

# **Análisis Armónico en Matlab**

**Análisis Estadístico de Datos Climáticos 2013**

# Espectro

$$cov_{xx}(k) = \frac{1}{N-k-1} \sum_{i=1}^{N-k} (x_i - \bar{x})(x_{i+k} - \bar{x})$$

$$corr_{xx}(k) = \frac{cov_{xx}(k)}{cov_{xx}(0)} = \frac{cov_{xx}(k)}{\sigma_x^2}$$

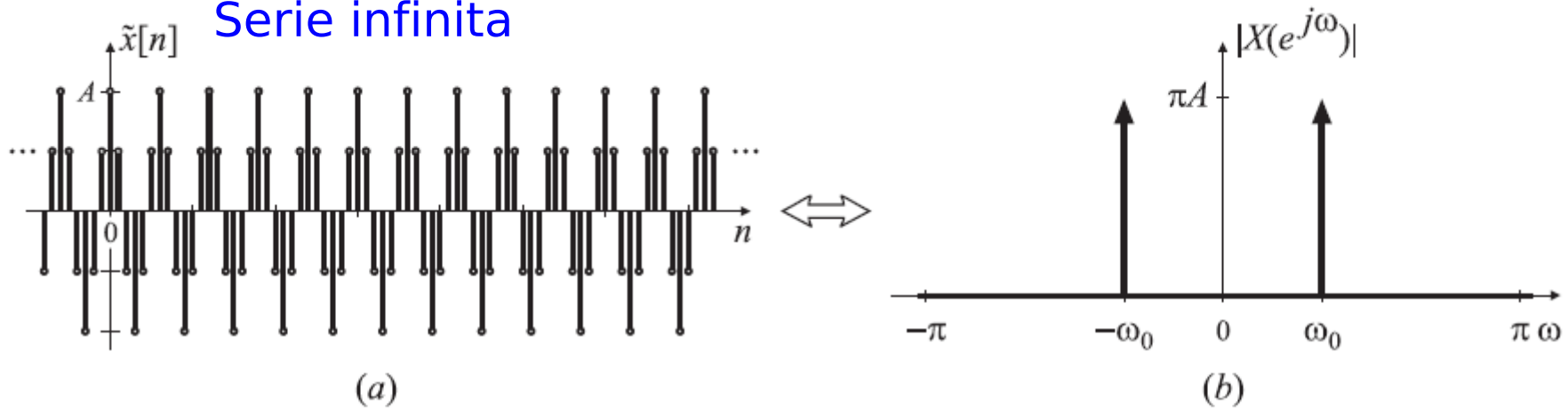
$$X_{xx}(f) = \sum_{k=0}^M corr_{xx}(k) e^{i2\pi fk/f_s}$$

$$X_{xx}(f) = \sum_{k=0}^M corr_{xx}(k) w(k) e^{i2\pi fk/f_s}$$

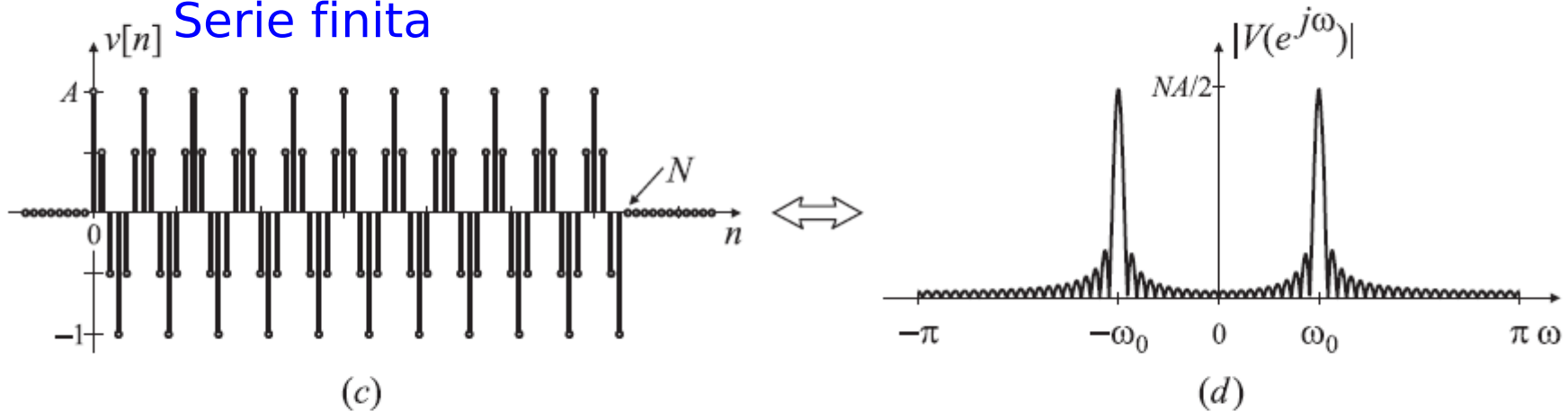
$$PSD = \frac{X_{xx}^*(f) X_{xx}(f)}{f_s} = \frac{|X_{xx}(f)|^2}{f_s}$$

# Fuga Espectral (“spectral leakage”)

Serie infinita



Serie finita



La truncación temporal transforma un espectro ideal formado por una serie de impulsos periódicos perfectamente localizados en frecuencia a una función continua y periódica que cubre todo el rango de frecuencias, con una serie de picos espurios denominados lóbulos laterales → fuga de energía

Ventana

Modulo de respuesta en frecuencia

Respuesta en dB

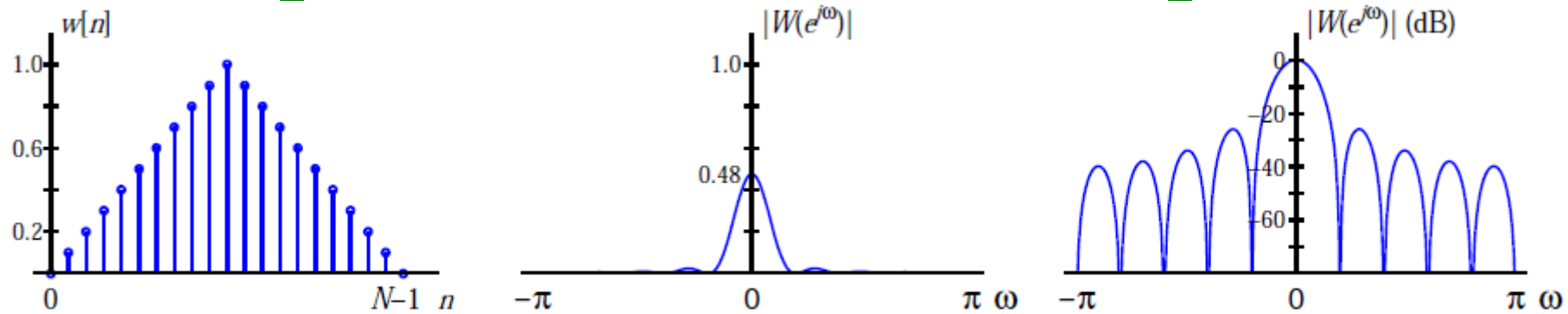
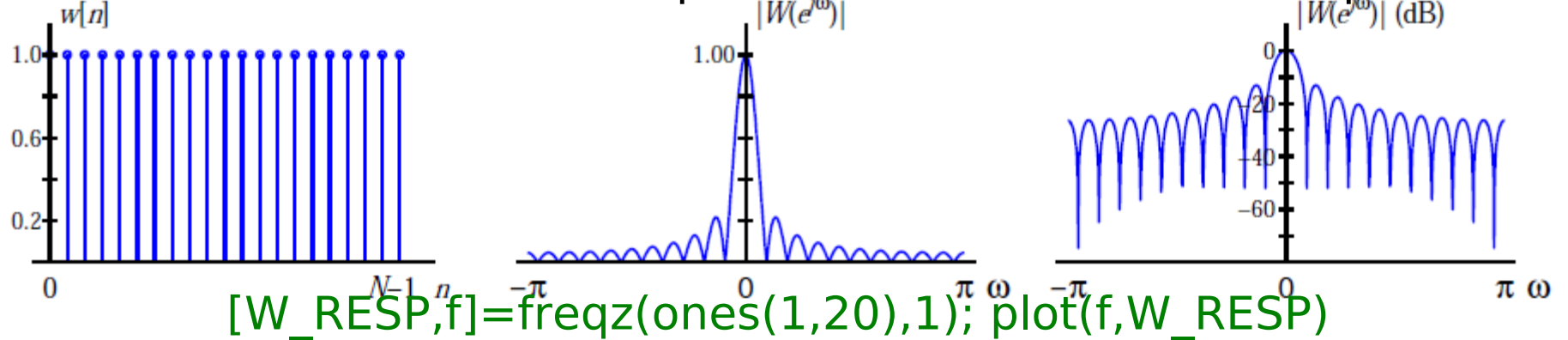


Fig. 11.27. Ventana triangular o de Bartlett.

