

La clase es de programación, prendan su máquina e inicien una sesión de MatLab.

Introducción. Ejemplo del uso de MatLab

Objetivo. Ilustrar con un ejemplo real, algunas de las capacidades del uso de MatLab para resolver problemas reales.

Presentación breve del Simulador del nuevo Proceso de impartición de justicia, para la determinación optima de los requerimientos de Jueces y salas

Función seno.

Objetivo. Manipular gráficamente los parámetros de la función seno, y relacionarla con su uso para representar, señales eléctricas, vibraciones, sonidos, luz, radiación electromagnética.

```
%% INICIA PROGRAMA DE CLASE
close all; % borra todas las graficas
clear all; % borra todas las variables

%%-----
% queremos graficar y analizar los parámetros de la funcion seno
%%-----

Tiempo = 1.0; % variables de tiempo
Puntos = 2000; % numero de puntos
dt = 1/Puntos; % cuanto tiempo le toca a cada punto

% variables de la funcion seno
Amplitud = 1;
Frec = 10; % frecuencia ciclos por segundo
Fase = 0; % partes de pi, ejemplo pi/10

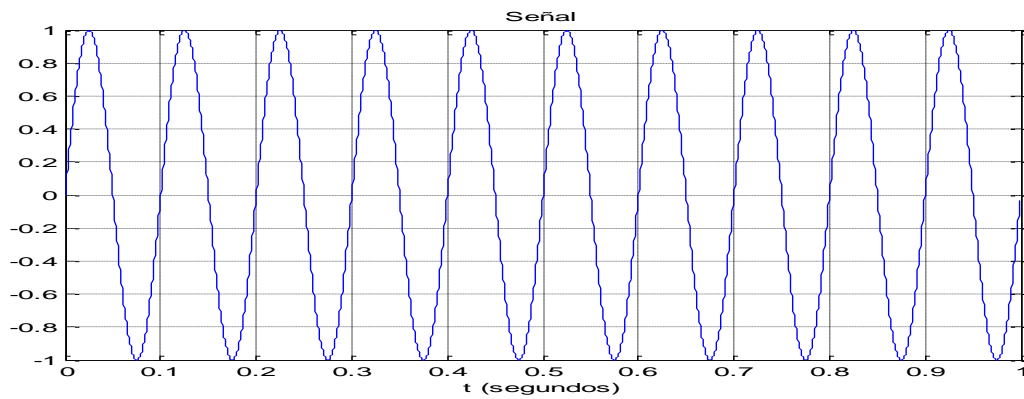
% La función senoidal, señal senoidal, onda senoidal
t = (0:dt:Tiempo-dt)'; % la absisa
y = Amplitud* sin(2*pi*Frec*t + Fase); % la ordenada

figure(1); % En figura 1 casi siempre
plot(t,y); % Graficamos x contra y
xlabel ('t (segundos)');
title ('Señal');
grid on;
```

Analizando el código vemos:

- Al abrir MatLab trabajamos en una sesión, corriendo scripts, todo dentro de una sesión, conviene iniciar limpiando variables.
- Hacer comentarios siempre, siempre.
- Agrupar secciones de código,
- Las variables que contienen los parámetros de la función. Amplitud, frecuencia y fase

- Las variables que representa la discretización de la función, número de puntos (frecuencia de muestreo), tiempo o duración representada, e incremento de tiempo entre punto y punto.
- Variar los parámetros, correr y relacionarlos con graficas de la función obtenidas.

