

N=11  
2ES.



# UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES  
" ACATLAN "

## LOCALIZACION EN ESPACIO FASE DE SEÑALES UNIDIMENSIONALES

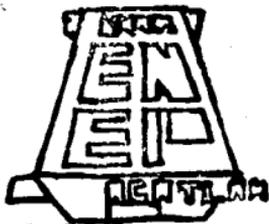
### T E S I S

Que para obtener el título de:  
LICENCIADO EN MATEMATICAS APLICADAS  
Y COMPUTACION

P r e s e n t a :

IGNACIO DAVID UGALDE OJEDA

Director de Tesis:  
DR. ALFREDO GOMEZ RODRIGUEZ



1992

TESIS CON  
FALLA DE ORIGEN



## **UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso**

### **DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

## INDICE

INTRODUCCION.....	3
-------------------	---

### CAPITULO 1 TRANSFORMADA DE GABOR.

1.1 Definición de una Transformada de Gabor.....	6
1.2 Propiedades de la Transformada de Gabor.....	12
1.3 Transformada Discreta de Gabor.....	14
1.4 Resolución de una Transformada de Gabor.....	15

### CAPITULO 2 APLICACIONES DE LA TRANSFORMADA DE GABOR.

2.1 Algoritmo de una Transformada de Gabor.....	18
2.2 Algoritmo para calcular la Transformada de Gabor en una computadora.....	19
2.3 Ejemplo 1.....	22
2.4 Ejemplo 2.....	24

**CONCLUSIONES.**

Conclusiones.....	26
Listado de los programas.....	29

**APENDICE A**

A.1 Transformada de Fourier.....	47
A.2 Propiedades de la Transformada de Fourier.....	48
A.3 Convolución y Correlación.....	51
A.4 Teorema de Parseval.....	53
A.5 Transformada Discreta de Fourier.....	54
A.6 La Transformada Rápida de Fourier (FFT).....	56
A.7 La Función Delta de Dirac.....	58
Bibliografía.....	59

## INTRODUCCION

La siguiente tesis tiene como objetivo, el estudio de la Transformada de Gabor (Transformada de Fourier con ventanas) así como su aplicación a las señales unidimensionales.

La Transformada de Fourier, utilizada en el procesamiento y análisis de señales, tiene aplicaciones en diferentes campos, como son la Biología, la Física y la Ingeniería. Cuando a una señal se le aplica la Transformada de Fourier, se obtiene su localización en el dominio de frecuencias, es bueno, pero no nos dice nada de su localización en el dominio del tiempo (posición).

Aunado a este problema cuando una señal tiene diferentes frecuencias, y se le calcula su Transformada de Fourier, ésta nos separa las frecuencias de la señal, pero no dirá en que tiempo ocurrieron.

Supongamos que usted se encuentra escuchando su concierto favorito y decide analizarlo. Para poderlo analizar se hará una gráfica de la señal, esta gráfica contendrá la amplitud de las notas en el tiempo, cada nota en un tiempo dado. Si le interesa saber las notas que fueron emitidas en el concierto, ocuparía la Transformada de Fourier, es decir a la señal se le calcula su transformada, en este cálculo nos dará una gráfica de amplitudes contra frecuencias, cada nota tiene una amplitud con una frecuencia.

Entonces usted sabrá cuantas notas fueron emitidas, pero no en que momentos ocurrieron, ya que están deslocalizadas en tiempo. Para resolver esto, nosotros ocuparemos la Transformada de Gabor que es una forma híbrida de la Transformada de Fourier.

La Transformada de Gabor nos permitirá localizar las señales tanto en frecuencias como en tiempo. Esta forma de localizar no es tan precisa, por el teorema de incertidumbre, si se localiza mejor en frecuencias hay un desbordamiento en el tiempo, si se localiza mejor en tiempos hay una pérdida en frecuencia, para poder resolverse, se utilizaron unas celdas de resolución (Fig 2), donde el eje vertical muestra la frecuencia  $\omega$ , y el eje horizontal la posición  $v$ .

Antes de adentrarnos en este problema, si no se recuerdan las propiedades básicas de la Transformada de Fourier sería recomendable revisara el apéndice A. En este apéndice se escribieron las propiedades más importantes, el teorema de convolución, el Teorema de Parseval y la Función Delta de Dirac.

En el primer capítulo se tratará el análisis matemático de la Transformada de Gabor y sus propiedades. En el segundo capítulo se aplicará la Transformada de Gabor a unos ejemplos, se describirá un algoritmo para calcularla y se ejecutará el programa realizado. Y, por último, se darán las conclusiones a que se llegó.

## TRANSFORMADA DE GABOR

Se analizará la Transformada de Gabor (Transformada de Fourier con ventana), sus aplicaciones, propiedades y se darán algunos ejemplos ilustrativos.

Notación.

Sea  $Z$  un entero,  $R$  número real y  $R^+$  número real positivo y denotamos el espacio de Hilbert de las funciones de cuadrado integrable, como  $L^2(\mathbb{R})$ .

Suponemos que nuestra señal es finita, está representada por una función  $f(x) \in L^2(\mathbb{R})$ . El producto interno de dos funciones se escribe :

$$\langle f(x), g(x) \rangle = \int_{-\infty}^{+\infty} f(x) \overline{g(x)} dx, \quad (1)$$

con  $f(x) \in L^2$  y  $g(x) \in L^2$ , donde  $\overline{g(x)}$  es el conjugado complejo de  $g(x)$ .

La norma de  $f(x)$  en  $L^2(\mathbb{R})$  está dada por

$$\|f\|^2 = \int_{-\infty}^{+\infty} |f(x)|^2 dx, \quad (2)$$

y denotamos la convolución de dos funciones  $f(x)$  y  $g(x)$  como

$$f(v) * g(v) = \int_{-\infty}^{+\infty} f(x) g(v-x) dx. \quad (3)$$

La Transformada de Fourier de  $f(x) \in L^2$  esta definida por

$$\hat{f}(\omega) = \int_{-\infty}^{+\infty} f(x) e^{-i2\pi\omega x} dx. \quad (4)$$

#### 1.1

#### DEFINICION DE UNA TRANSFORMADA DE GABOR (TRANSFORMADA DE FOURIER CON VENTANA)

Usando la Transformada de Fourier de una función  $f(x)$ , la localización de la señal en el espacio de las frecuencias es adecuada, pero se encuentra deslocalizada en posición (tiempo).

Para esto Gabor definió una nueva descomposición usando una ventana en la integral de Fourier. Esta ventana es trasladada a lo largo de los ejes para cubrir la señal. Tiene una posición  $v$  y una frecuencia  $\omega$ , la transformada de Fourier con ventana de una función  $f(x) \in L^2(\mathbb{R})$  está definida por :

$$Gf(\omega, v) = \int_{-\infty}^{+\infty} e^{-12\pi i \omega x} g(x - v) f(x) dx. \quad (5)$$

En la original Transformada de Gabor, la función ventana  $g(x)$  es una Gaussiana (Fig 1). La función de ventana es generalmente una función real par y el módulo de la transformada de esta función es concretamente de bajas frecuencias.

Para normalizar lo propuesto, supongamos que el módulo de  $g(x)$  es igual a 1 es decir :

$$\|g\|^2 = \int_{-\infty}^{+\infty} |g(x)|^2 dx = 1. \quad (6)$$

Una Transformada de Gabor puede ser interpretada como un producto interno. Para comprobar esto, se introduce una familia de funciones

$$g_{\omega_0, v_0}(x) = e^{12\pi i \omega_0 x} g(x - v_0) \quad (7)$$

y de la función  $f(x)$  con la familia de funciones  $(g_{\omega, v}(x))_{(\omega, v) \in \mathbb{R}^2}$ , se tiene el producto interno.

$$Gf(\omega, v) = \langle f(x), g_{\omega, v}(x) \rangle \quad (8)$$

En la física cuántica, esta familia de funciones es llamada, familia de estados coherentes. La Transformada de Fourier de  $(g_{\omega, \nu}(x))_{(\omega, \nu) \in \mathbb{R}^2}$

$$g_{\omega_0, \nu_0}(x) = e^{i2\pi\omega_0 x} g(x - \nu_0),$$

está dada por

$$\hat{g}_{\omega_0, \nu_0}(\omega) = e^{-i2\pi\nu_0(\omega - \omega_0)} \hat{g}(\omega - \omega_0). \quad (9)$$

Desarrollaremos la Transformada de Fourier de la función  $(g_{\omega, \nu}(x))_{(\omega, \nu) \in \mathbb{R}^2}$ . Para facilitar los cálculos, separaremos a la función en dos funciones y calcularemos la transformadas por separado; después utilizaremos el teorema de convolución.

$$g_{\omega_0, \nu_0}(x) = e^{i2\pi\omega_0 x} g(x - \nu_0),$$

separando la función

$$h = e^{i2\pi\omega_0 x} \quad \text{y} \quad r = g(x - \nu_0),$$

entonces la transformada de  $h$  es

$$\begin{aligned} \hat{h}(\omega) &= \int_{-\infty}^{+\infty} e^{i2\pi\omega_0 x} e^{-i2\pi\omega x} dx \\ &= \int_{-\infty}^{+\infty} e^{(\omega_0 - \omega)i2\pi x} dx, \end{aligned}$$

esta integral es una función delta ,

$$\hat{h}(\omega) = \delta(\omega - \omega_0)$$

, la transformada de r

$$\hat{r}(\omega) = \int_{-\infty}^{+\infty} g(x - v_0) e^{-i2\pi\omega x} dx$$

por el teorema de translación en el tiempo

$$\hat{r}(\omega) = e^{-i2\pi\omega v_0} \hat{g}(\omega).$$

Ahora por el teorema de convolución , la convolución de una Función Delta con otra función, es la función desplazada en frecuencia,

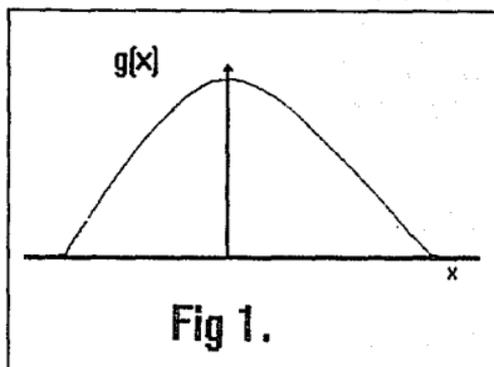
$$\hat{h}(\omega) * \hat{r}(\omega) = \delta(\omega - \omega_0) * e^{-i2\pi\omega v_0} \hat{g}(\omega)$$

$$= e^{-i2\pi v_0(\omega - \omega_0)} \hat{g}(\omega - \omega_0).$$

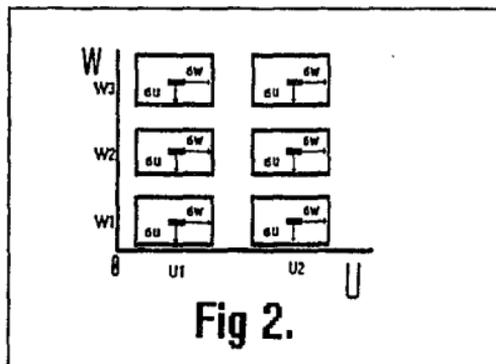
es la Transformada de Fourier de  $(g_{\omega, v}(x))_{(\omega, v) \in \mathbb{R}^2}$  .

Una familia de estados coherentes, corresponde a la translación en el dominio del tiempo (parámetro  $v$ ) y en el dominio de la frecuencia (parámetro  $\omega$ ) de la función  $g(x)$ .

Esta doble translación es representada en un plano fase-espacio, donde un eje corresponde al parámetro del tiempo  $v$ , y el otro al parámetro de frecuencias  $\omega$  (Fig 2).



**GAUSSIANA**



Representación del espacio - fase, el eje vertical muestra la frecuencia  $w$ , el eje horizontal nos muestra la posición  $u$ . Una Transformada de Gabor con coeficientes  $G_f(w,u)$  nos da una descripción de  $f(x)$ , con la resolución de la celda.

Para poder describir cómo una Transformada de Gabor se representa en frecuencia y tiempo, denotamos por  $\sigma_v$  a la desviación estándar de  $g(x)$

$$\sigma_v^2 = \int_{-\infty}^{+\infty} x^2 |g(x)|^2 dx, \quad (10)$$

y por  $\sigma_\omega$  a la desviación estándar de la Transformada de Fourier de  $g(x)$ :

$$\sigma_\omega^2 = \int_{-\infty}^{+\infty} \omega^2 |\hat{g}(\omega)|^2 d\omega. \quad (11)$$

La función  $g_{\omega_0, v_0}(x)$  es centrada en  $v_0$  y tiene una desviación estándar  $\sigma_v$  en el tiempo (posición). La Transformada de Fourier dada en (9) es centrada en  $\omega_0$  y tiene una desviación estándar  $\sigma_\omega$ .

De la interpretación de la Transformada de Gabor como un producto interno y de la aplicación del teorema de Parseval en (8), tenemos que

$$Gf(\omega_0, v_0) = \int_{-\infty}^{+\infty} f(x) \overline{g_{\omega_0, v_0}(x)} dx,$$

que se puede reescribir como

$$Gf(\omega_0, v_0) = \int_{-\infty}^{+\infty} \hat{f}(\omega) \overline{\hat{g}_{\omega_0, v_0}(\omega)} d\omega. \quad (12)$$

Desarrollaremos lo propuesto para verificarlo, se utilizará la definición de Transformada Inversa y la función Delta.

$$\begin{aligned}
 Gf(\omega_0, \nu_0) &= \int_{-\infty}^{+\infty} f(x) \overline{g_{\omega_0, \nu_0}(x)} \, dx \\
 &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \hat{f}(\omega) e^{i2\pi\omega x} \, d\omega \int_{-\infty}^{+\infty} \overline{\hat{g}_{\omega_0, \nu_0}(\omega') e^{i2\pi\omega' x}} \, d\omega' \, dx \\
 &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \hat{f}(\omega) \overline{\hat{g}_{\omega_0, \nu_0}(\omega')} e^{i2\pi\omega x} e^{-i2\pi\omega' x} \, dx \, d\omega' \, d\omega \\
 &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \hat{f}(\omega) \overline{\hat{g}_{\omega_0, \nu_0}(\omega')} \, \delta(\omega - \omega') \, d\omega' \, d\omega \\
 &= \int_{-\infty}^{+\infty} \hat{f}(\omega) \overline{\hat{g}_{\omega_0, \nu_0}(\omega)} \, d\omega.
 \end{aligned}$$

La primera integral muestra que en el dominio del tiempo (posición),  $Gf(\omega_0, \nu_0)$  depende esencialmente de los valores de  $f(x)$  para  $x \in [\nu_0 - \sigma_\nu, \nu_0 + \sigma_\nu]$ . La segunda integral prueba que en el dominio de las frecuencias,  $Gf(\omega_0, \nu_0)$  depende de los valores de  $\hat{f}(\omega)$  para  $\omega \in [\omega_0 - \sigma_\omega, \omega_0 + \sigma_\omega]$ .