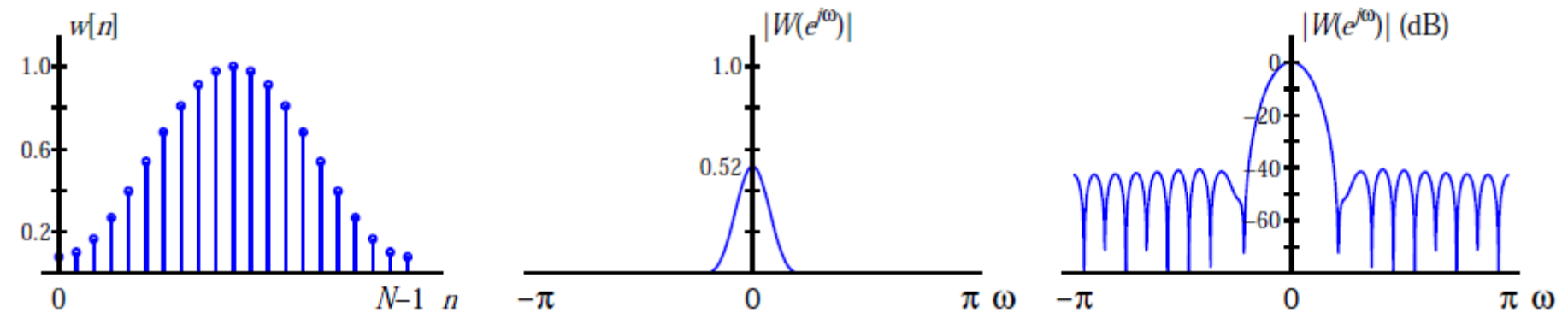
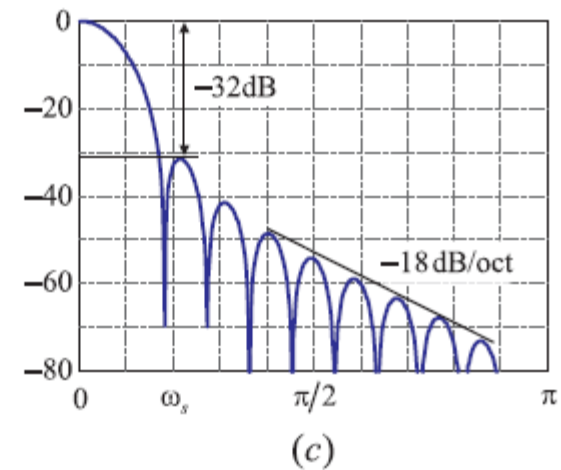
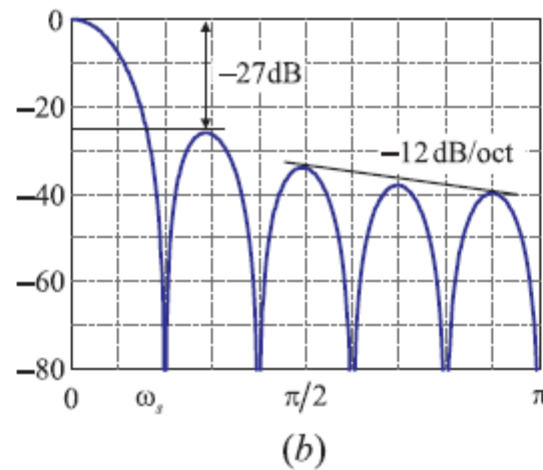
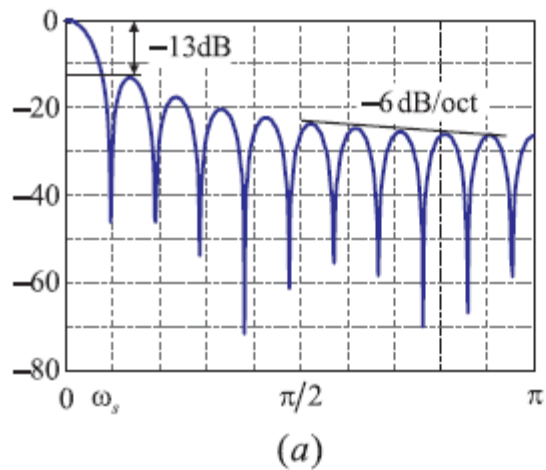


Fig. 11.29. Ventana de Hamming.



**Fig. 11.23.** Atenuación y pendiente de decrecimiento de los lóbulos laterales de distintas ventanas: (a) rectangular; (b) triangular; (c) von Hann.

Cuanto menor sea el máximo del primer lóbulo lateral y más rápido decaigan los lóbulos hay menos fuga espectral

## Espectro de potencia

Generamos 3 series temporales

```
t = 1 : 1000; t = t';  
x = 2*sin(2*pi*t/50) + sin(2*pi*t/15) + 0.5*sin(2*pi*t/5);  
  
%% Adicionamos ruido  
randn('seed',0)  
n = 5*randn(size(x),1);  
xn = x + n;  
  
%% Adicionamos una tendencia lineal  
xt = x + 0.005*t;
```

## Calculemos el periodograma computando la transformada de Fourier discreta

```
% FFT calcula la TFD para nfft frecuencias =  $2^n$  mas  
% cercana a la longitud de los datos. La senal pasa a tener  
% 24 ceros al final.
```

```
nfft=1024;
```

```
Xxx = fft(x,nfft);
```

```
%% Densidad espectral
```

```
Fs=1; % frecuencia de muestreo
```

```
Ls=1000; % longitud serie
```

```
Pxx2 = abs(Xxx).^2/(Fs*Ls);
```

```
%% Pxx es simetrico. Multiplic. x2 para mantener energia total
```

```
Pxx = [Pxx2(1); 2*Pxx2(2:512)];
```

```
%% Frecuencias  $f=k/T$ ;  $f_{Nyquist}=(N/2)/(N\Delta T)=Fs/2=1/2$ 
```

```
f = 0 : 1/(1024-1) : 1/2;
```

```
plot(f,Pxx), grid
```



El resultado es el mismo si hacemos

```
[Pxx,f] = periodogram(x,[],1024,1);
```

```
1arg=serie
```

```
2arg=ventana para no  
tener spectral leakage
```

Defecto: cuadrada

```
[Ej: parzenwin(length(x))
```

```
triang(length(x))
```

```
hann(length(x)]
```

```
3arg=nfft
```

```
4arg=Fs
```

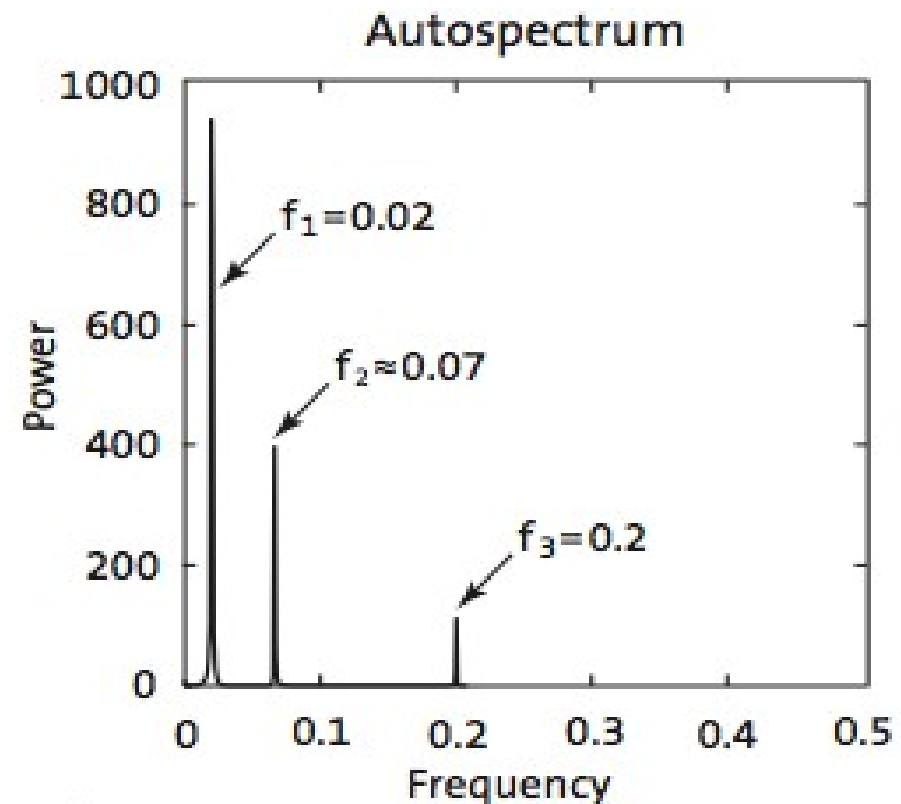
```
%%
```

```
plot(f,Pxx), grid
```

```
xlabel('Frequency')
```

```
ylabel('Power')
```

```
title('Auto-Spectrum')
```