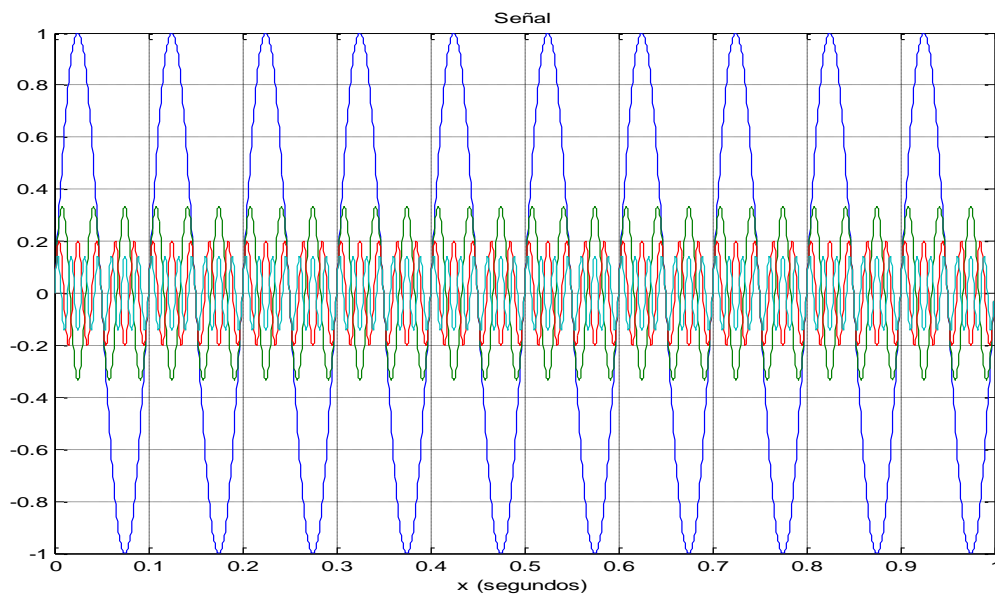


Sumas de senos.

Objetivo. Construir otras funciones a partir de sumas de funciones senoidales.

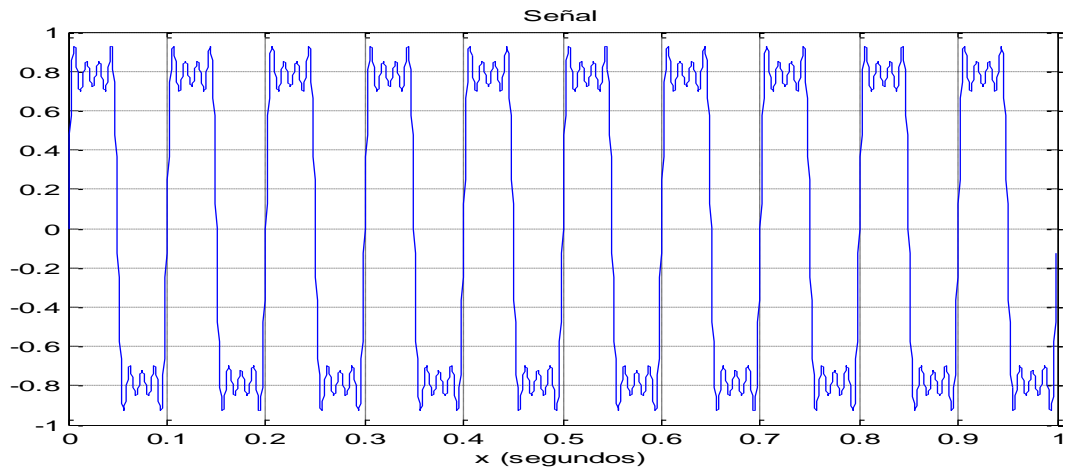
Expanda el escript anterior para generar y graficar, funciones senoidales de acuerdo a la siguiente tabla, y gráfíquelas.

Función	Amplitud	Frecuencia	Fase
y	1	10	0
y1	1/3	30	0
y2	1/5	50	0
y3	1/7	70	0
y4	1/9	90	0



Grafique $y_{\text{Suma}} = y_1 + y_2 + y_3 + y_4$

La suma se está pareciendo a una señal cuadrada, si seguimos sumando funciones con la secuencia de amplitud y frecuencia indicadas, el resultado se va pareciendo más a una señal cuadrada de la misma frecuencia y amplitud que y1.



Señal Cuadrada.

Objetivo. Descomponer una función (por ejemplo cuadrada) en la suma de funciones seno correspondiente (el ejemplo anterior).

```
%% INICIA PROGRAMA DE CLASE
```

```
close all; % borra todas las graficas
clear all; % borra todas las variables
```

```
%%-----
% queremos graficar la funcion cuadrada
%%-----
```