El dominio de frecuencia-tiempo, el cual es cubierto por  $Gf(\omega_0, \nu_0)$  puede ser representado en el espacio fase por la celda de resolución  $[\nu_0 - \sigma_\nu, \nu_0 + \sigma_\nu] \times [\omega_0 - \sigma_\omega, \omega_0 + \sigma_\omega]$  como se muestra en la figura 2. La superficie y forma de la celda de resolución es independiente para  $\nu_0$  y  $\omega_0$ . De aplicar el principio de incertidumbre a la función  $g(x)$  implica que

$$\sigma_\nu^2 \sigma_\omega^2 \geq 1/4. \quad (13)$$

## 1.2

### PROPIEDADES DE LA TRANSFORMADA DE GABOR

Una Transformada de Gabor es isométrica para  $L^2(\mathbb{R})$  a  $L^2(\mathbb{R}^2)$  (teorema de Parseval).

$$\int_{-\infty}^{+\infty} |f(x)|^2 dx = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} |Gf(\omega_0, \nu_0)|^2 d\omega d\nu \quad (14)$$

La función  $f(x)$  se puede reconstruir desde  $Gf(\omega, \nu)$  con la fórmula

$$f(x) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} Gf(\omega_0, \nu_0) g(\nu - x) e^{i2\pi\omega x} d\omega d\nu \quad (15)$$

Desarrollaremos la fórmula (15) para comprobarla, se utilizarán las definiciones de Transformada de Fourier, Transformada de Gabor y la Función Delta de Dirac.

$$\begin{aligned} f(x) &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} e^{-i2\pi\omega'x} g(x - \nu) f(x) g(\nu - x) e^{i2\pi\omega x} d\omega d\nu dx \\ &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} g(x - \nu) f(x) g(\nu - x) e^{i2\pi\omega x} e^{-i2\pi\omega'x} dx d\omega d\nu \\ &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} g(x - \nu) f(x) g(\nu - x) \delta(\omega - \omega') d\omega d\nu \\ &= \int_{-\infty}^{+\infty} g(\nu)^2 f(x) d\nu \end{aligned}$$

y por (6)

$$= f(x).$$

En sí se puede decir que la fórmula (15) es la Transformada Inversa de Gabor.

### 1.3

#### TRANSFORMADA DISCRETA DE GABOR

Utilizaremos la Transformada Discreta de Gabor, para calcular la transformada en una computadora. Hagamos que para  $Gf(\omega, \nu)$  los valores de  $(\omega, \nu) \in \mathbb{R}^2$ , se tomen una muestra uniformemente de  $\omega$  y  $\nu$ , entonces tenemos que  $\omega_0$  y  $\nu_0$  son intervalos de muestras en ambos dominios, y una Transformada de Gabor Discreta,

$$\forall n \in \mathbb{Z}, \quad \forall m \in \mathbb{Z}$$

esta defina por

$$G_d f(m, n) = Gf(m \omega_0, n \nu_0) =$$

$$= \int_{-\infty}^{+\infty} e^{-i2\pi m \omega_0 x} g(x - n \nu_0) f(x) dx \quad (16)$$

La discretización corresponde a saltos uniformes del plano espacio-fase como se muestra en la figura 3. La transformada discreta de Gabor es equivalente a la división del eje de las frecuencias en intervalos separados por  $\omega_0$ . En cada uno de estos intervalos, la señal se muestreo tomando un valor de  $1/\nu_0$ .

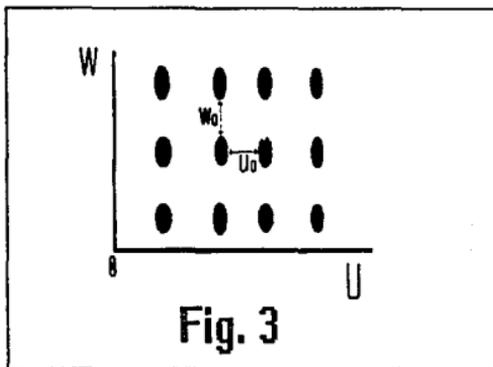
Una transformada discreta de Gabor puede ser expresada como un producto interno en  $L^2(\mathbb{R})$

$$\begin{aligned}
 G_d f(m,n) &= \langle f(x) , e^{i2\pi m_0 x} g(x - nu_0) \rangle \\
 &= \langle f(x) , g_{m_0, nu_0}(x) \rangle \quad (17)
 \end{aligned}$$

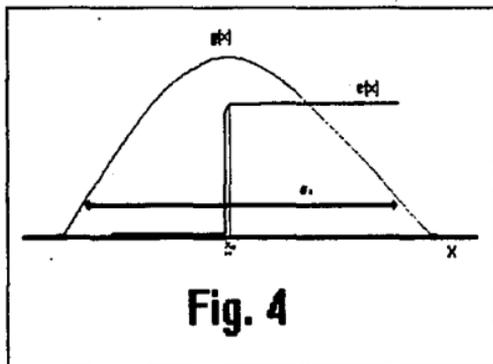
#### 1.4

#### RESOLUCION DE UNA TRANSFORMADA DE GABOR

La resolución de una transformada de Gabor en tiempo (posición) y frecuencia, es constante. En el dominio del tiempo, la información proviene de la descomposición, es por lo tanto deslocalizada con intervalos de tamaño  $\sigma_u$ . La desviación estándar  $\sigma_u$  de  $g(x)$ , define una resolución de referencia. Si la señal tiene una discontinuidad como un borde, con una Transformada de Gabor, es difícil la localización de este borde con una precisión mejor que  $\sigma_u$  (fig 4). Esta limitación generalmente no es aceptable. Si la señal tiene importantes formas de tamaños diferentes, no vamos a poder definir una resolución óptima para analizarla. Con la ventana, es difícil que se analicen ambos, lo fino y lo largo de la estructura. Esta resolución fija introduce frecuencias altas engañosas con una descomposición de forma local.



Muestreo de una Transformada Discreta de Gabor en el espacio - fase. La resolución de las celdas son idénticas en todo el espacio - fase, el muestreo es uniforme.



Con una Transformada de Gabor, una figura como un borde  $e(x)$  no puede ser localizado, con una precisión mejor que la desviación estándar de la función ventana  $g(x)$ .

Veamos un ejemplo, tenemos que  $e(x)$  es un borde como se muestra en la fig 4 y suponemos que

$$e(x) = \begin{cases} 0 & \text{si } x \leq x_0 - \Delta x/2 \\ 1/2 + 1/2 \operatorname{sen}(\pi/\Delta x(x - x_0)) & \text{si } x_0 - \Delta x/2 < x < x_0 + \Delta x/2 \\ 1 & \text{si } x \geq x_0 + \Delta x/2 \end{cases}$$

Hagamos  $\omega_0 = \pi/\Delta x$ , y supongamos que sobre un punto  $x_0$ , se pueden expresar sus coeficientes como  $G_e(\omega, x_0) = \langle e(x), e^{i2\pi\omega x} g(x - x_0) \rangle$  y decrece rápidamente cuando  $\omega$  es más largo que  $\omega_0$ . En la vecindad de  $x_0$ , el borde  $e(x)$  es una onda senoidal de frecuencia  $\omega_0$ . En sí el borde está bien localizado y tiene solamente la mitad del período de la onda senoidal.

Como una consecuencia de esto, cuando la frecuencia  $\omega$  es más larga con respecto a  $\omega_0$ , los módulos de los coeficientes  $G_e(\omega, x_0)$  decrecen lentamente. Aunque la señal  $e(x)$  es localizada como una onda senoidal de frecuencia  $\omega_0$ , cuando la frecuencia es  $2\omega_0$ , los coeficientes de Gabor de  $|G_e(2\omega_0, x_0)|$ , son como la mitad del valor de  $|G_e(\omega_0, x_0)|$ .

Una Transformada de Gabor es útil para analizar señales, cuando todas las muestras aparecen aproximadamente con la misma escala.

## APLICACIONES DE LA TRANSFORMADA DE GABOR

Los ejemplos que a continuación se tratan, son una forma para entender mejor como trabaja la Transformada de Gabor. Se realizó un programa en lenguaje C y se usó el paquete de MatLab, para resolver numéricamente esta transformada, se ocupó la Transformada Rápida de Fourier (The Fast Fourier Transform) que se representa con las siglas de FFT.

La transformada discreta que se dio anteriormente fué :

$$Gf(m \omega_0, n \nu_0) = \int_{-\infty}^{+\infty} e^{-i2\pi \omega_0 x} g(x - n\nu_0) f(x) dx$$

para  $\forall n \in \mathbb{Z}$  ,  $\forall m \in \mathbb{Z}$

la ventana que se utilizará será la Gaussiana

$$g = e^{-x^2/2a^2} \tag{15}$$

## 2.1

### ALGORITMO PARA UNA TRANSFORMADA DE GABOR

Recordemos que la función que introduciremos tiene que estar muestreada.

Donde

$F(x)$  : función.

$N$  : número de puntos de la muestra.

$\Delta t$  : paso de la muestra de la función.

I) Se calcula la Gaussiana con el vector de desplazamiento

a) Vector de desplazamiento. ( $x - nu_0$ )

Este vector nos servirá para desplazar toda la Gaussiana por la señal. El valor que tomará  $u_0$  será el del paso de muestreo de la señal,  $u_0 = \Delta t$ .

b) Dado el vector se calcula la Gaussiana.

II) Se multiplica la Gaussiana calculada por la función  $F(x)$

$$F(x)' = g(x - nu_0) F(x)$$

III) A la multiplicación  $F(x)'$ , se calcula la FFT y se guarda el resultado en una matriz.

Este procedimiento se hará  $n$  veces.

Entonces la matriz que se forma tendrá la transformada de la función  $F(x)$ , es decir cada renglón es la Transformada de Gabor de  $F(x)$  en el tiempo  $n$ .

$$\begin{array}{cccccc}
 & 1 & 2 & 3 & \dots & N \\
 \begin{array}{c} 1 \\ \cdot \\ \cdot \\ \cdot \\ n \end{array} & \left[ \begin{array}{c} \text{FFT de } F(x) \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{array} \right]
 \end{array}$$

La frecuencia que se calcule con la Transformada de Gabor, debe concordar con la frecuencia de la Transformada de Fourier de la señal.

## 2.2

### ALGORITMO PARA CALCULAR LA T. DE GABOR EN UNA COMPUTADORA

Pseudo Algoritmo.

Para calcular la Transformada Discreta de Gabor

$$Gf(m, \omega_0, n, \nu_0) = \int_{-\infty}^{+\infty} e^{-i2\pi(\omega_0 x - n\nu_0)} g(x - n\nu_0) f(x) dx$$

Entrada :

Función  $f(x)$ .  
Puntos de la muestra  $N$ .  
Paso de muestreo  $dx$ .  
Número de iteraciones  $n$ .  
Dilatación de la Gaussiana  $a$ .

Salida :

Matriz resultante de  $n \times N$ .

Paso 1

Inicializar vector de desplazamiento.  
Tomar  $w[i] = i \cdot dx$ ;

Paso 2

Para  $t=1$ ;  $t \leq n$  ;  $t++$

{

Paso 1

Para  $j=1$ ;  $j \leq N$ ;  $j++$

$w[j] = w[j] - t \cdot dx$ ;

Paso 2

$gauss(w, g, a)$ ; /\* Subrutina para el cálculo de  
la gaussiana.  
/\*  $g$  - valor de la gaussiana.

Paso 3

Para  $j=1$ ;  $j \leq N$ ;  $j++$

$h[j] = g[j] \cdot x[j]$ ;

$h1[j] = 0.0$ ;

/\* Multiplicación de la Gaussiana por la  
función.

Paso 4

$fft(h, h1, N, nu)$ ;

/\* Subrutina para el cálculo de la FFT de la multiplicación

Paso 5

```
Para j=1;j<=N;j++
ma[t][j] = h[j];
/* Asignar el valor de FFT a la
matriz ma.
```

Para probar el algoritmo se ejecutó el programa en C, el programa tiene varios ejemplos. La mayoría de estos con funciones sencidales.

Antes de ejecutar el programa, analizaremos cuál podría ser el resultado que esperamos.

La función que introduciremos es un seno. La transformada de Fourier de un seno, son dos deltas. La transformada de Fourier de una Gaussiana es otra Gaussiana.

Cuando multiplicamos en el algoritmo la función por la Gaussiana, y después calculamos su transformada de Fourier. Por el teorema de convolución, la transformada del seno por la Gaussiana sería lo mismo que la convolución de la transformada de cada función, entonces la convolución de las transformada del seno con la transformada de la Gaussiana, es una Gaussiana pero con la frecuencia de la Delta.

Entonces el resultado que se espera son cuatro Gaussianas separadas en frecuencia y tiempo. Cada frecuencia dos Gaussianas.

## 2.3

### EJEMPLO 1

Se analizará una señal de 2 senos, que se translapan en el tiempo, cada seno tendrá una envolvente.

Los datos son :

Número de puntos de la muestra N	:	64
Paso de muestreo	$\Delta t$ :	1
Frecuencia de la señal	f1 :	0.125
Frecuencia de la 2 <sup>a</sup> señal	f2 :	0.2
Dilatación de la Gaussiana	a :	10
Número de iteraciones en el tiempo n	:	10

Ejecución del programa...