Para la serie con ruido

%%

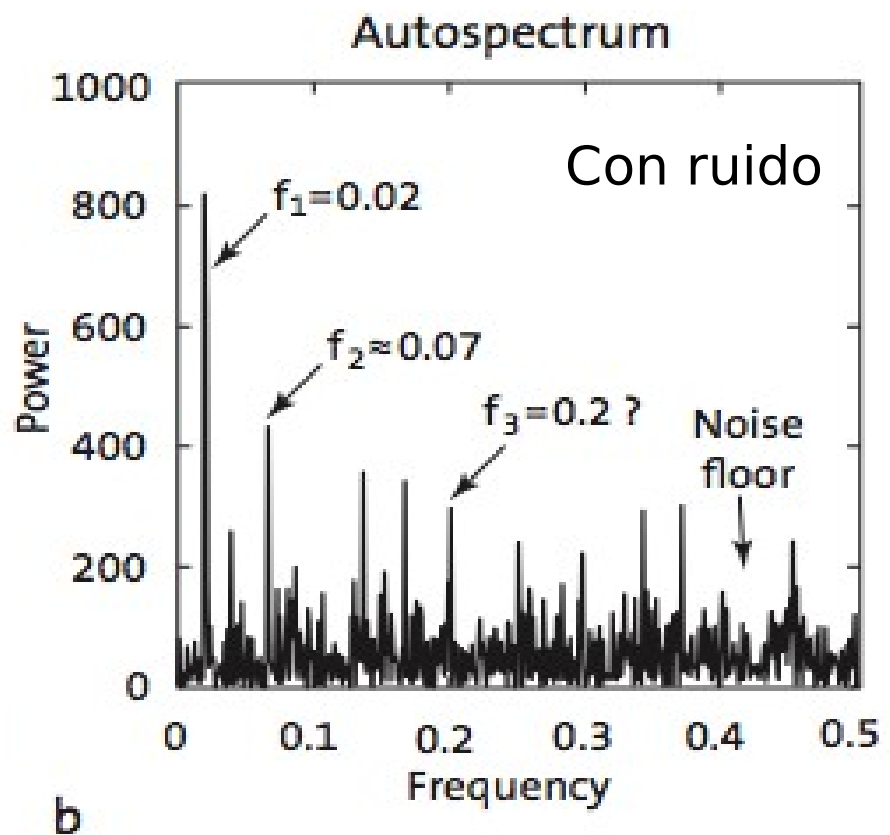
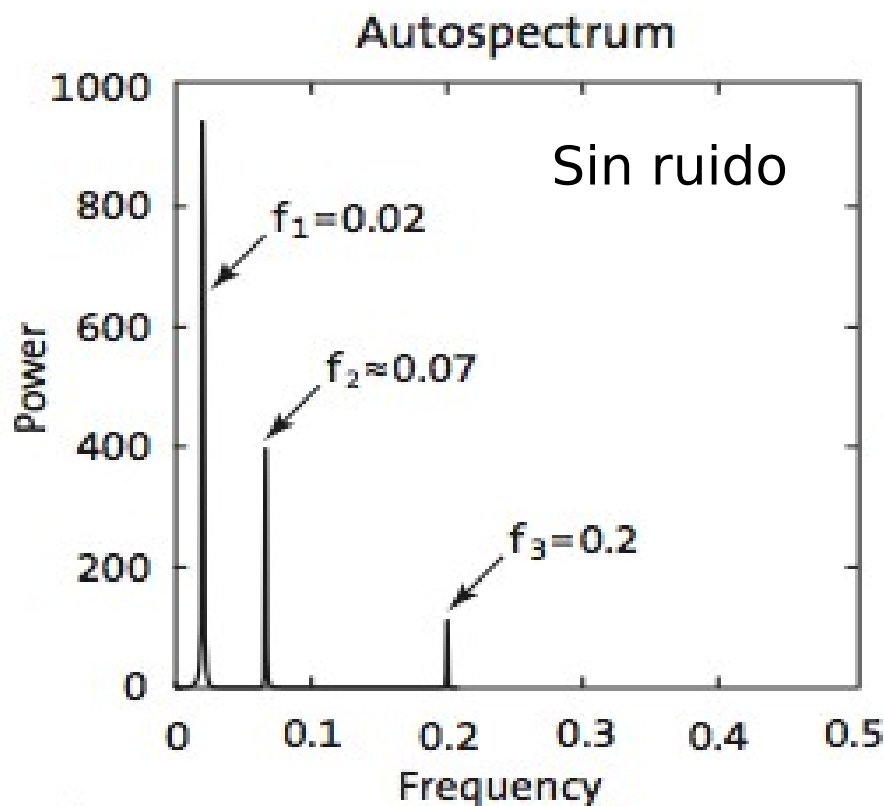
```
[Pxx,f] = periodogram(xn,[],1024,1);
```

```
plot(f,Pxx), grid
```

```
xlabel('Frequency')
```

```
ylabel('Power')
```

```
title('Auto-Spectrum')
```



## Metodo de Welch

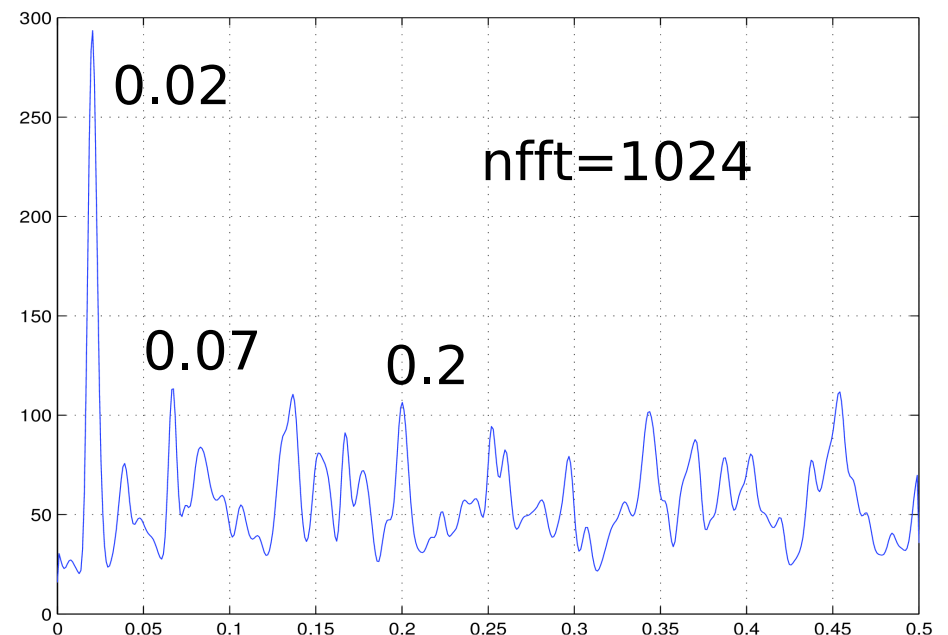
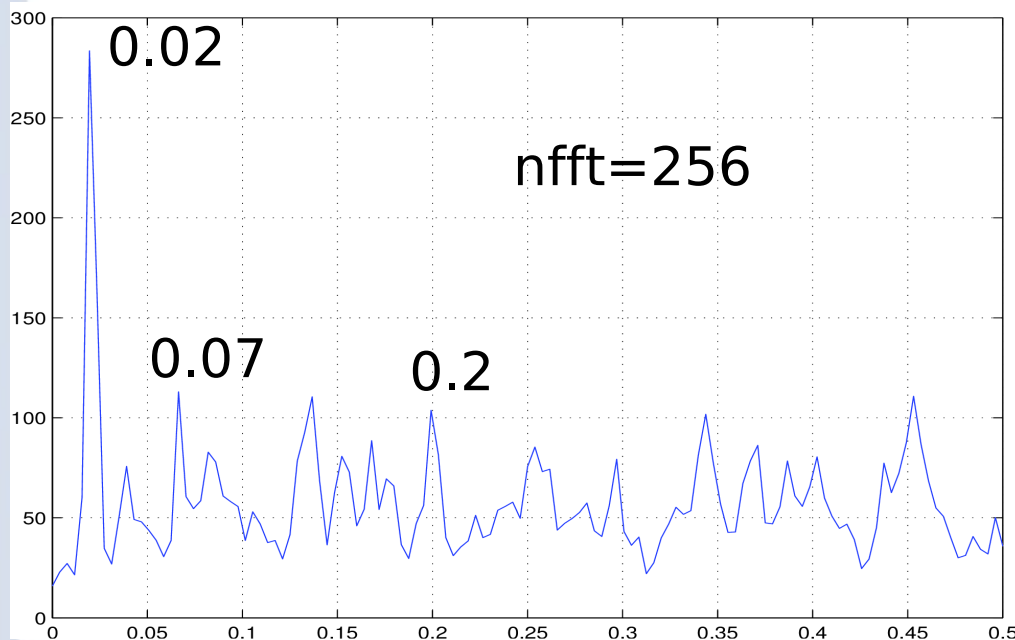
Tamaño segmento

```
[Pxx,f] = pwelch(xn,[],[],256,1);
```

```
plot(f,Pxx), grid
```