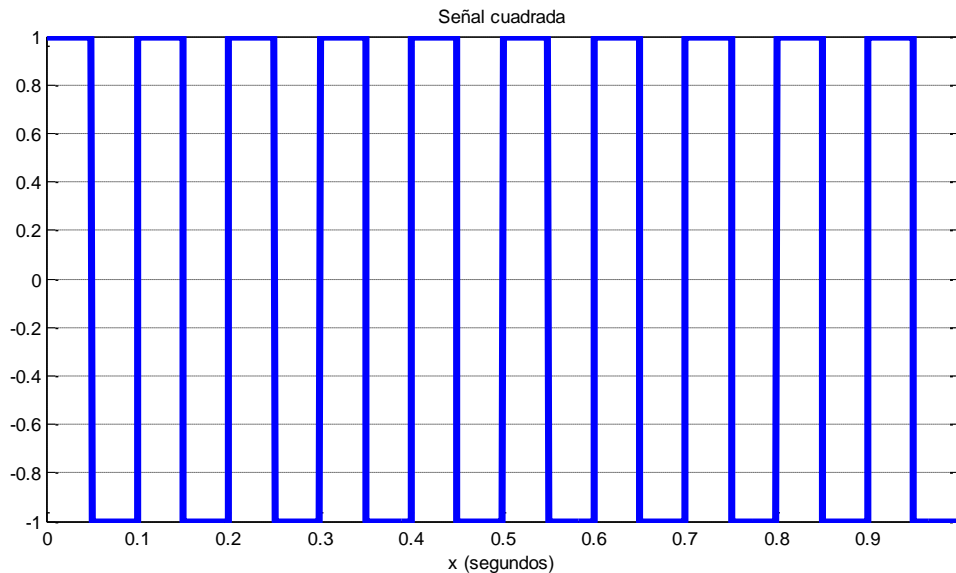
```
Tiempo = 1.0; % variables de tiempo
Puntos = 2000; % numero de puntos
dt = 1/Puntos; % cuanto tiempo le toca a cada punto
```

```
% variables de la funcion cuadrada
Amplitud = [1 1/3 1/5];
Frec = [10 30 50]; % frecuencia ciclos por segundo
Fase = [0,0,0]; % partes de pi, ejemplo pi/10
```

```
t = (0:dt:Tiempo-dt)'; % la abscisa
n = 1;
sq = Amplitud(n)*square(2*pi*Frec(n)*t+Fase(n));
```

```
figure(1); % En figura 1 casi siempre
plot(t,sq); % Graficamos x contra y
xlabel('x (segundos)');
title('Señal cuadrada');
grid on;
```

El código básicamente es el mismo que el script para la función seno del primer ejercicio. Solo cambia la llamada a la función de Matlab square en lugar de la función sin.



Y si hubiera una manera de analizar esta señal cuadrada y obtener las funciones seno que la construyeron. Como en el ejercicio de la suma de senos.

Si la hay.

Agreguemos el siguiente código al script y analicémoslo.

```
y=fft(sq);           % a la señal cuadrada sq apliquémosle la función de matlab fft
g=fftshift(y);      % al resultado y apliquémosle fftshift
power=abs(g).^2;    % al resultado le sacamos el valor absoluto lo elevamos al cuadrado.
```

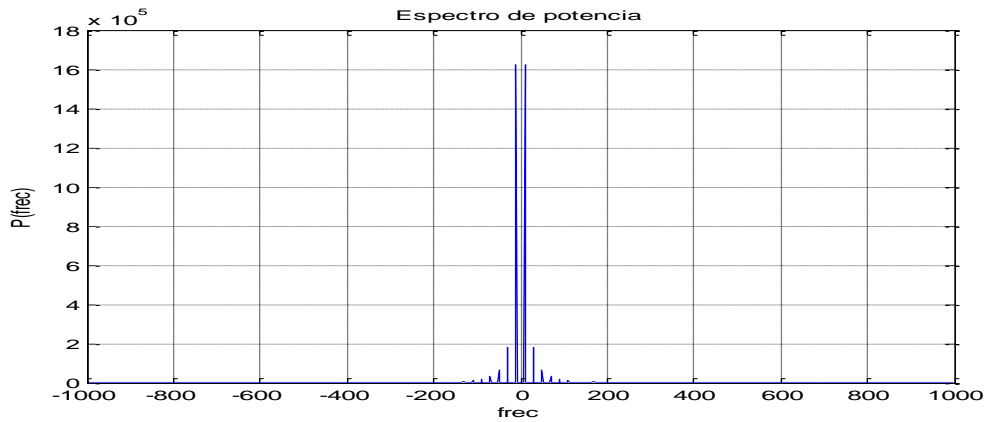
```
figure(2);          % abrir nueva figura
df=1/(Puntos*dt);  % el vector para el eje x
f=(-Puntos/2:Puntos/2-1)*df;
                    % graficamos
```

```
plot(f,power)
xlabel('frec')
ylabel('P(frec)')
title('Espectro de potencia')
grid on;
```