## TRANSFORMADA DE GABOR



**1: Lee los datos de un archivo**



**2: Lee los datos desde el teclado**



**3: Ejecuta un ejemplo**



**4: Salir al D.O.S.**

**Introduzca la opción :**

## EJEMPLOS DE TRANSFORMADA DE GABOR



1:  $\text{Cos}(2\pi f t) + \text{Cos}(2\pi f t)$



2:  $\text{Sen}(2\pi f t) + \text{Sen}(2\pi f t)$

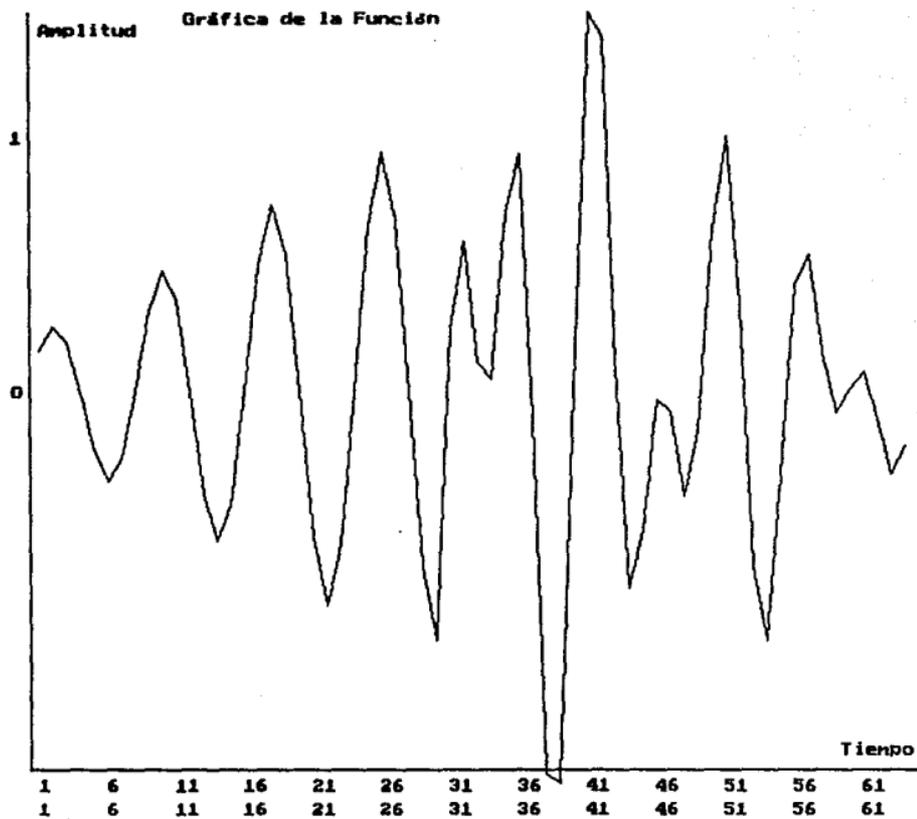


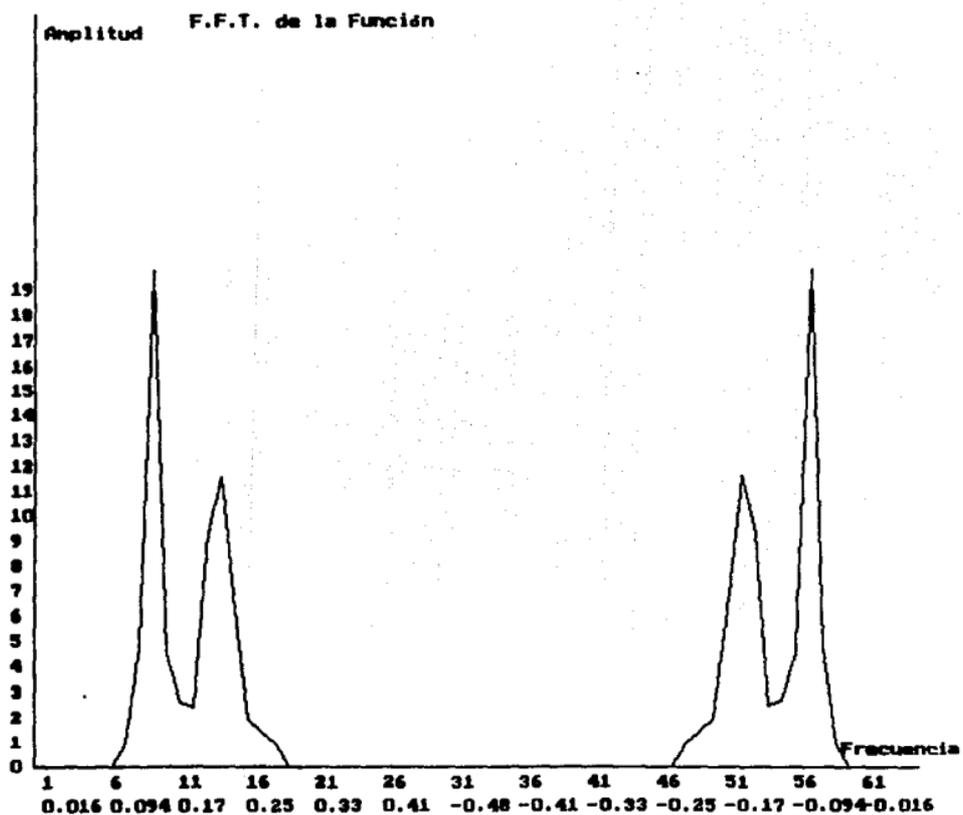
3:  $\text{Sen}(2\pi f t) + \text{Sen}(2\pi f t)$



4: Ejemplo de Frecuencia de Musica

Introduzca la opción :





