aire. Se usan 529 ondas planas, y el eje k queda discretizado en 31 puntos (de Γ a Γ).

NOTA: como la diferencia entre densidades y velocidades del sonido entre los dos medios es tan grande, los resultados no dependen del valor exacto de estos parámetros. Así, por ejemplo, es de esperarse que estos resultados también valgan para tubos de aluminio en vez de acero, tubos huecos en vez de tubos macizos, etc.

Gráficos:

```
>> plot(1:31,w)
```

Grafica las bandas en función del índice de la discretización del camino en k. k=1 es el punto Γ , k=11 es el punto X, k=21 el punto M, y k=31 nuevamente el punto Γ .

```
>> plot(ks,w)
```

Lo mismo pero en funcion de la longitud del camino recorrido en k. $ks=0$ es el punto Γ , $ks=\pi$ es el punto X, $ks=2\pi$ es el punto M y $ks=(2+\sqrt{2})\pi$ es nuevamente Γ .

Rutina:

```
function [k,ks,w]=sonicbands(d,a0,rhoh,rhoc,ch,cc,N,M)

% d: diametro (en cm)
% a0: parametro de red (en cm)
% rhoh,rhoc: densidad del medio extrerior y de los cilindros (kg/cm^3)
% ch,cc: velocidad del sonido en el medio extrerior y en los cilindros
(m/s)
% N+1: numero de ondas planas en cada direccion -> numero total de
ondas planas=(N+1)^2
% ADVERTENCIA: usar N par!
% M: numero de (k,w) que calcula para cada camino -> numero total de
(k,w)=3M+1

% Define parametros

r0=d/2/a0;
bh=ch^2*rhoh;
bc=cc^2*rhoc;

% Calcula factor de llenado red cuadrada

f=pi*r0^2;

% Inicializa variables

k=zeros(2,3*M+1);
w=zeros((N+1)*(N+1),3*M+1);

% Calcula vectores de la red reciproca (cuadrada)

j=1;
for n=0:1:N;

for m=0:1:N;

g(j,:)=2*pi*[n-N/2,m-N/2];

j=j+1;

end;
end;

% Calcula matriz |G-G'|^2

G= repmat(diag(g*g'),1,(N+1)^2);
```

```

GG=-2*g*g'+G+G';
GG=GG+eye((N+1)^2,(N+1)^2); % suma 1 en la diagonal para evitar la
division por cero

% Calcula la matriz de b^(-1)

b=(bh/bc-1)*2*f*bessel(1,sqrt(GG)*r0)./(sqrt(GG)*r0);
b=b-diag(diag(b))+diag((bh/bc*f+(1-f))*ones(1,(N+1)^2)); % corrige los
valores de la diagonal (G-G'=0)

% Calcula bandas direccion Gamma-X

for j=1:M;

j
k(:,j)=[(j-1)*pi/M,0]
ks(j)=pi*(j-1)/M;

a=matrix_calc(k(:,j)',g,GG,N,f,r0,rhoh,rhoc);

w2=sort(eig(a,b));

w(:,j)=sqrt(w2)/(2*pi)*ch/(a0*.01)/1000; % sqrt(w2): frecuencia angular
adimensional; w: frecuencia dimensional (en kHz)

end;

% Calcula bandas direccion X-M

for j=1:M;
j
k(:,j+M)=[pi,(j-1)*pi/M]
ks(j+M)=pi+pi*(j-1)/M;

a=matrix_calc(k(:,j+M)',g,GG,N,f,r0,rhoh,rhoc);

w2=sort(eig(a,b));

w(:,j+M)=sqrt(w2)/(2*pi)*ch/(a0*.01)/1000; % sqrt(w2): frecuencia
angular adimensional; w: frecuencia dimensional (en kHz)

end;

% Calcula bandas direccion M-Gamma

for j=M:-1:0;

j
k(:,3*M-j+1)=[j*pi/M,j*pi/M]
ks(3*M-j+1)=pi+pi+sqrt(2)*pi*(M-j)/M;

a=matrix_calc(k(:,3*M-j+1)',g,GG,N,f,r0,rhoh,rhoc);

```

```

w2=sort(eig(a,b));

w(:,M-j+1+2*M)=sqrt(w2)/(2*pi)*ch/(a0*.01)/1000; % sqrt(w2): frecuencia
angular adimensional; w: frecuencia dimensional (en kHz)

end;

% Calcula matriz de rho^(-1)*(k+G)*(k+G')

function a=matrix_calc(k,g,GG,N,f,r0,rhoh,rhoc)

Gk=repmat(g*k',1,(N+1)^2);
Gk2=(k*k')*ones((N+1)^2,(N+1)^2)+Gk+Gk'+g*g';

a=(rhoh/rhoc-1)*2*f*bessel(1,sqrt(GG)*r0)./(sqrt(GG)*r0).*Gk2;
a=a-diag(diag(a))+diag((rhoh/rhoc*f+(1-f))*ones(1,(N+1)^2)).*Gk2; %
corrige los valores de la diagonal

```