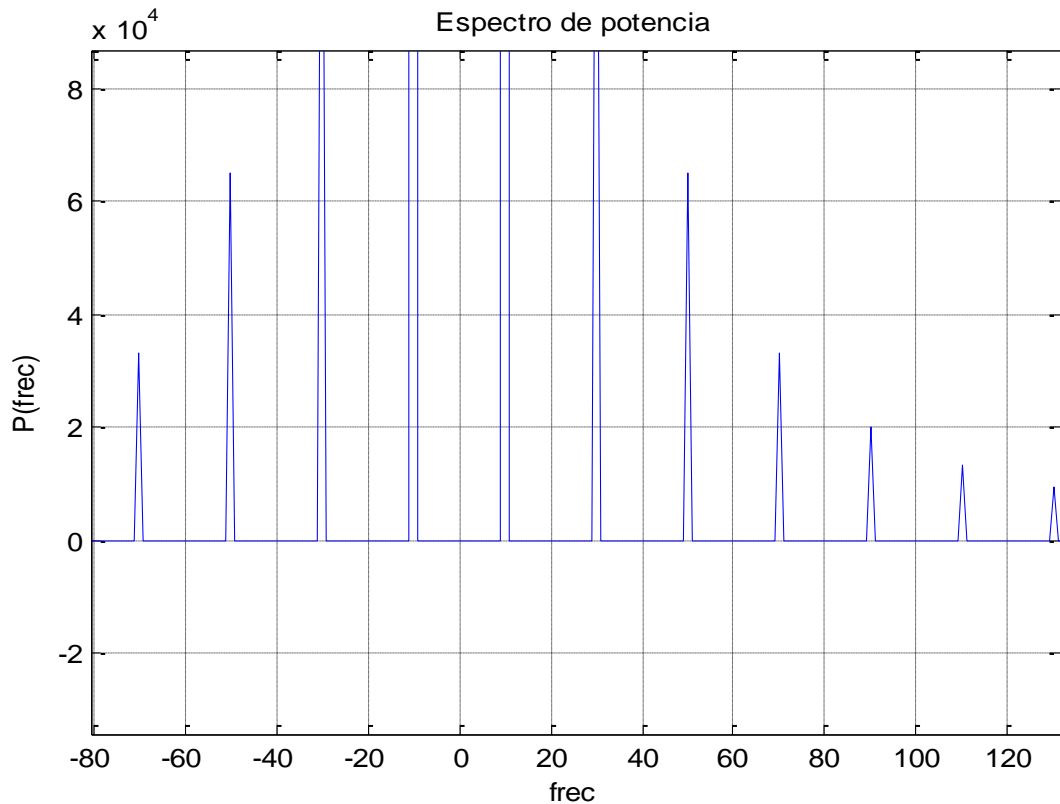
La función de MatLab fft (Transformada rápida de Fourier) es un algoritmo numérico para calcular la Transformada de Fourier, que toma una señal o función y obtiene los parámetros de las funciones seno que la conforman ¡Genial!

La función fftshift se utiliza, por la implementación numérica del método y lo que hace es poner el cero en el centro, desplaza los valores la mitad.

La función abs y elevar al cuadrado, se utiliza porque el resultado es complejo y se quiere poner solo magnitud del valor complejo.



En el eje x tenemos la frecuencia (no tiempo, tenemos frecuencia, ese es el truco) se observan piquitos, hagamos un acercamiento zoom al centro.



Para las frecuencias positivas, hay un piquito en la función en 10, en 30, en 50 en 70, 90, 110, etc etc. (la parte negativa no la discutiremos en este momento) justo en las frecuencias que sumamos de senos, y el tamaño del piquito también corresponde a la amplitud relativa de 1, 1/3, 1/5, etc, etc.

Conclusiones.

- Al obtener la fft de una señal o función, nos regresa la frecuencia de las funciones senos que la componen. ¡Ultra genial!