Amplitud

Gráfica de la Gaussiana

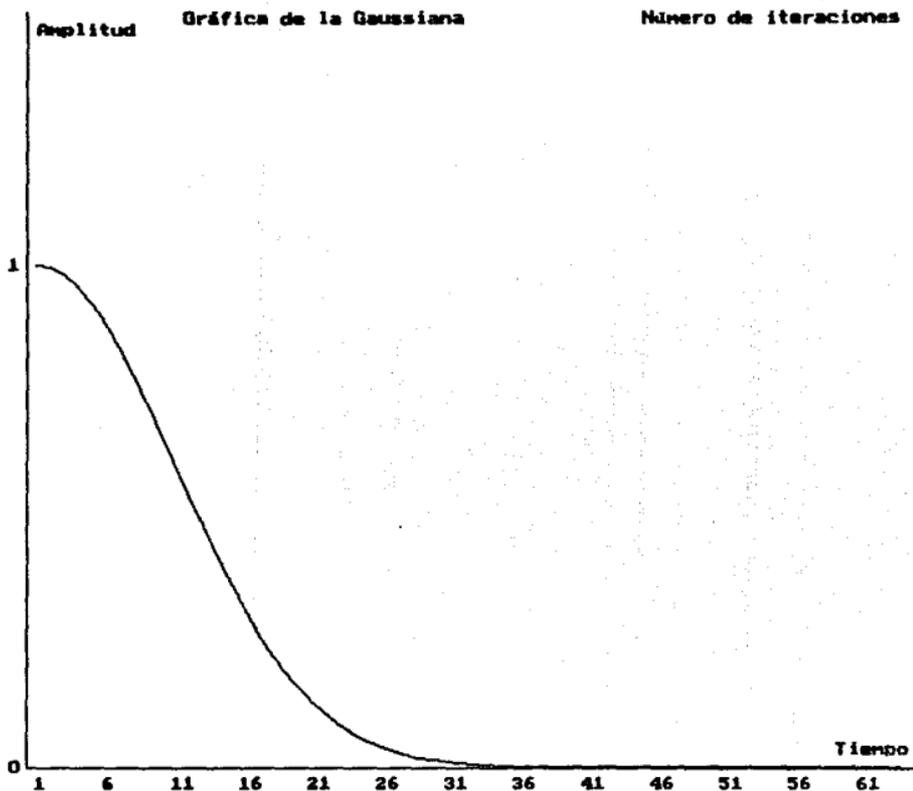
Número de iteraciones : 1

1

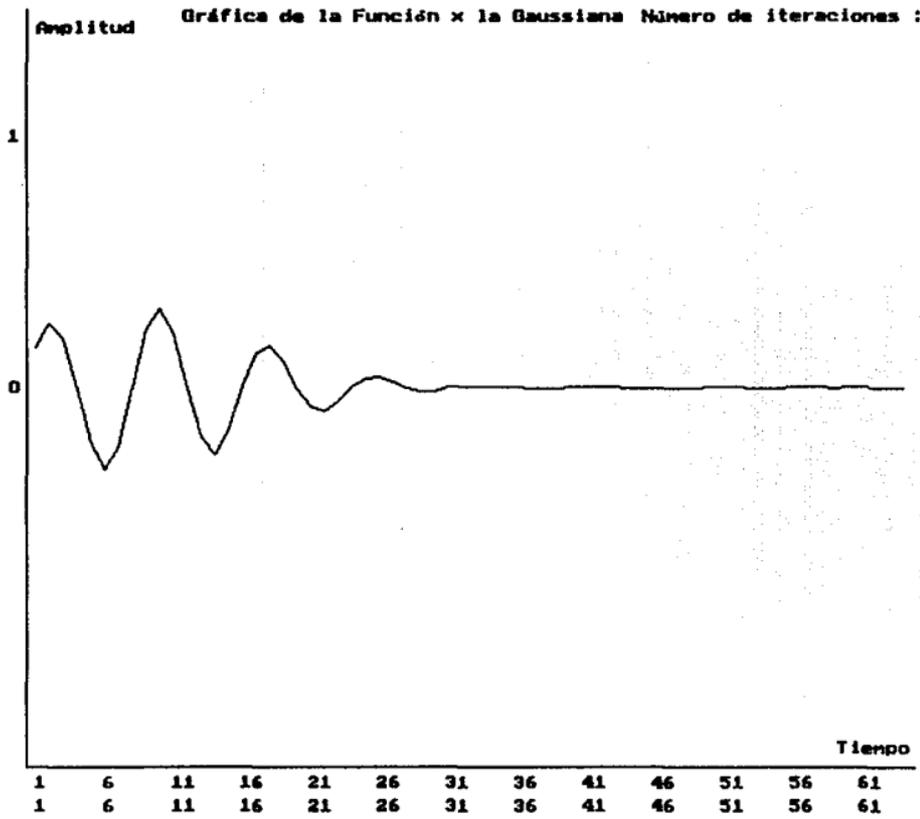
0

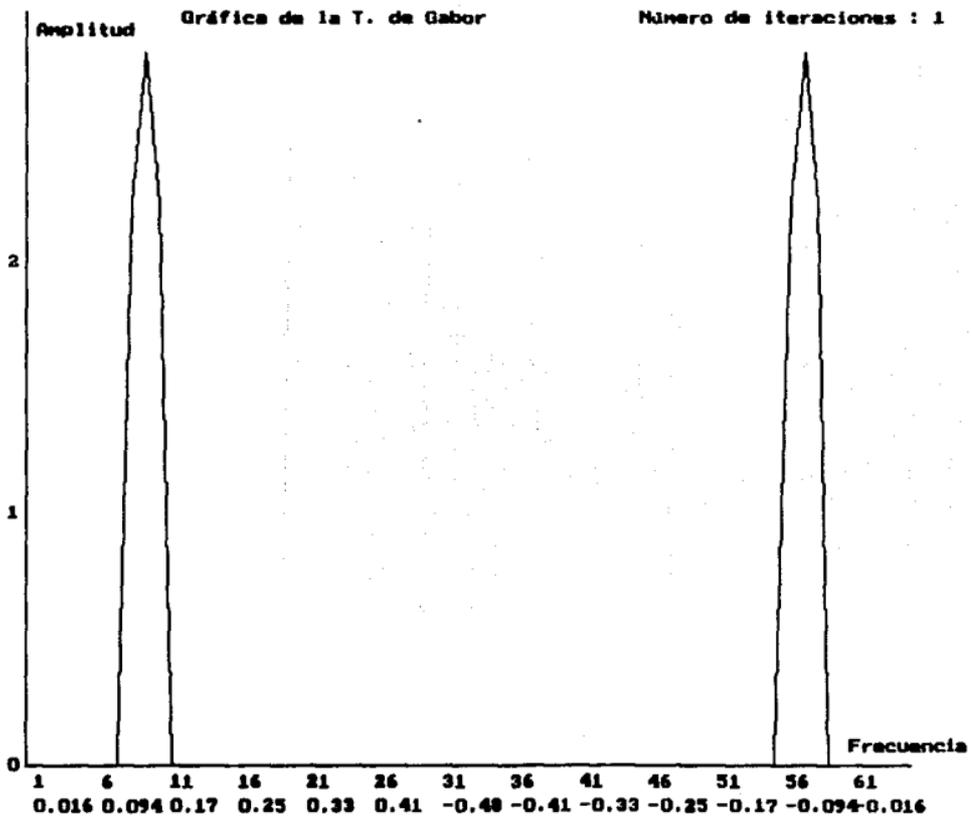
Tiempo

1 6 11 16 21 26 31 36 41 46 51 56 61



Gráfica de la Función x la Gaussiana Número de iteraciones : 1

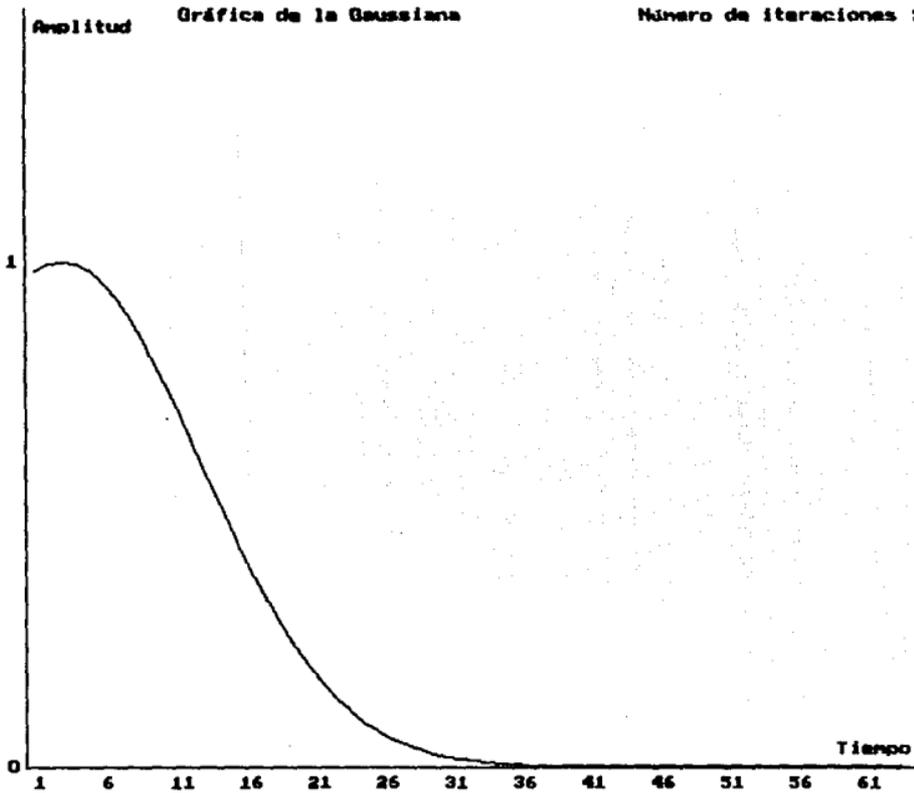


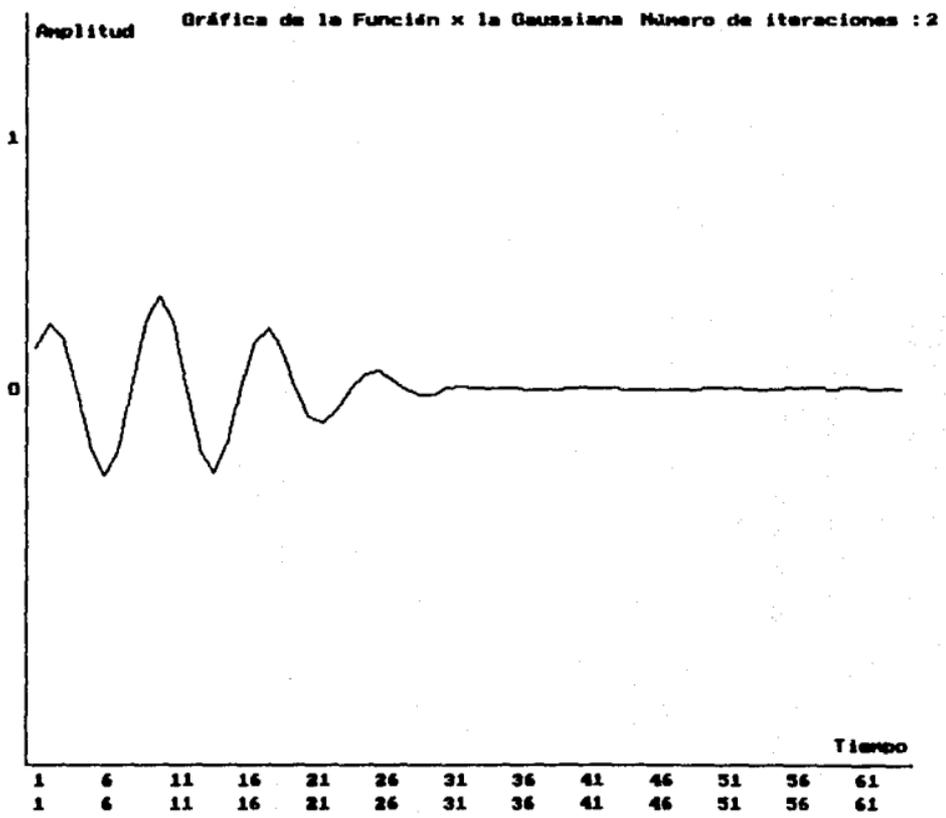


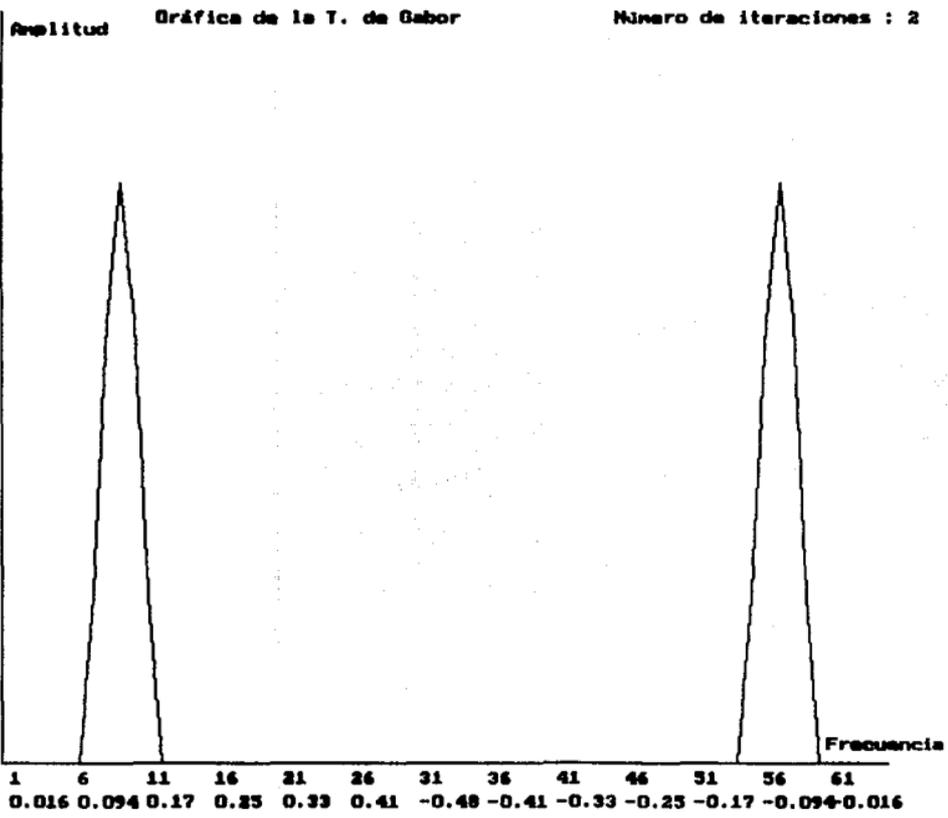
Amplitud

Gráfica de la Gaussiana

Número de iteraciones : 2



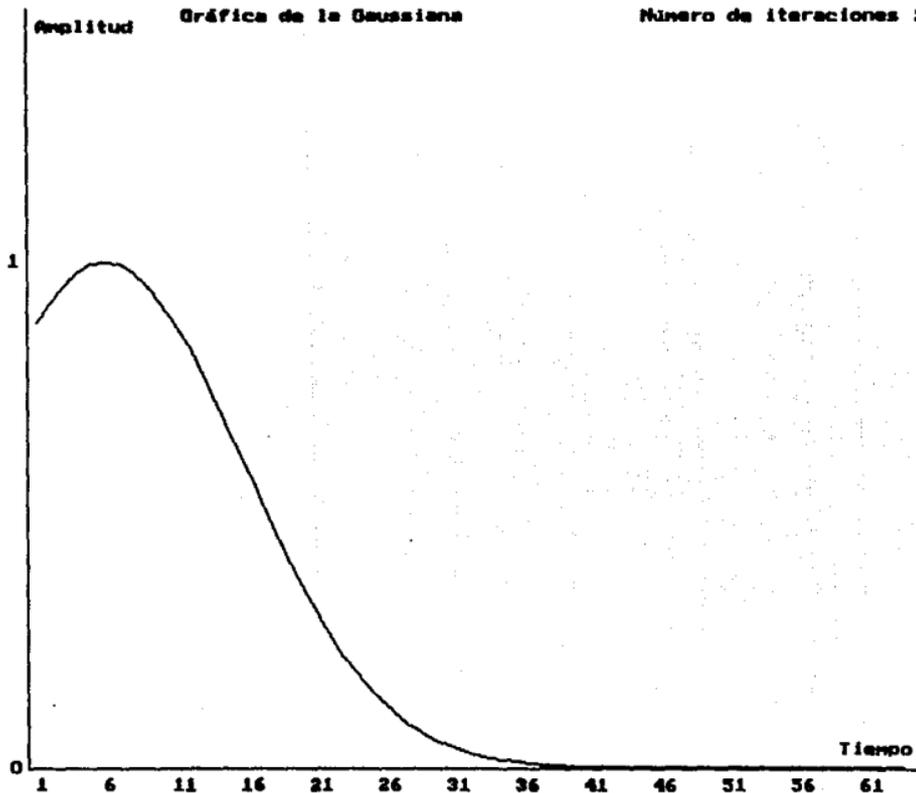




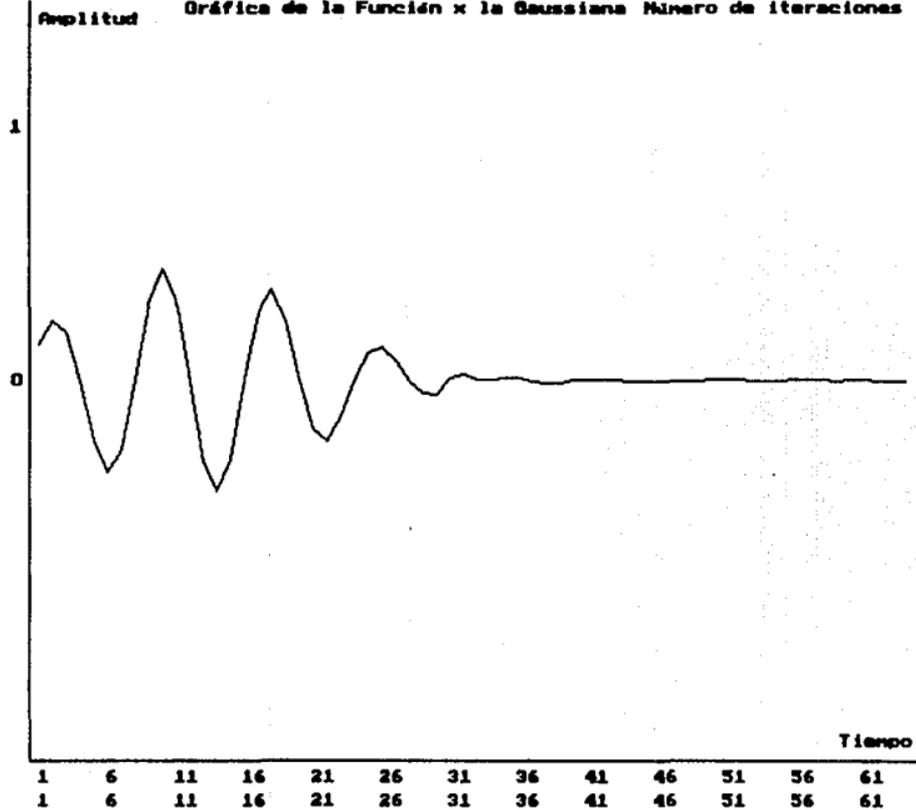
Amplitud

Gráfica de la Gaussiana

Número de iteraciones : 3

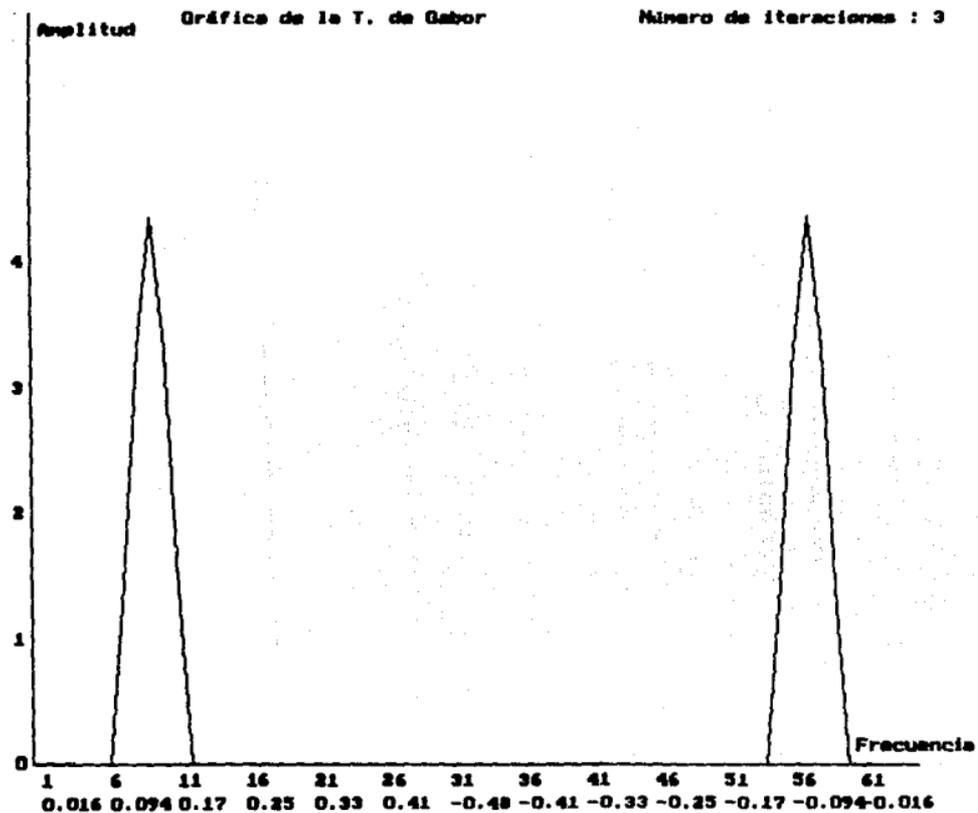