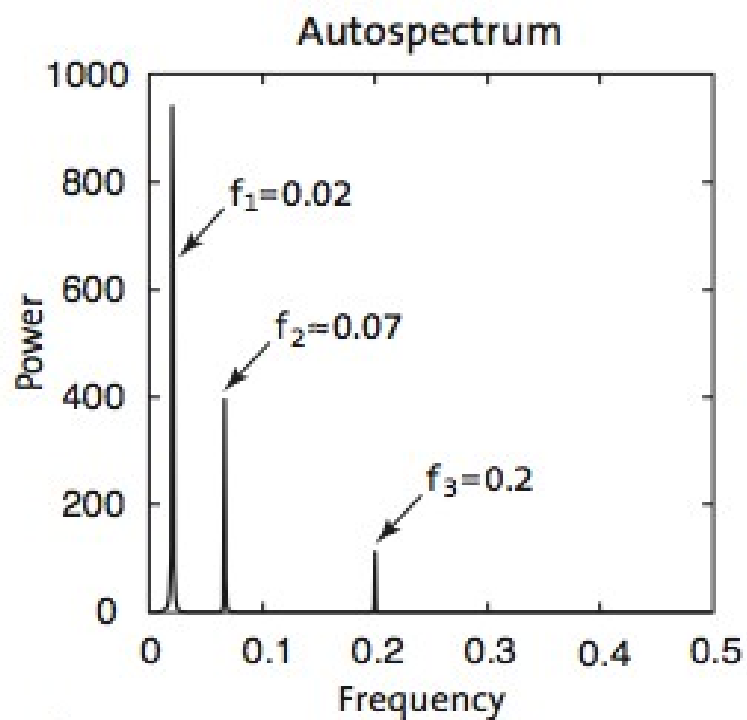
superposicion

% Por defecto usa: 8 segmentos con 50% de superposicion,  
% cada segmento es multiplicada por una ventana de  
% Hamming. El espectro resultante es el promedio de los  
% 8 espectros.

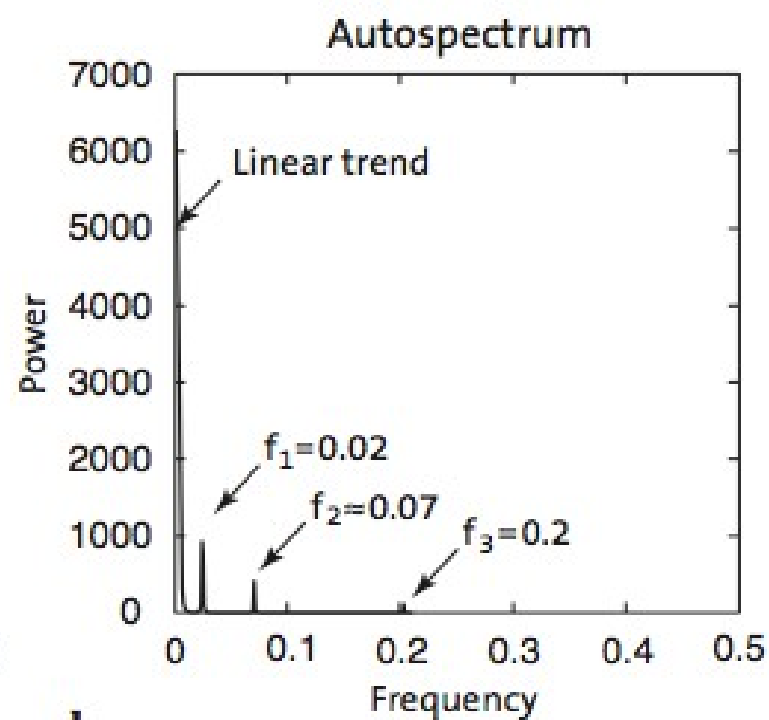


Para la serie con tendencia

```
[Pxxt,f] = periodogram(xt,[],1024,1);  
plot(f,Pxxt), grid  
xlabel('Frequency')  
ylabel('Power')
```



a



b

%Para quitar la tendencia  
`xdt = detrend(xt);`

## Espectro cruzado

$$cov_{xy}(k) = \frac{1}{N-k-1} \sum_{t=1}^{N-k} (x_t - \bar{x})(y_{t+k} - \bar{y})$$

$$corr_{xy}(k) = \frac{cov_{xy}(k)}{cov_{xy}(0)} = \frac{cov_{xy}(k)}{\sigma_x \sigma_y}$$

$$X_{xy}(f) = \sum_{k=0}^M corr_{xy}(k) e^{i2\pi fk/f_s}$$

Espectro  
cruzado

$$C_{xy} = \frac{|X_{xy}(f)|^2}{X_{xx}(f)X_{yy}(f)}$$

coherencia

```
%% Generamos series temporales periodicas de periodo T=5,  
desfasadas T/5
```

```
t = 1 : 1000;
```

```
x = 2*sin(2*pi*t/5);
```

```
y = 2*sin(2*pi*t/5 + 2*pi/5);
```

```
plot(t,x,'b-',t,y,'r-')
```

```
axis([0 50 -2 2]), grid
```

```
%%Espectro cruzado con metodo de Welch
```

```
[Pxy,f] = cpsd(x,y,[],0,1024,1);
```

Arg 1 y 2 - series temporales

Arg 3- ventana; Defecto la cuadrada.