- Toda señal periódica se puede descomponer en suma de señales senoidales de magnitud y frecuencia adecuadas.
- Fourier nos dice como a partir de una señal o función (en este caso señal cuadrada), se pueden obtener los valores de frecuencia y magnitud de cada uno de las funciones seno que la componen.

Wikipedia

Jean-Baptiste Joseph Fourier (Auxerre, Francia, 21 de marzo de 1768 - París, 16 de mayo de 1830), matemático y físico francés conocido por sus trabajos sobre la descomposición de funciones periódicas en series trigonométricas convergentes llamadas Series de Fourier,

Sonidos.

Objetivo. Identificar que los sonidos, se pueden descomponer como sumas de senos y analizar el espectro de frecuencias.

En otro script ingresemos y analicemos el código siguiente, lo nuevo es:

- La línea `[pita,fs]=wavread('sinus_1000Hz_40dBSPL.wav');` que lo que hace es abrir el archivo `'sinus_1000Hz_40dBSPL.wav'` (archivo de sonido que puede ser tocado por el player de Windows) y nos regresa eso sonido como un vector de datos en la variable `pita`, y la frecuencia de muestreo en `fs`.
- La línea `soundsc(pita,fs)` que reproduce el sonido que está en la variable `pita`, con frecuencia de muestreo `fs`

%% INICIA PROGRAMA DE CLASE

```
close all; % borra todas las graficas
clear all; % borra todas las variables
```

```
[pita,fs]=wavread('sinus_1000Hz_40dBSPL.wav');
%[pita,fs]=wavread('kungfu.wav');
whos
```

```
Puntos = length(pita); %1000 %
time=(1/44100)*Puntos;
t=linspace(0,time,Puntos);
```

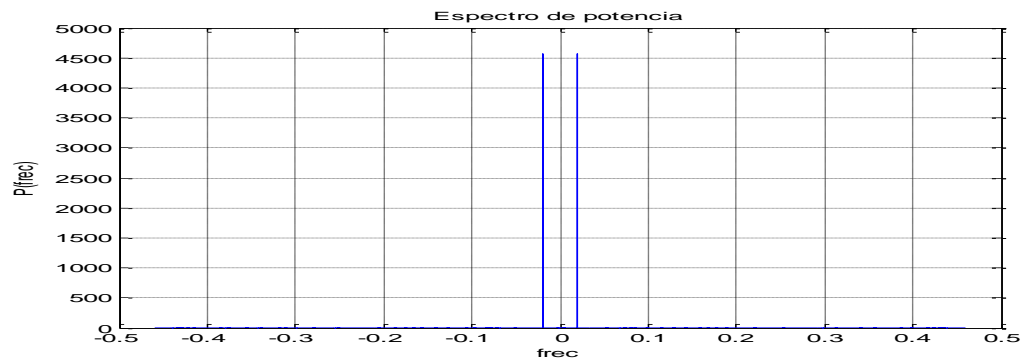
```
figure(1);
plot(t,pita(1:Puntos))
xlabel('tiempo (seg)');
ylabel('relative signal')
```

```
soundsc(pita,fs) % plays mono
```

```
dt = time;
y=fft(pita);
g=fftshift(y);
power=abs(g).^2;
df=1/(Puntos*dt);
```

```
f=(-Puntos/2:Puntos/2-1)*df;
whos
```

```
figure(2);  
plot(f,power)  
    xlabel('frec')  
    ylabel('P(frec)')  
    title('Espectro de potencia')  
    grid on;
```