Gráfica de la Función x la Gaussiana Número de iteraciones : 3



Amplitud Gráfica de la T. de Gabor

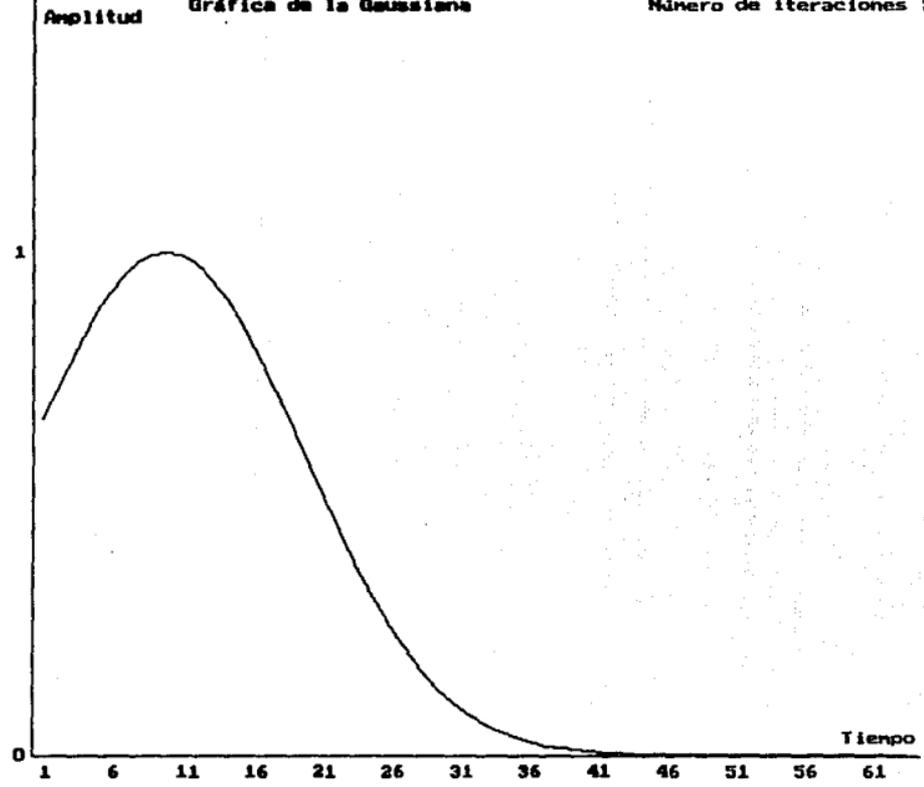
Número de iteraciones : 3



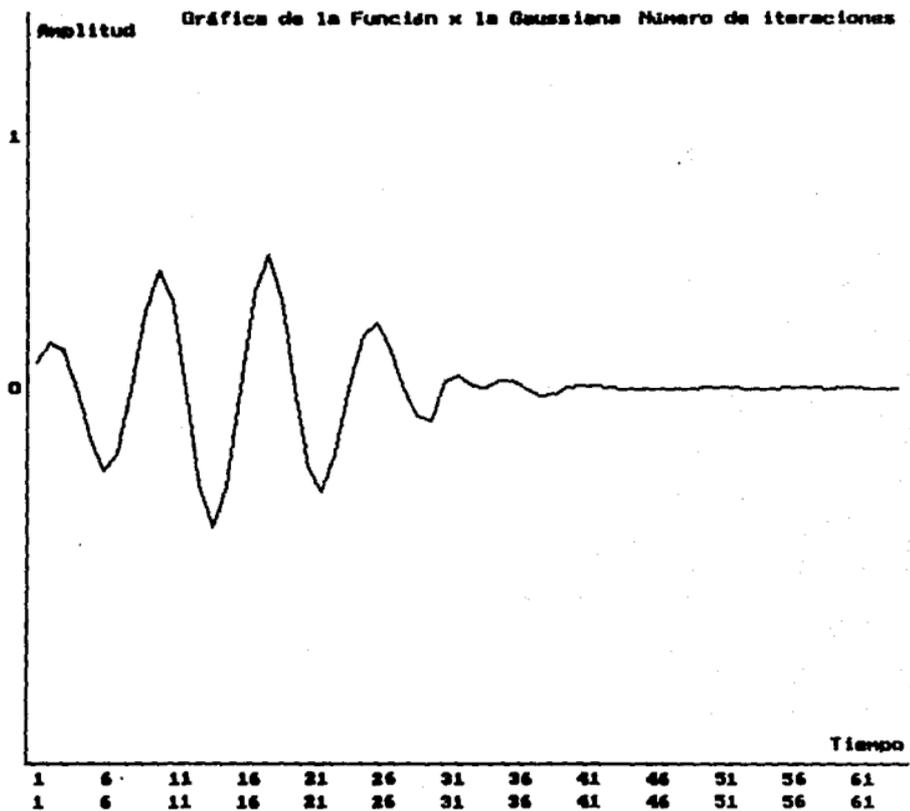
Amplitud

Gráfica de la Gaussiana

Número de iteraciones : 4



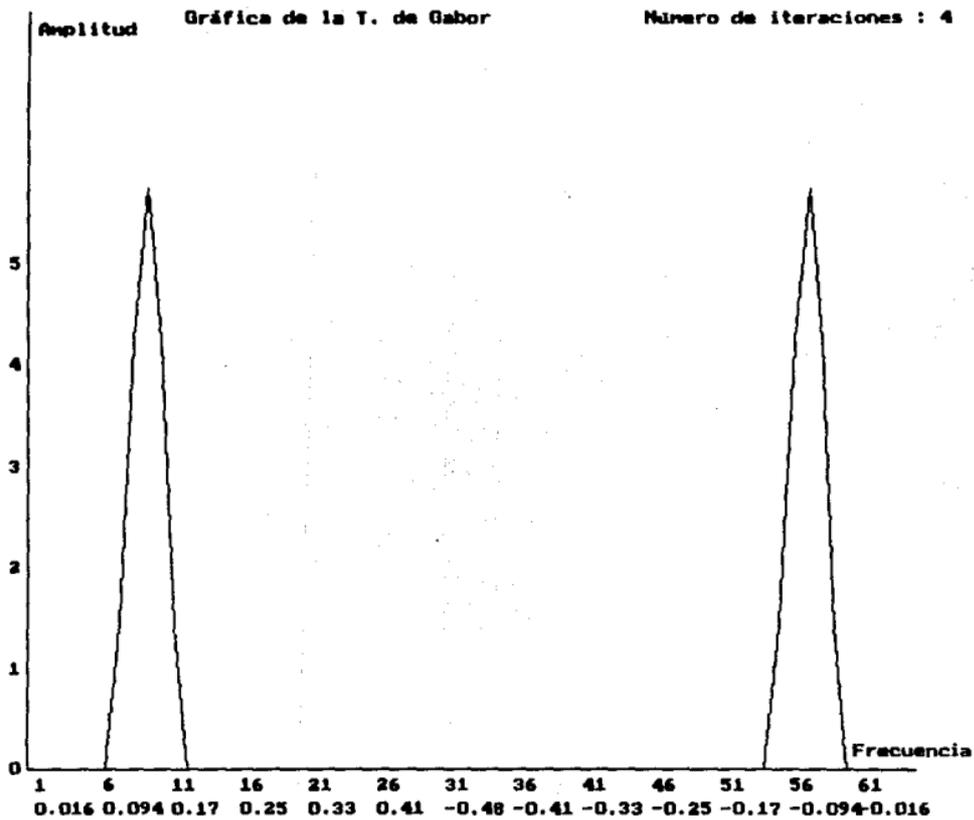
Gráfica de la Función x la Gaussiana Número de iteraciones : 4



Amplitud

Gráfica de la T. de Gabor

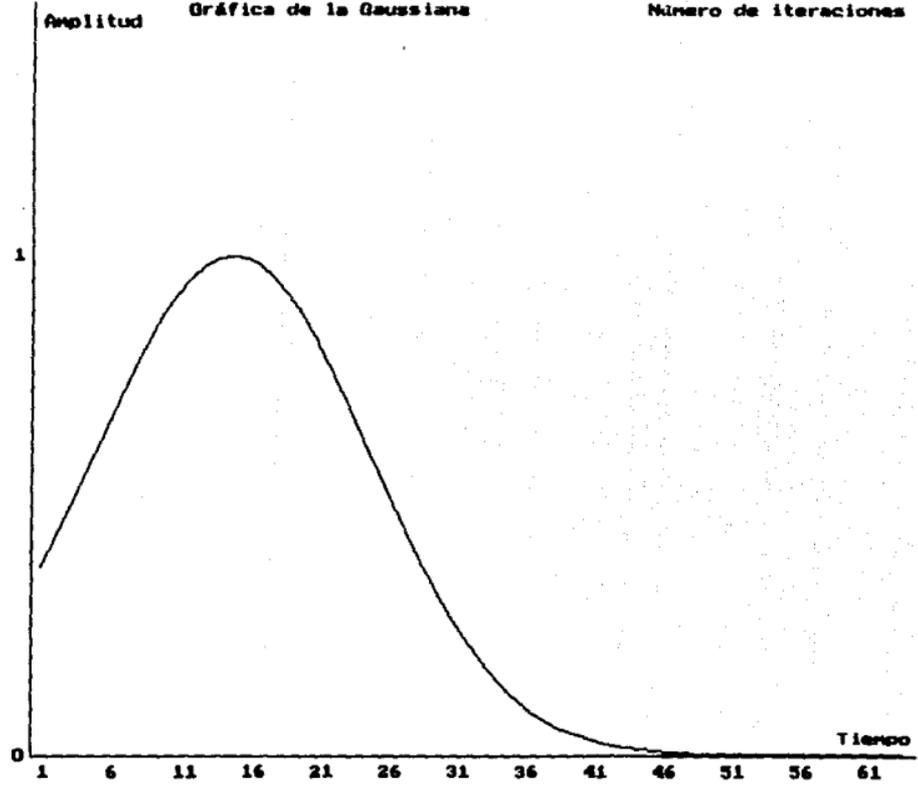
Número de iteraciones : 4



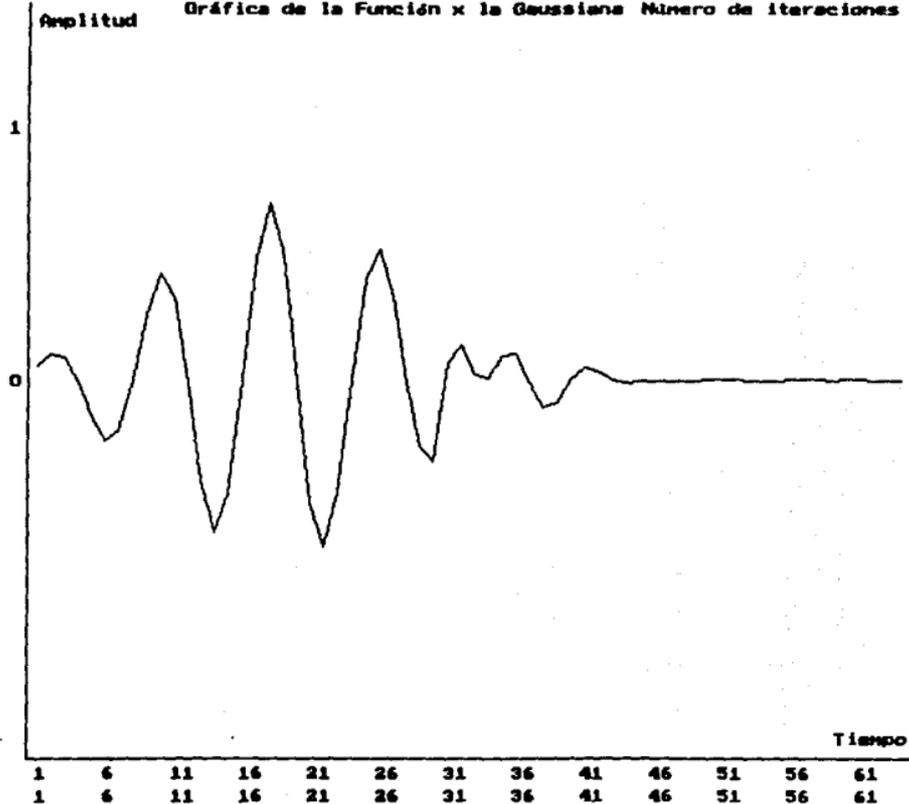
Amplitud

Gráfica de la Gaussiana

Número de iteraciones : 5

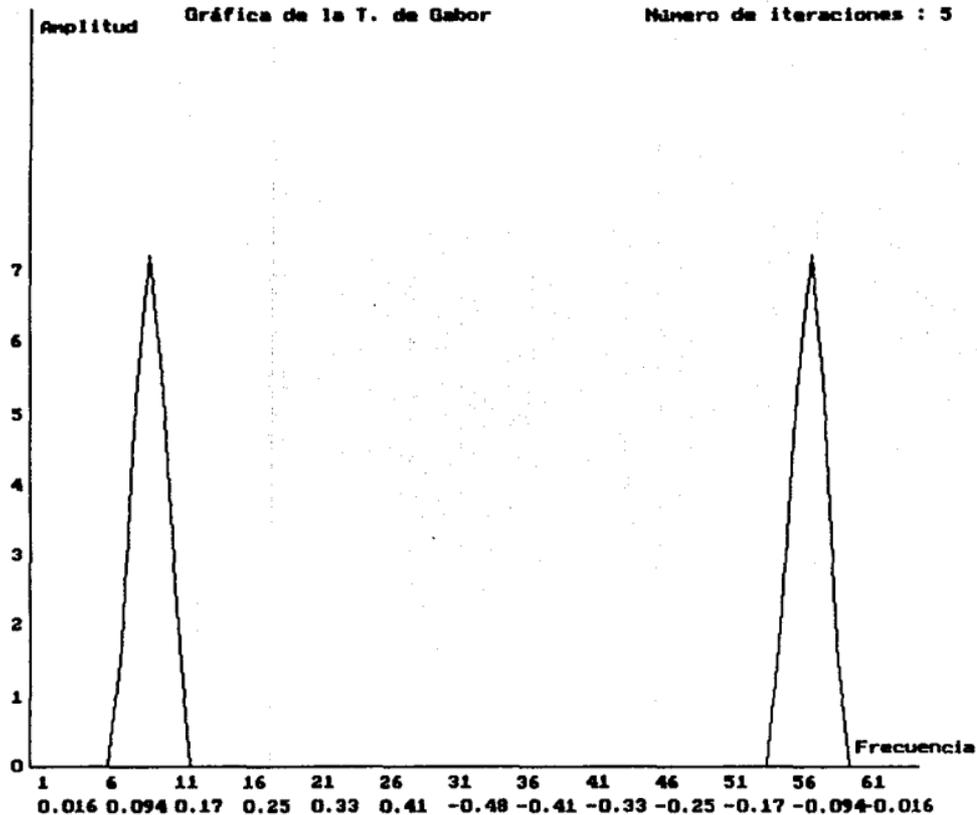


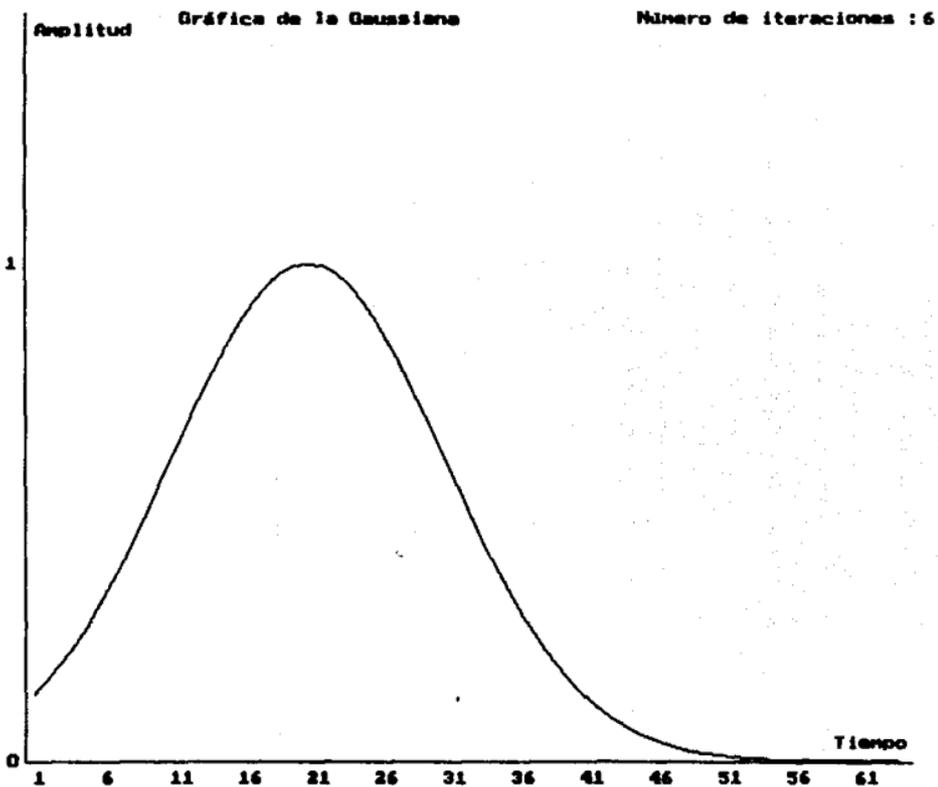
Gráfica de la Función x la Gaussiana Número de iteraciones : 5