Arg 4- no. de tiempos que se superponen en las ventanas

Arg 5- nfft

Arg 6- Fs

```
%%
```

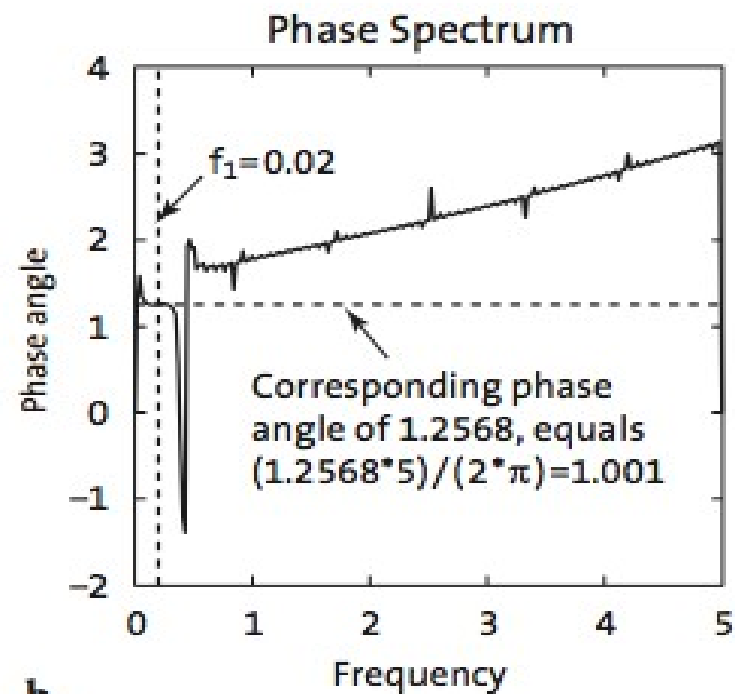
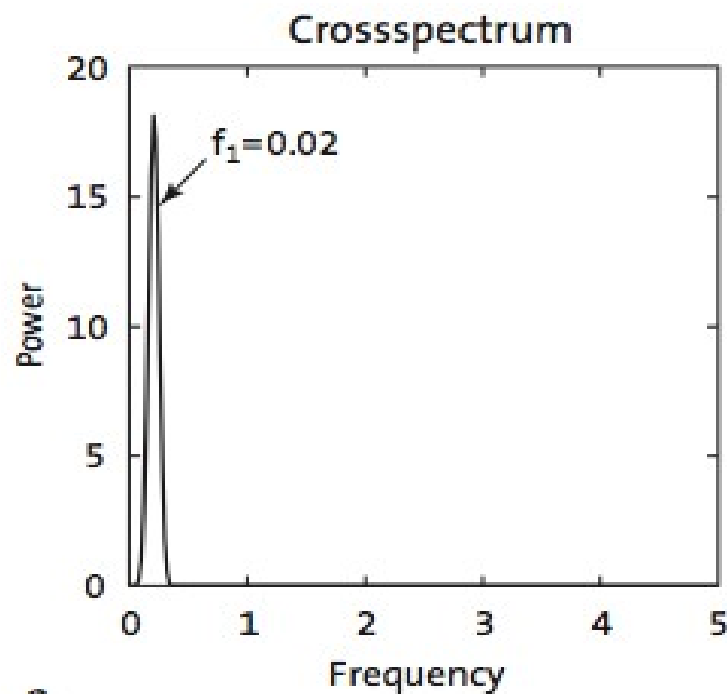
```
plot(f,abs(Pxy)), grid, xlabel('Frequency')
```

```
ylabel('Power'), title('Cross-Spectrum')
```

```
%%
```

```
phase = angle(Pxy);
```

```
plot(f,phase), grid  
xlabel('Frequency')  
ylabel('Phase Angle')  
title('Phase Spectrum')
```



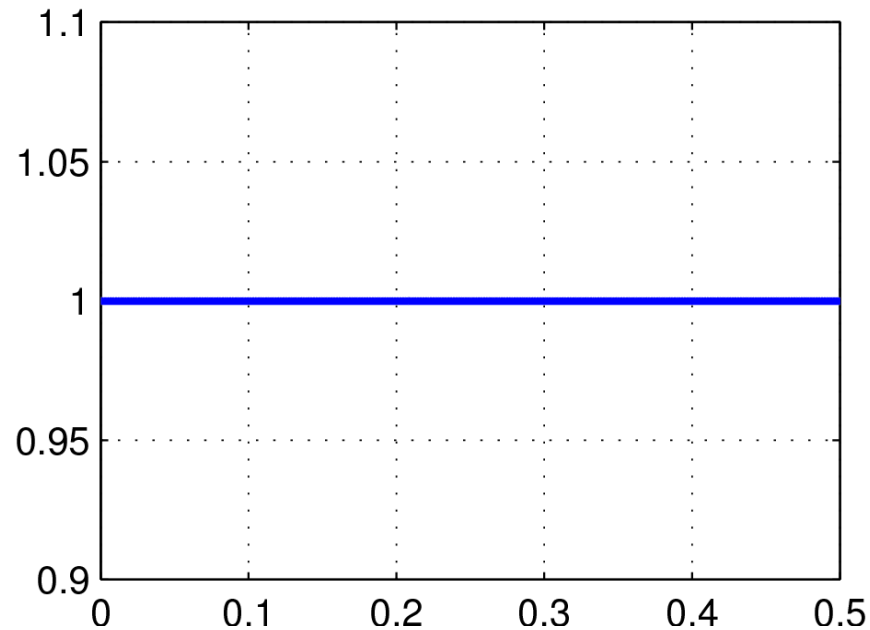
```
interp1(f,phase,0.2)
```

%% Calcula la coherencia. En este caso la coherencia es 1 para todas las frecuencias pues no hay ruido.

```
[Cxy,f] = mscohere(x,y,[],0,1024,1);
```

```
%%
```

```
plot(f,Cxy), grid
```





```
%Caso de dos series periodicas con diferentes periodos  
% y un periodo comun desfasadas
```

```
clear
```

```
t = 1 : 1000;
```

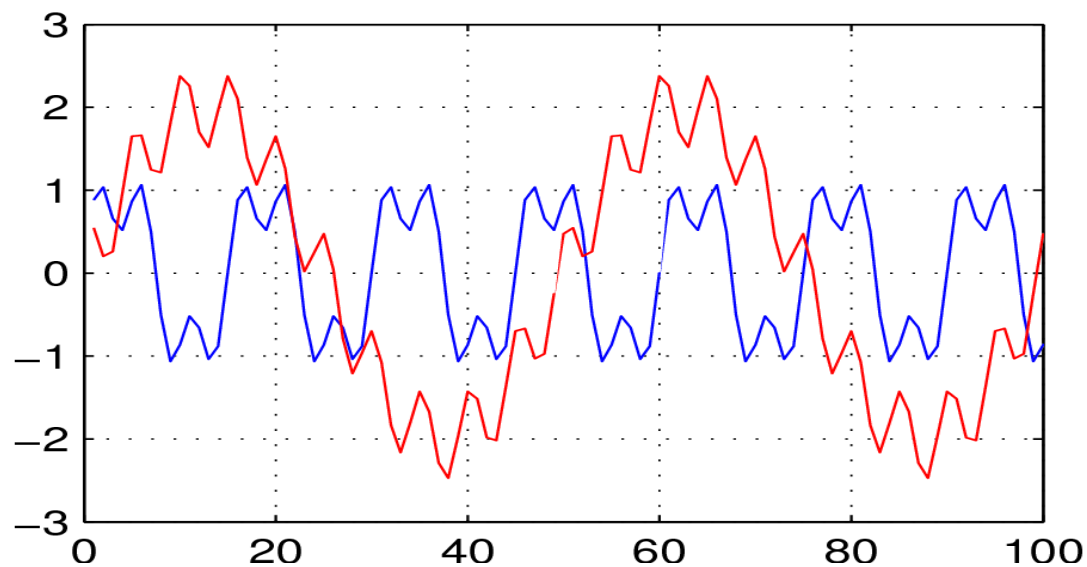
```
x = sin(2*pi*t/15) + 0.5*sin(2*pi*t/5);
```

```
y = 2*sin(2*pi*t/50) + 0.5*sin(2*pi*t/5+2*pi/5);
```

```
%%
```

```
plot(t,x,'b-',t,y,'r-')
```

```
axis([0 100 -3 3]), grid
```



```
[Pxy,f] = cpsd(x,y,[],0,1024,1);
```