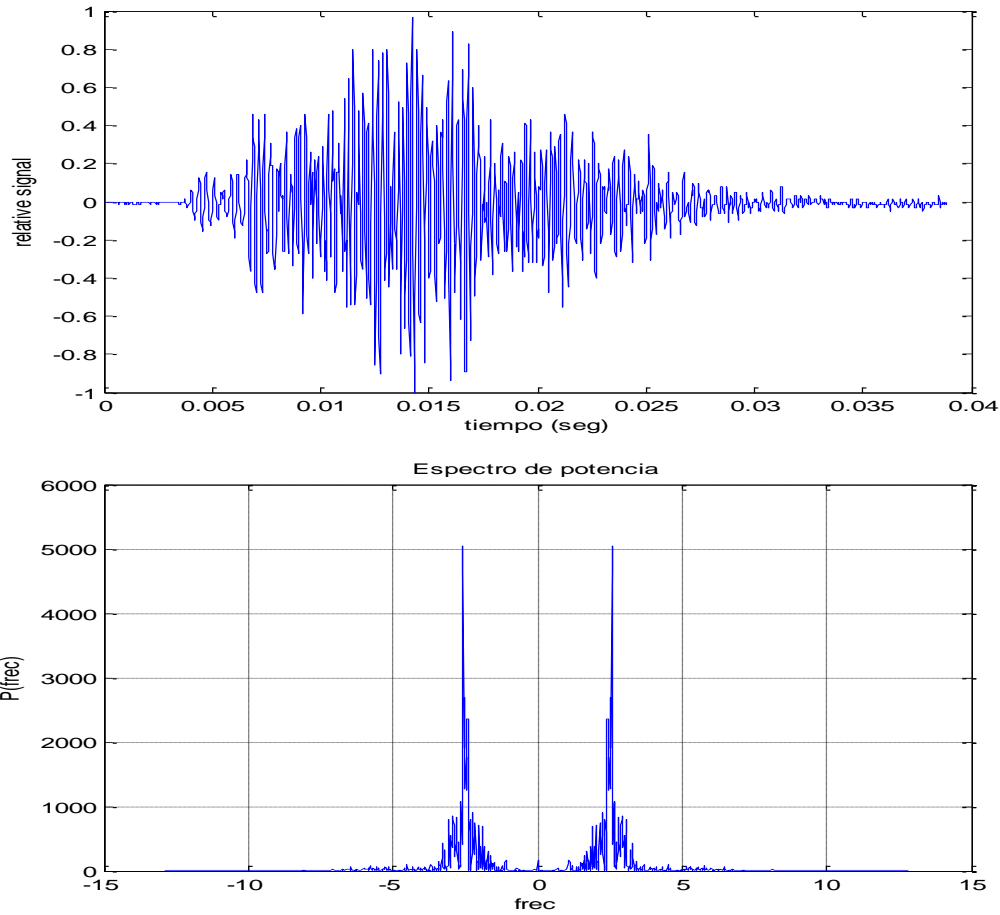


El espectro de frecuencia, en su parte positiva, solo tiene un pico, es una senoidal pura.

Cambiando de archivo

```
[pita,fs]=wavread('kungfu.wav');
```

Corremos, escuchamos, miramos la señal en el tiempo, y analizamos su contenido de frecuencias.



Sonido sintético.

Objetivo. Si los sonidos son vectores o variables, entonces se pueden crear sonidos y tocarse.

```

clc;
clear all;
f = 200;
fs= 20*f;
Ts = 1/fs;
Amp = 2;
t_total = 2;

t=[0:Ts:t_total];

s = Amp * sin(2*pi*t*f);
whos s

figure(1);
plot (s);
s_norm=s/max(abs(s));
whos s

figure(2);
plot(s_norm)

```



```
num_per = 3;
figure(3);
plot( t(1:(num_per*1/f)/Ts), s_norm(1:(num_per*1/f)/Ts ));
grid on;

%wavwrite(s_norm,fs,'pita1.wav');
sound(s_norm,fs);
```

Sonido 'Yo'

Objetivo. Capturar con el micrófono mis palabras, graficarlas en el tiempo, reproducirlas, aplicarles transformada de Fourier, graficar la transformada y discutir resultados.

```
Fs = 11025;
num_seg = 4.5;
```

```
y = wavrecord(num_seg*Fs,Fs);
```

```
figure(1);
plot(y)
```

```
sound(y,Fs);
```

Tarea.

- Investigar quien es Fourier y su transformada.
- Hacer un programa que toque de forma consecutiva, una octava, do, re, mi, fa, sol, la, si
- Hacer un programa que toque al mismo tiempo todas las notas de una octava.
- Obtener el espectro del inciso anterior, graficarlo y explicarlo.
- Hacer un programa que capture a ustedes cantando una nota.
- Hacer un programa que capture su voz diciendo la palabra 'yo'
- Obtener el espectro de los dos puntos anteriores.
- EXTRA no entregable para cuando no puedan dormir, hacer en MatLab teclado sintetizador.

Notas.

Utilización de versión MatLab R2009b y el toolbox de señales.