Gráfica de la T. de Gabor

Número de iteraciones : 5





Amplitud

Gráfica de la Función x la Gaussiana Número de iteraciones : 6

1

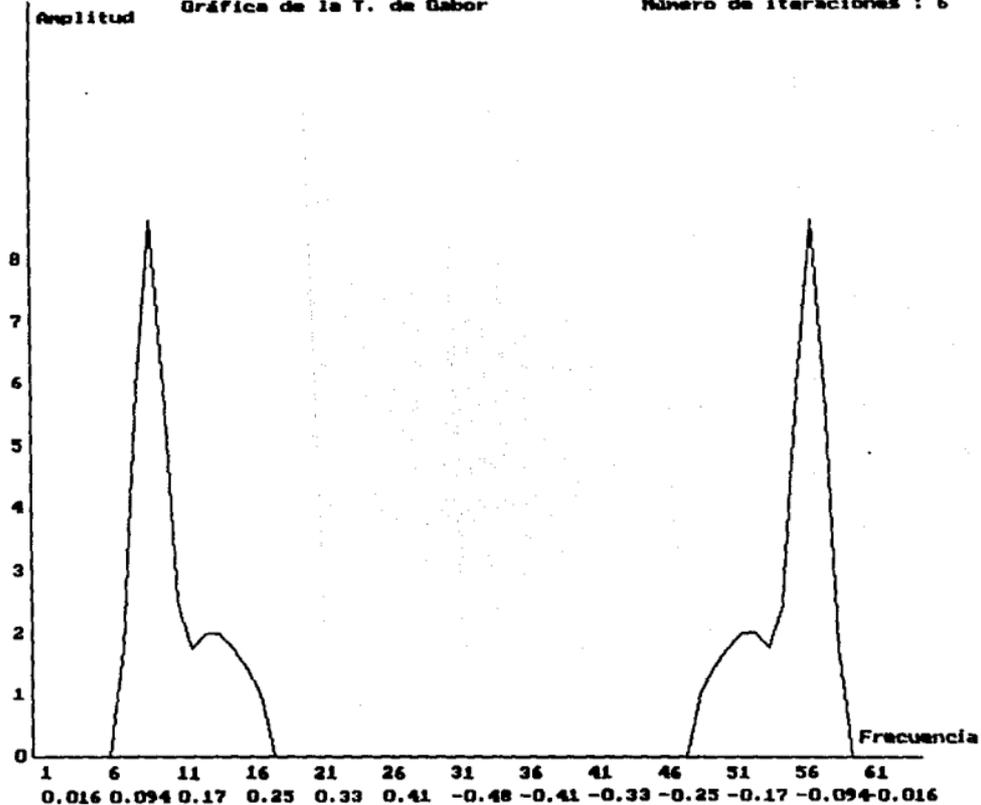
0

1 6 11 16 21 26 31 36 41 46 51 56 61  
1 6 11 16 21 26 31 36 41 46 51 56 61

Tiempo

Amplitud Gráfica de la T. de Gabor

Número de iteraciones : 6



Amplitud

Gráfica de la Gaussiana

Número de iteraciones : 7

1

0

Tiempo

1

6

11

16

21

26

31

36

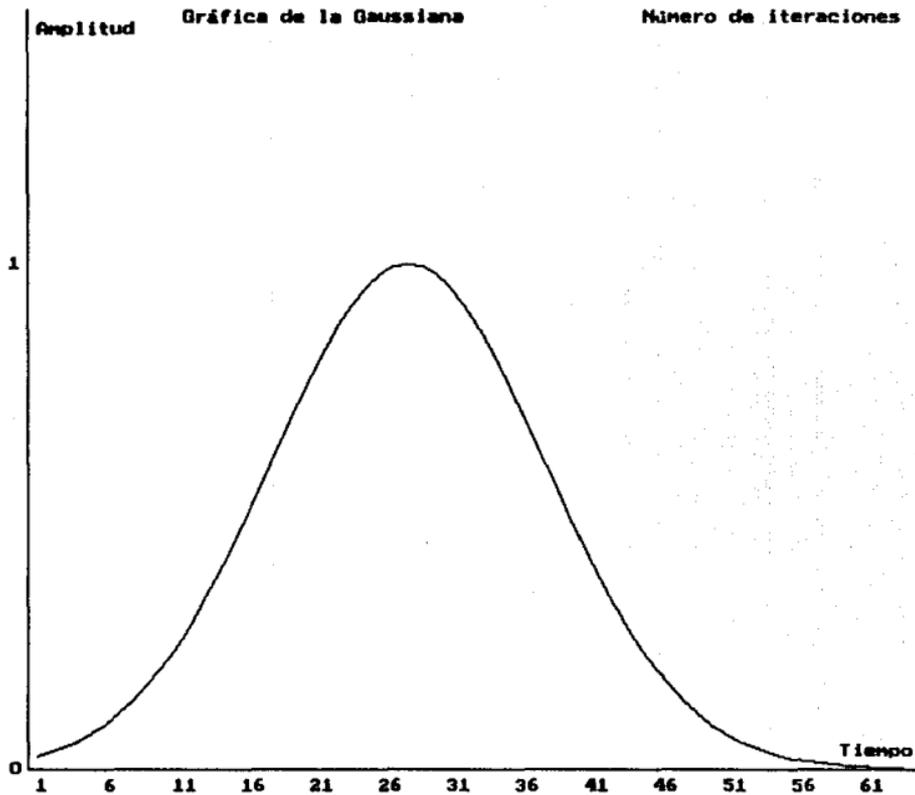
41

46

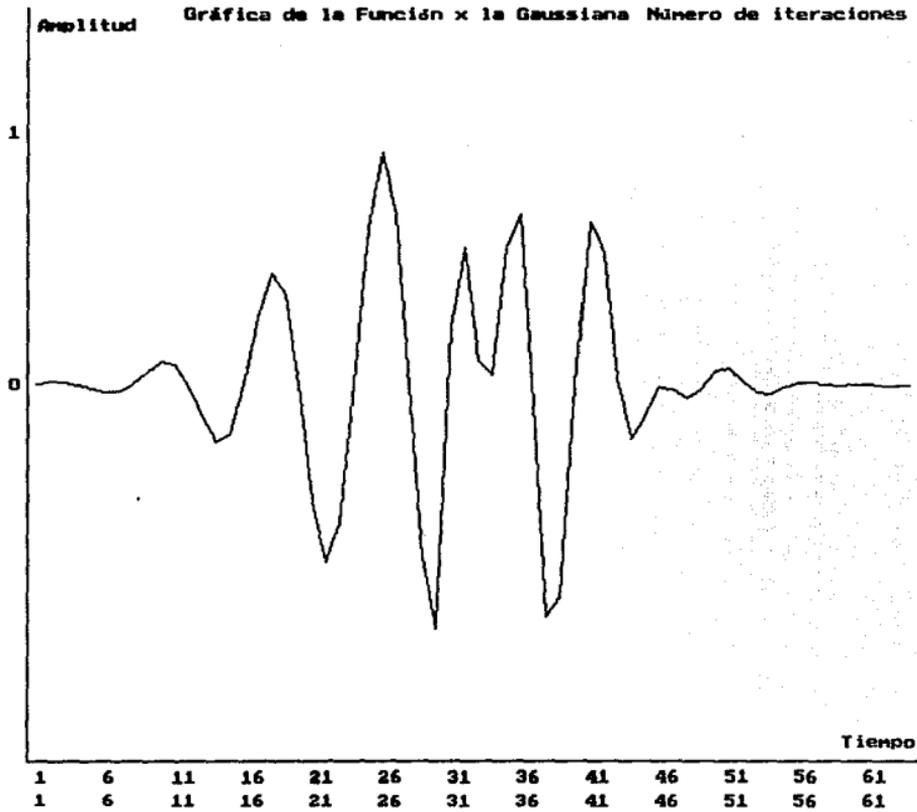
51

56

61



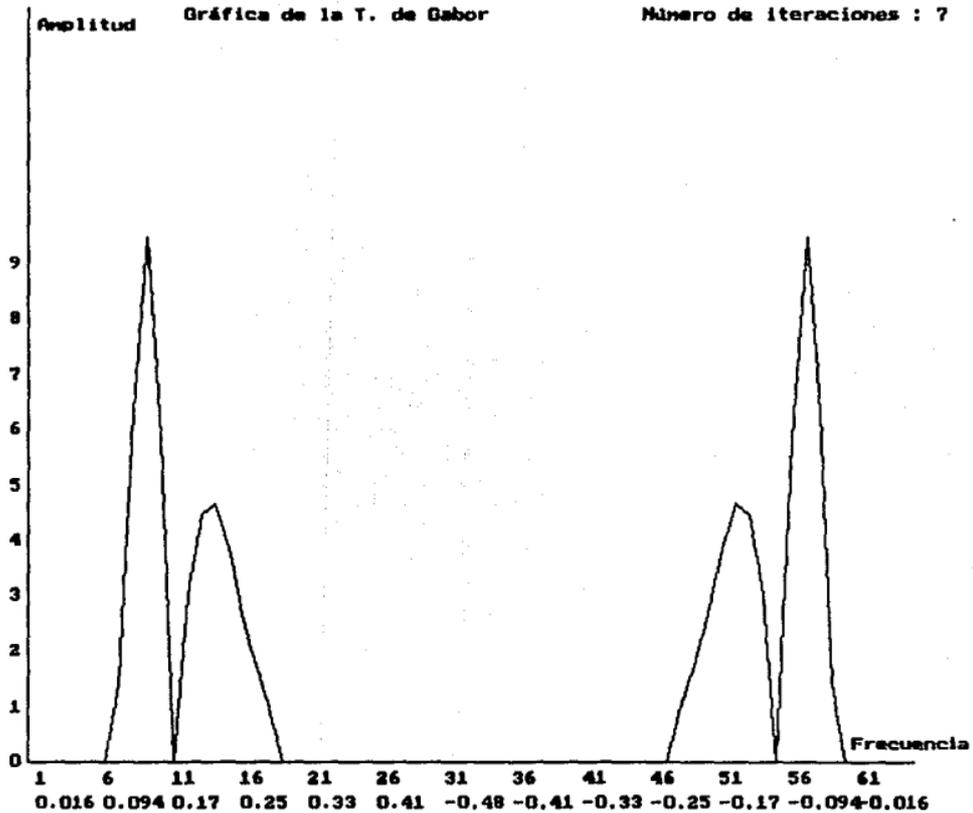
Gráfica de la Función x la Gaussiana Número de iteraciones : 7