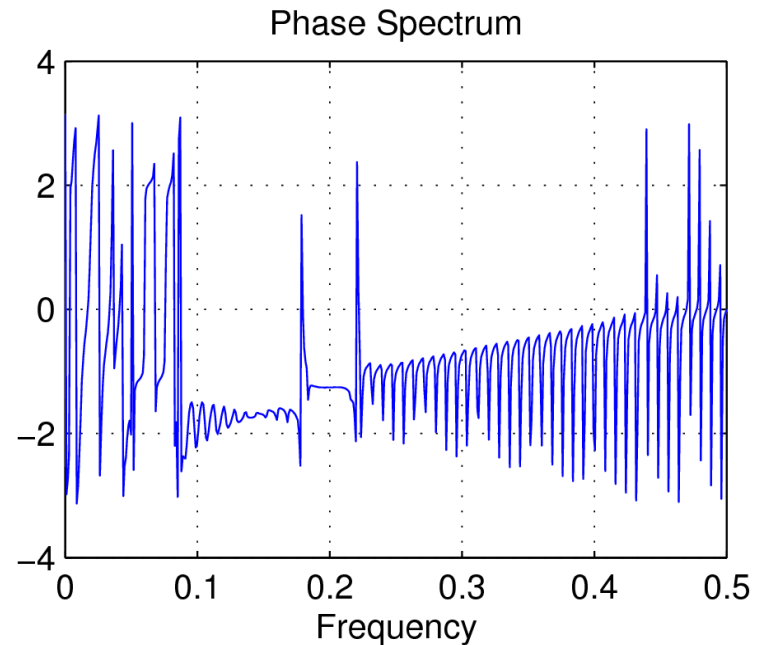
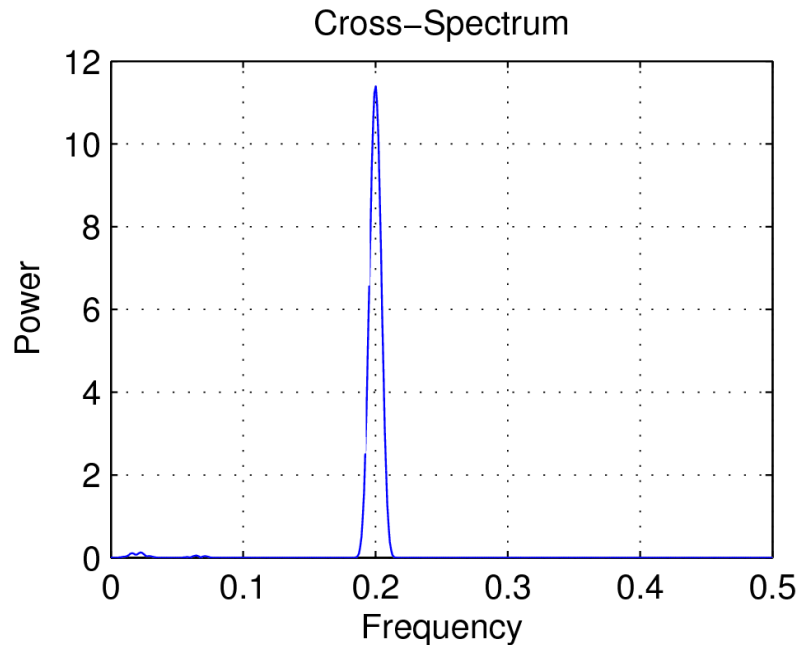
```
%%
```

```
plot(f, abs(Pxy)), grid, xlabel('Frequency')  
ylabel('Power'), title('Cross-Spectrum')
```

```
phase = angle(Pxy);
```

```
plot(f,phase), grid
```

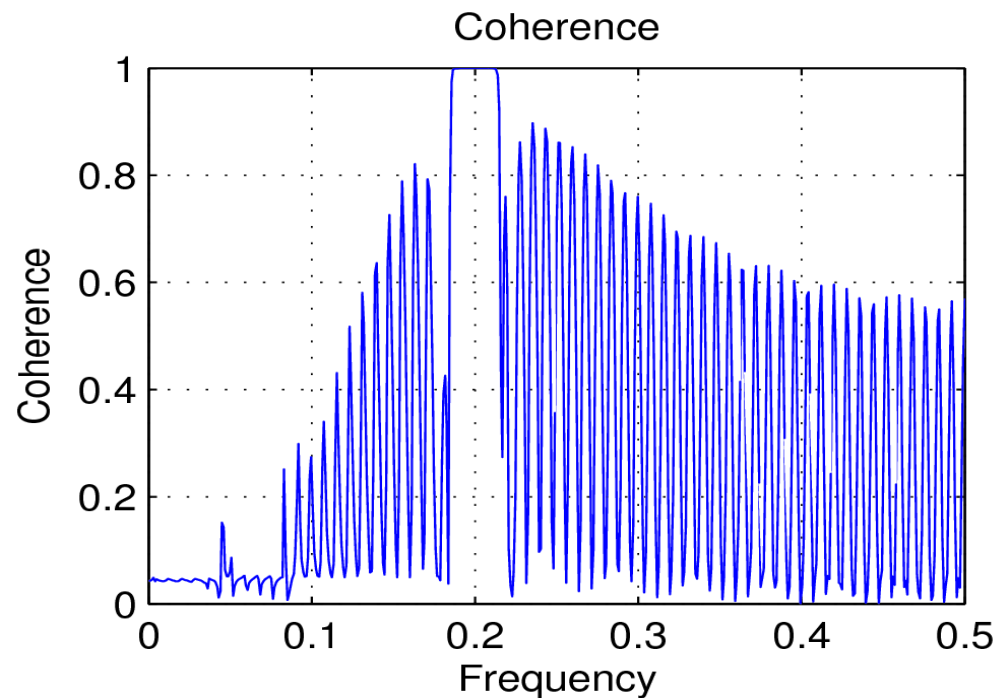
```
interp1(f,phase,0.2) %esto da -1.2572
```



```
[Cxy,f] = mscohere(x,y,[],0,1024,1);
```

```
%%
```

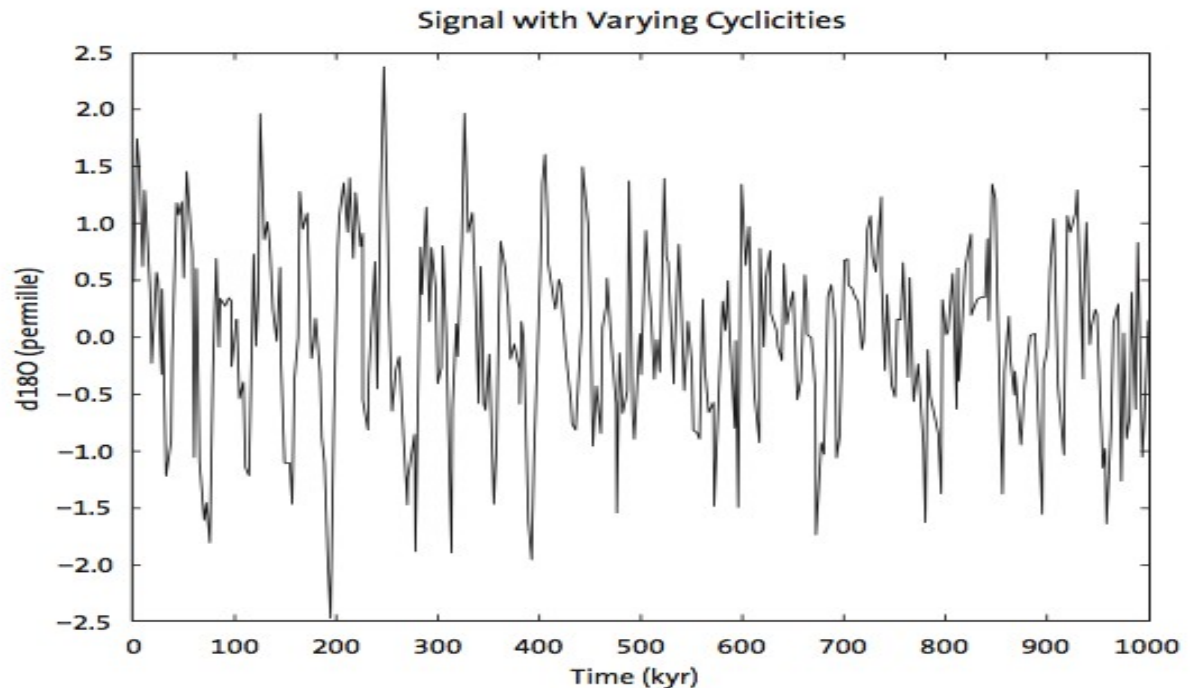
```
plot(f,Cxy), grid  
xlabel('Frequency')  
ylabel('Coherence')  
title('Coherence')
```



# Espectro de potencia evolutivo

- Calcula el espectro de potencia para ventanas superpuestas de la serie temporal.
- Estas ventanas son menores que las ventanas usadas por el método de Welch para mejorar la relación señal/ruido del espectro.