Amplitud

Gráfica de la T. de Gabor

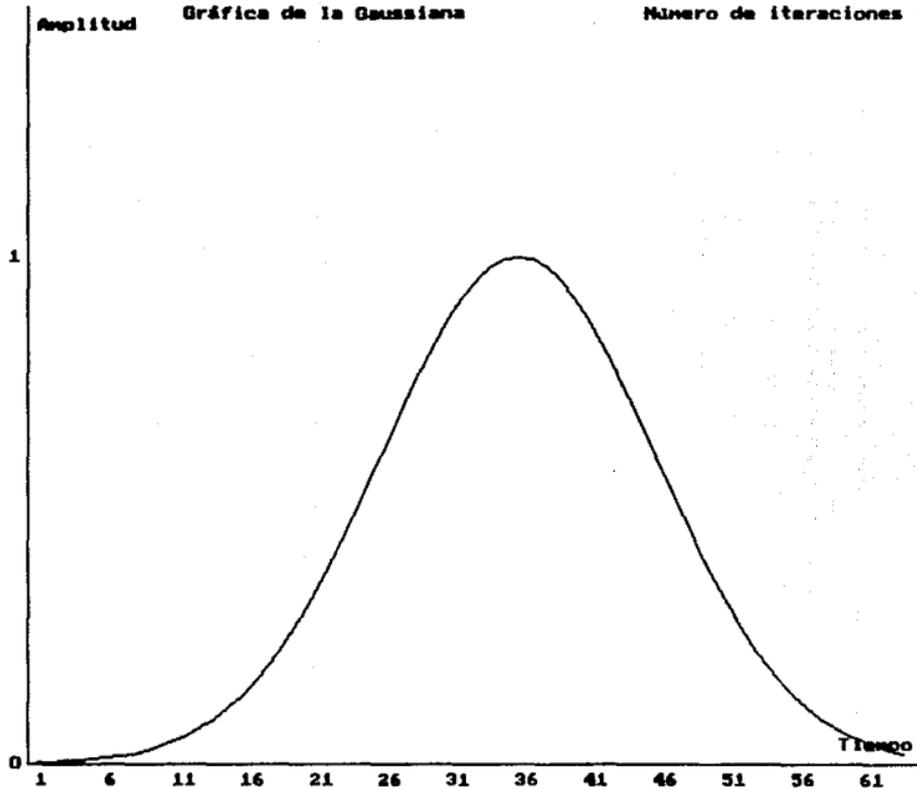
Número de iteraciones : 7



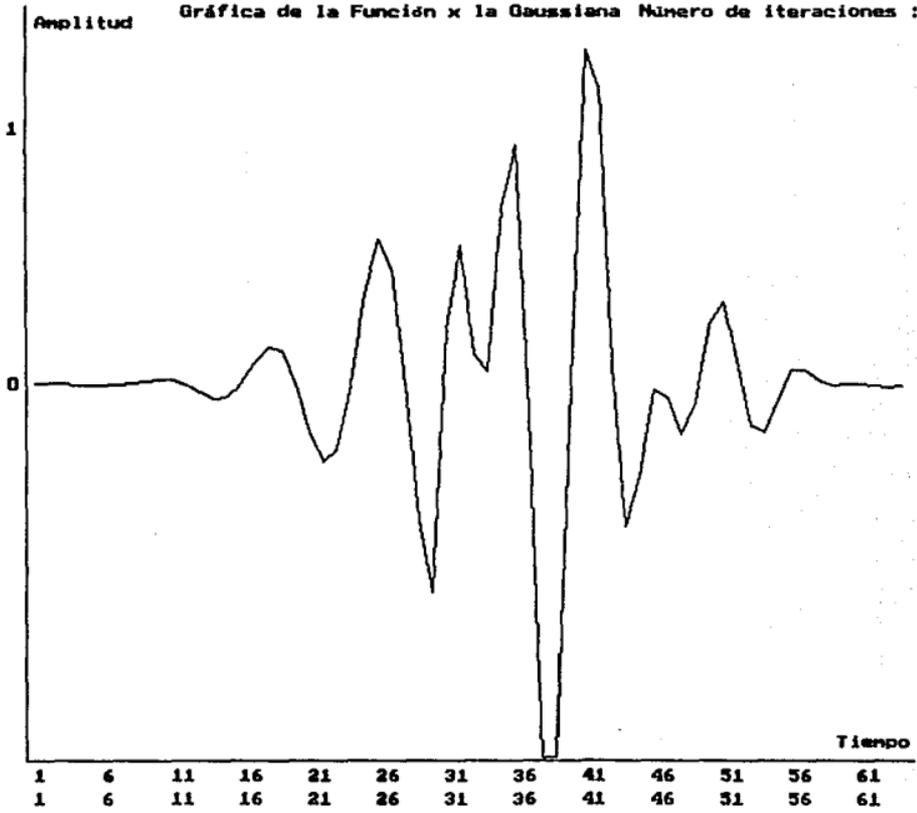
Amplitud

Gráfica de la Gaussiana

Número de iteraciones : 8



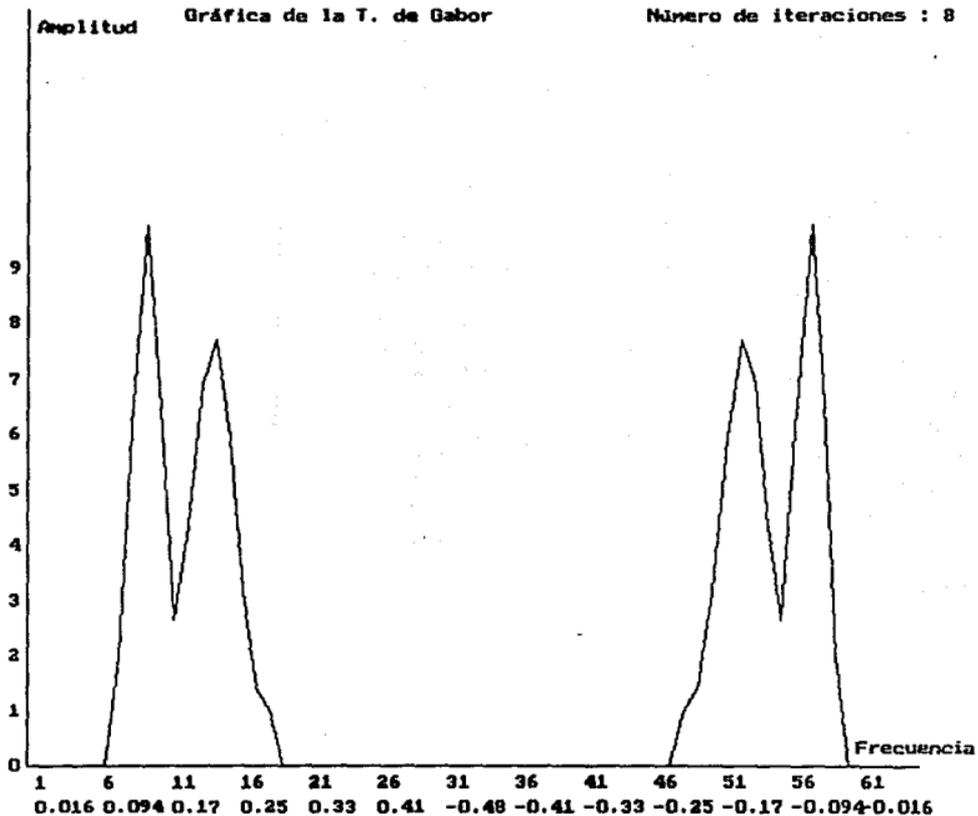
Amplitud Gráfica de la Función x la Gaussiana Número de iteraciones : 8