**Ejemplo:** se tiene una serie larga con periodicidades que cambian en el tiempo



```
[Pxx,f]=periodogram(serie,[],1024,1/3)
```

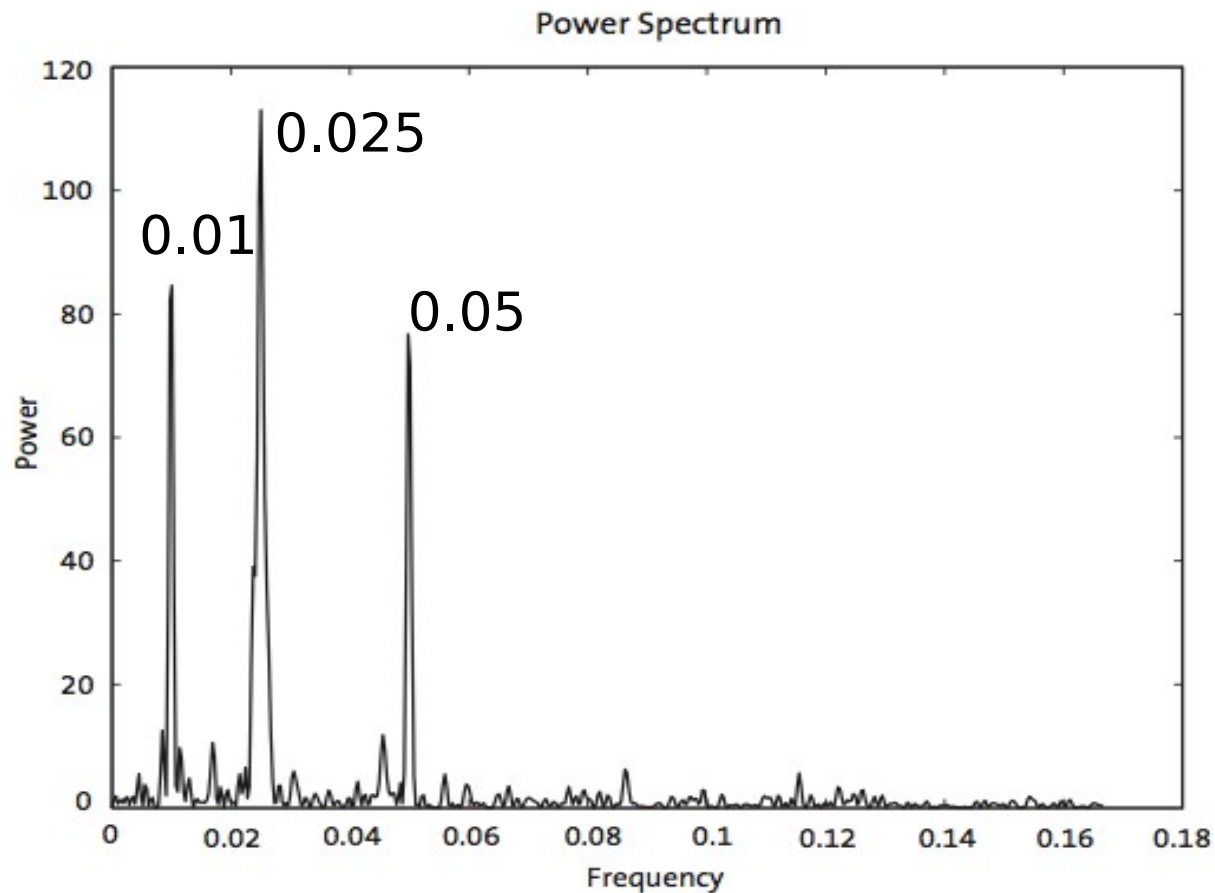
```
plot(f,Pxx)
```

```
xlabel('Frequency')
```

```
ylabel('Power')
```

```
title('Power Spectrum')
```

$F_s = 1/3 \text{ kyr}^{-1}$



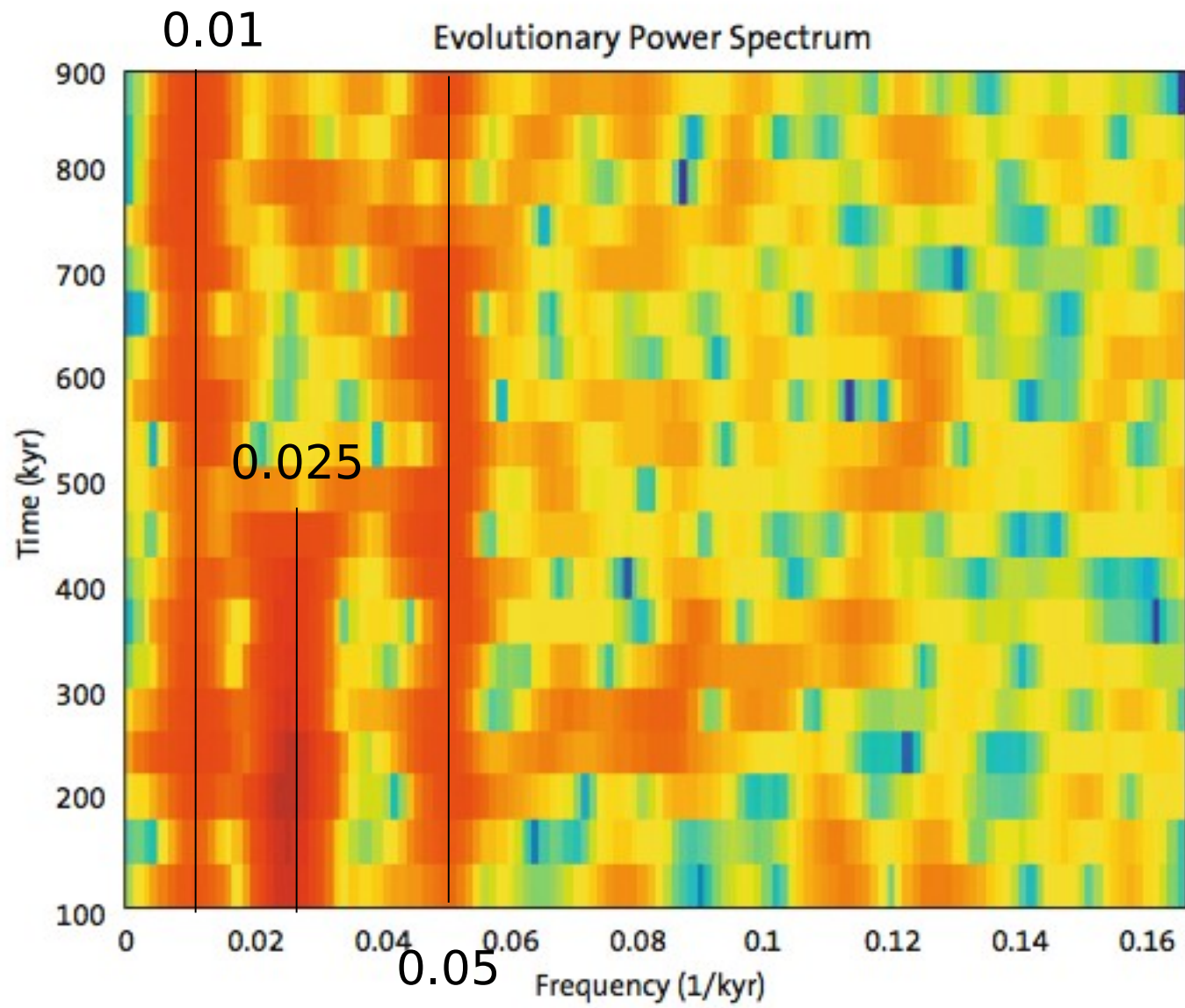
Función spectrogram

```
spectrogram(serie,64,50,256,1/3)
```

Por defecto, la serie temporal es dividida en 8 segmentos con un 50% de superposición. En cada segmento se usa una ventana de Hamming para evitar el “spectral leakage”.

En el ejemplo se usa una ventana de 64 puntos con una superposición de 50 puntos. La transformada de Fourier se calcula con  $nfft=256$ .







# Wavelets

A diferencia de la transformada de Fourier la transformada de “ondeletas” (wavelets) usa funciones de base que ya tienden a cero al inicio y final.

Las wavelets son pequeños paquetes de ondas definidos por una frecuencia y un decaimiento.

Las wavelets pueden estirarse y trasladarse en el tiempo y en frecuencia, lo cual permite mapear cambios en el dominio tiempo-frecuencia.

Matemáticamente la transformada de wavelet descompone una señal  $y(t)$  en funciones elementales  $\Psi_{a,b}$  derivadas de la wavelet madre por traslación y estiramiento

$$\Psi_{a,b} = \frac{1}{(a)^{1/2}} \Psi \left( \frac{t-b}{a} \right)$$

donde  $b$  marca la traslación y  $a$  ( $>0$ ) la escala de la wavelet. La transformada de wavelet de una función  $y(t)$   
Es

$$W(b,a) = \frac{1}{(a)^{1/2}} \int \Psi^* \left( \frac{t-b}{a} \right) y(t) dt$$

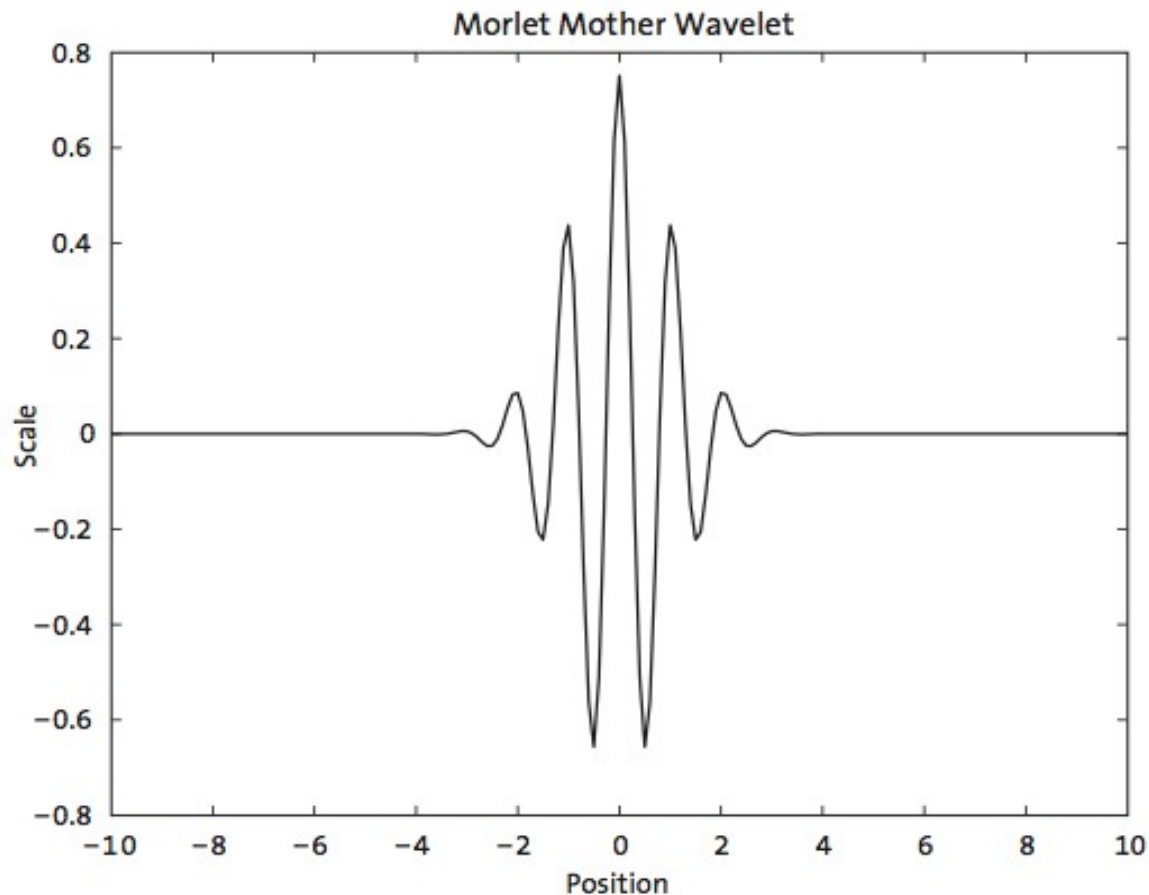
donde  $\Psi^*$  es el complejo conjugado

La wavelet madre mas usada es la de Morlet

$$\psi_0(\eta) = \pi^{-1/4} \exp(i\omega_0\eta) \exp(-\eta^2 / 2)$$

$\eta$  - tiempo

$w_0$  - no. de oscilaciones en la wavelet ( $w_0=6$  en la fig)



Para calcular la transformada es necesario elegir el no. de escalas, que dice cuanto se va a estirar la wavelet para mapear la variabilidad en la serie temporal

```
scales = 1 : 120;
```

```
coefs = cwt(serie,scales,'morl'); %misma serie usada  
para el espectro de potencia evolutivo
```

```
f = scal2frq(scales,'morl',3); %convierte escala a psueod  
frecuencias
```

```
%Grafica el espectro de potencia de wavelets: valor  
absoluto de los coeficientes de wavelets
```

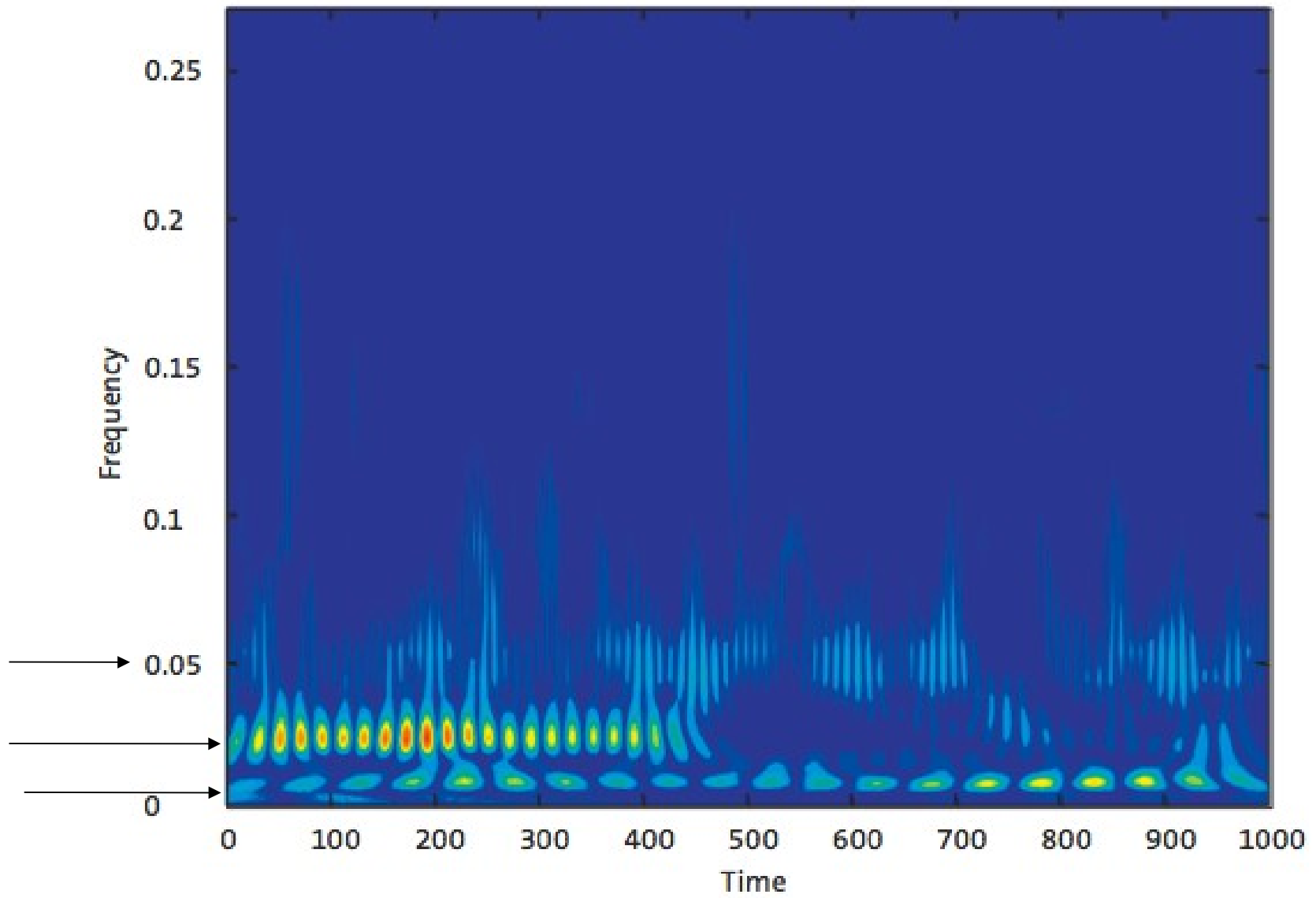
```
contour(t,f,abs(coefs),'LineStyle', 'none', ...
```

```
'LineColor',[0 0 0],'Fill','on')
```

```
xlabel('Time'),ylabel('Frequency')
```

```
title('Wavelet Power Spectrum')
```

Wavelet Powerspectrum



## Software de Wavelet

<http://paos.colorado.edu/research/wavelets/software.html>