Amplitud

Gráfica de la T. de Gabor

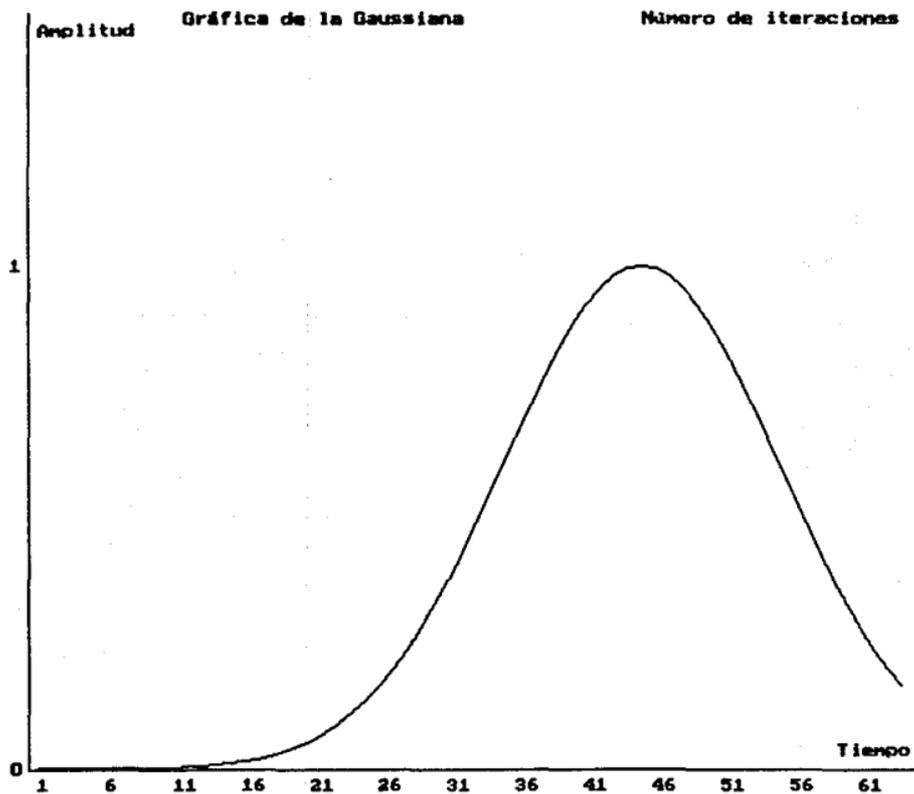
Número de iteraciones : 8



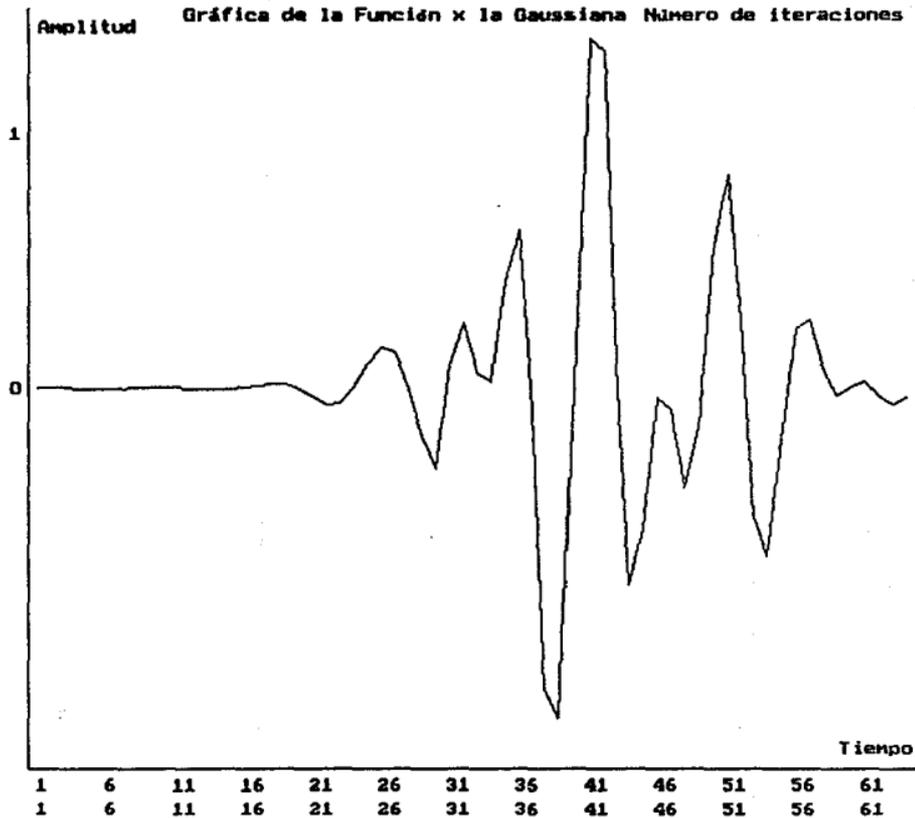
Amplitud

Gráfica de la Gaussiana

Número de iteraciones : 9

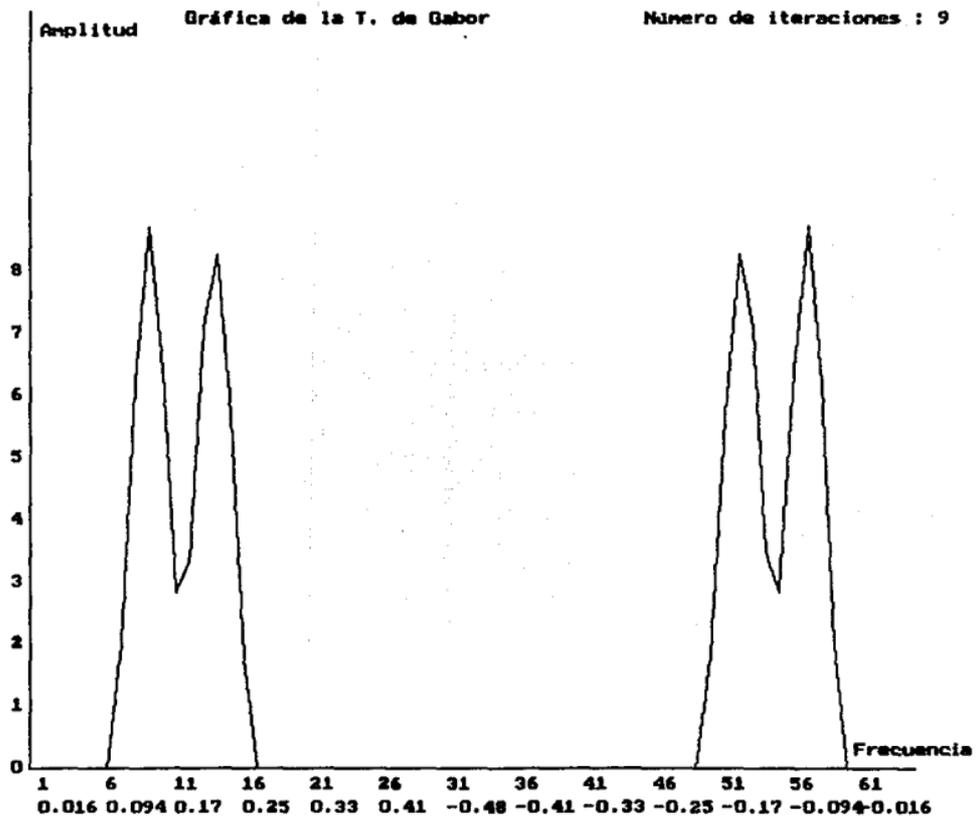


Gráfica de la Función x la Gaussiana Número de iteraciones : 9



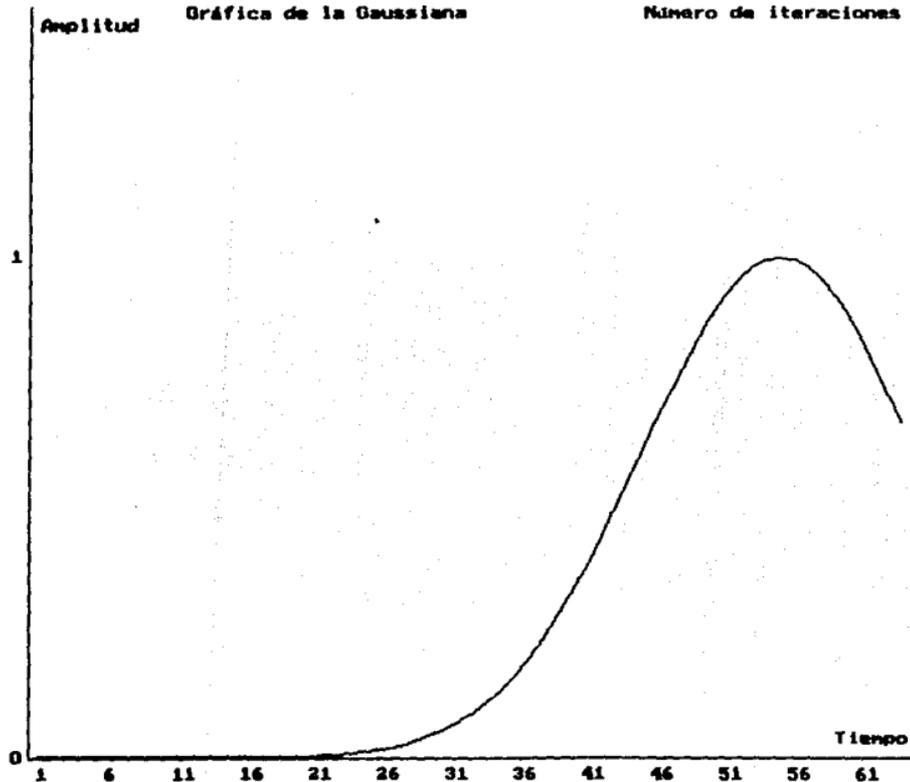
Amplitud **Gráfica de la T. de Gabor**

Número de iteraciones : 9

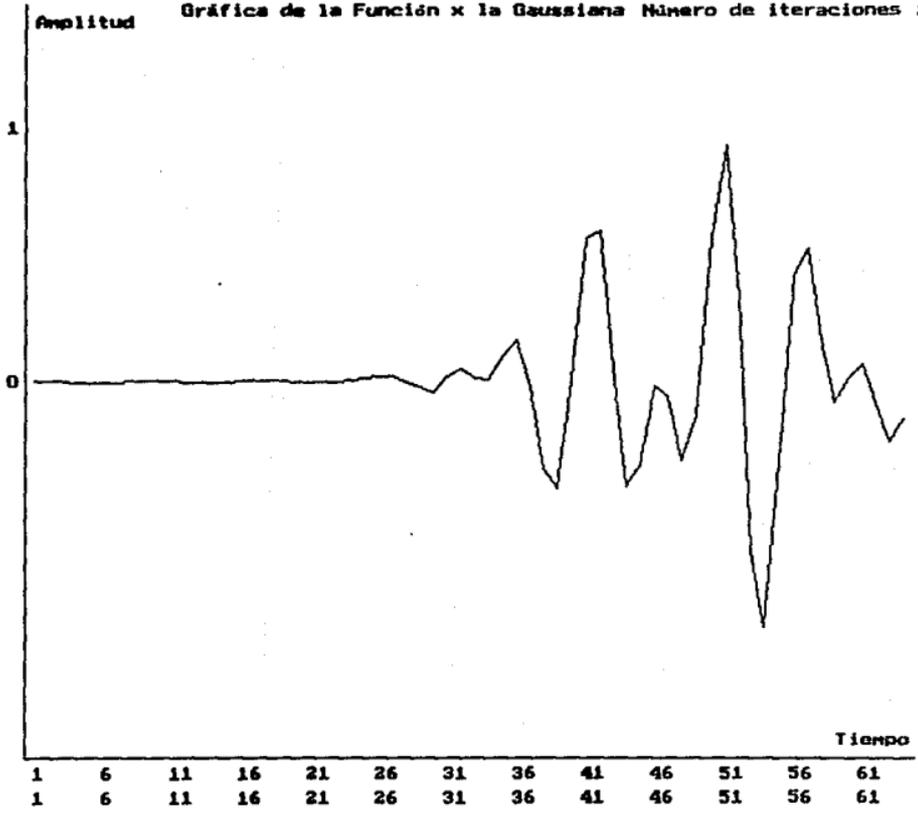


Gráfica de la Gaussiana

Número de iteraciones : 10

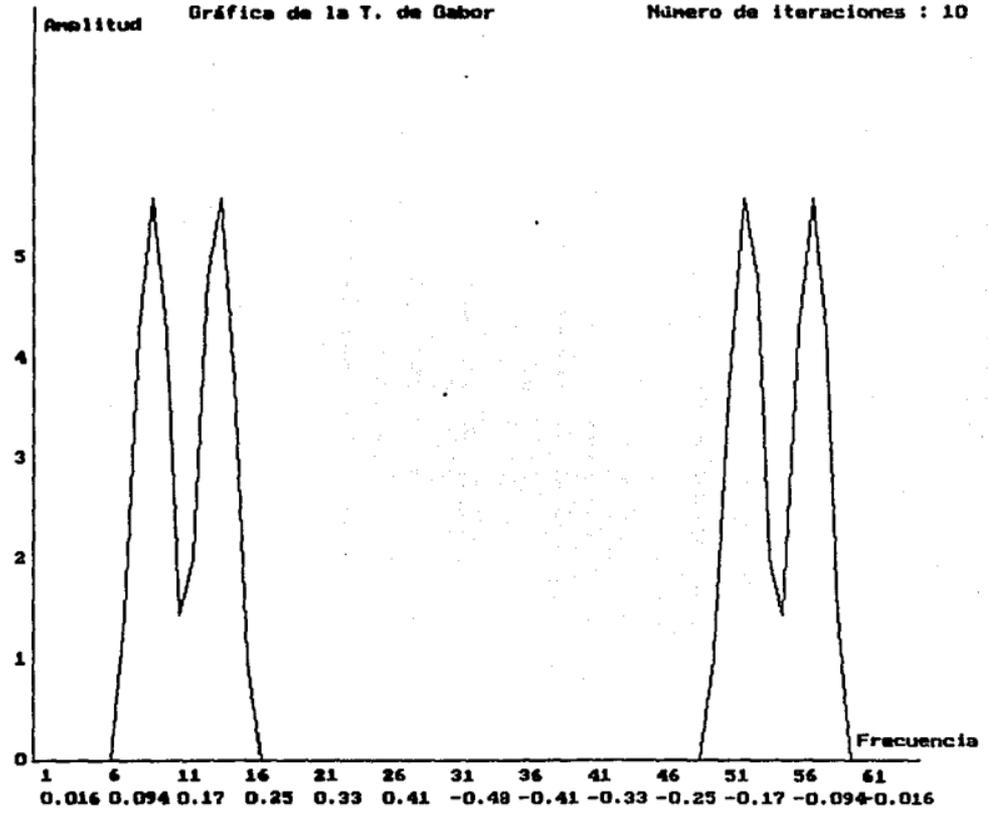


Gráfica de la Función x la Gaussiana Número de iteraciones : 10

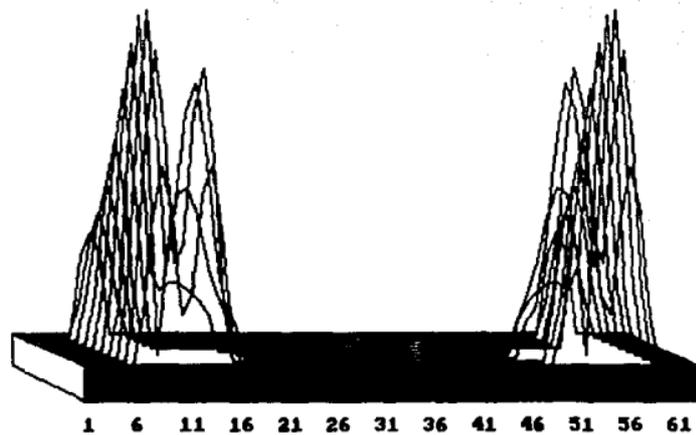


Amplitud Gráfica de la T. de Gabor

Número de iteraciones : 10

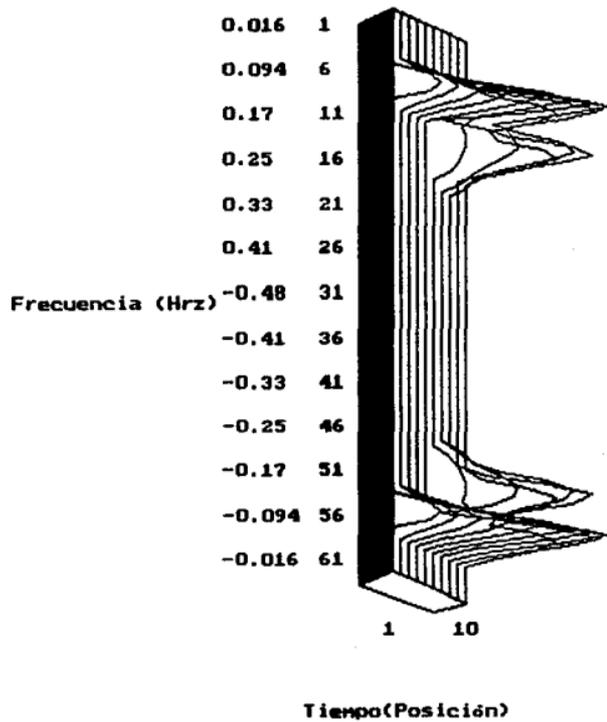


Gráfica de la T. de Gabor en 3-D



Frecuencia

Rotación de la Gráfica en 3-D



## Análisis del ejemplo.

Analicemos el ejemplo; la señal son dos senos, cada seno tienen una envolvente Gaussiana, la primera gráfica es la señal (función), a esta función se le calculó su Transformada de Fourier ( $2^{\text{a}}$  gráfica). La Transformada de Fourier de un seno con una envolvente gaussiana, serían dos Gaussianas. Se calculó la Transformada para ver si concordaban las frecuencias de la Transformada de Fourier con la Transformada de Gabor.

Después de las dos primeras gráficas, se muestra la gráfica de la Gaussiana que se está utilizando como ventana en el cálculo de la Transformada de Gabor. A continuación se muestra la gráfica de la Gaussiana multiplicada por la señal (función) en el tiempo 1. A esta multiplicación se le calcula la Transformada de Fourier, que es la Transformada de Gabor en el tiempo 1. Se calcula el vector de desplazamiento y con este vector se calcula la Gaussiana en el tiempo 2, se multiplica por la señal y se calculó su Transformada de Fourier. Este cálculo se hará 10 veces.

Es importante recalcar que en la iteración 5 se localiza el otro seno de la señal, pero no se refleja aún en la Transformada de Gabor, es en la siguiente iteración cuando se refleja el segundo seno. En ese momento se están localizando las frecuencias de los dos senos en el tiempo 6. Se sigue con las iteraciones y se localizan las frecuencias de ambos senos en el tiempo de la iteración.

Se grafica el resultado de todas las Transformadas de Gabor en tercera dimensión. En esta gráfica se pueden ver todas las Transformadas de Gabor que se calcularon.

La siguiente gráfica es en tercera dimensión pero rotada, para poder observar la frecuencia y tiempo. En esta gráfica se ve como se localizó la señal en tiempo y frecuencia.

#### 2.4 EJEMPLO 2

Se analizará una señal de 2 senos, las frecuencias que se utilizarán para calcularla serán con  $f = 2^{n/12}$ , cada seno tendrá una envolvente.

Los datos son :

Número de puntos de la muestra N : 64  
Paso de muestreo  $\Delta t$  : 1  
Introduzca el valor de 1 - 12 : 5.5  
Introduzca el valor de 1 - 12 : 4.5  
Dilatación de la Gaussiana a : 10  
Número de iteraciones en el tiempo n : 10

Ejecución del programa...

## TRANSFORMADA DE GABOR



**1: Lee los datos de un archivo**



**2: Lee los datos desde el teclado**



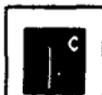
**3: Ejecuta un ejemplo**



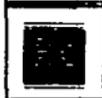
**4: Salir al D.O.S.**

**Introduzca la opción :**

## EJEMPLOS DE TRANSFORMADA DE GABOR



1:  $\text{Cos}(2\pi f t) + \text{Cos}(2\pi f t)$



2:  $\text{Sen}(2\pi f t) + \text{Sen}(2\pi f t)$

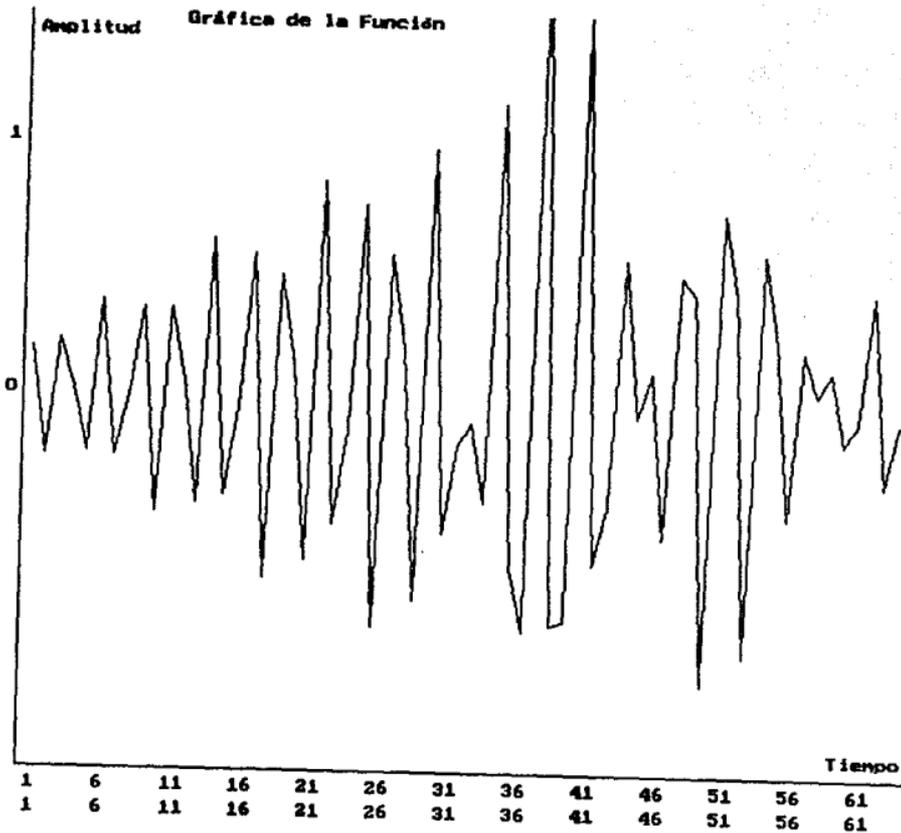


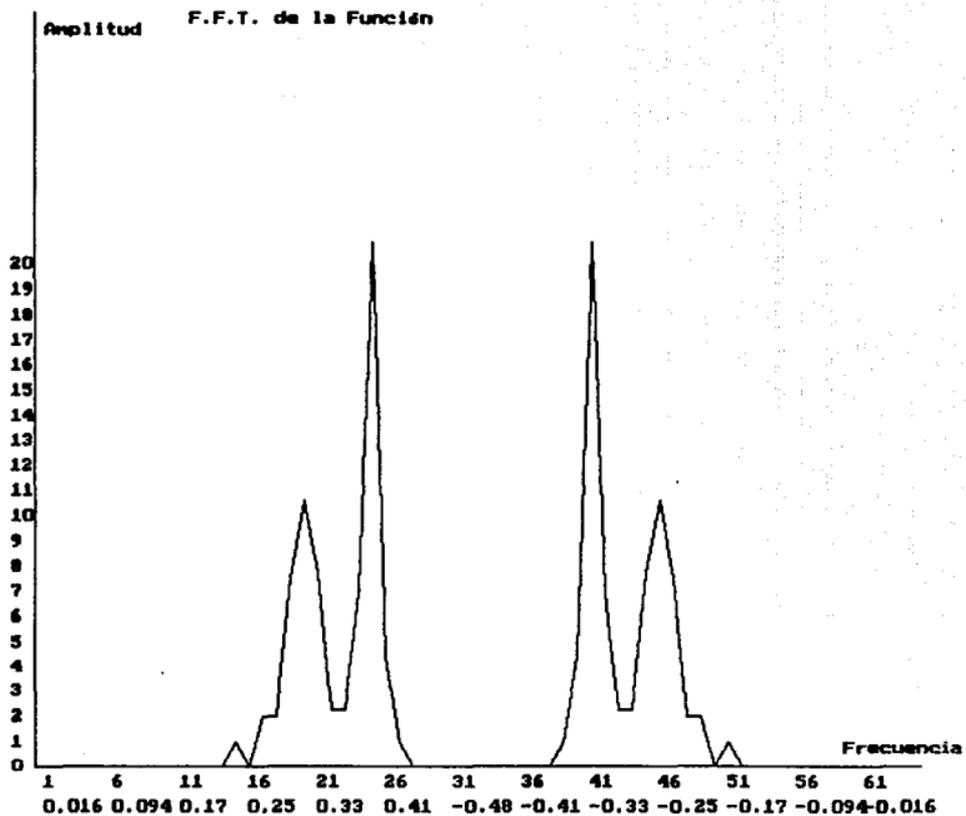
3:  $\text{Sen}(2\pi f t) + \text{Sen}(2\pi f t)$



4: Ejemplo de Frecuencia de Musica

Introduzca la opción :





Amplitud

Gráfica de la Gaussiana

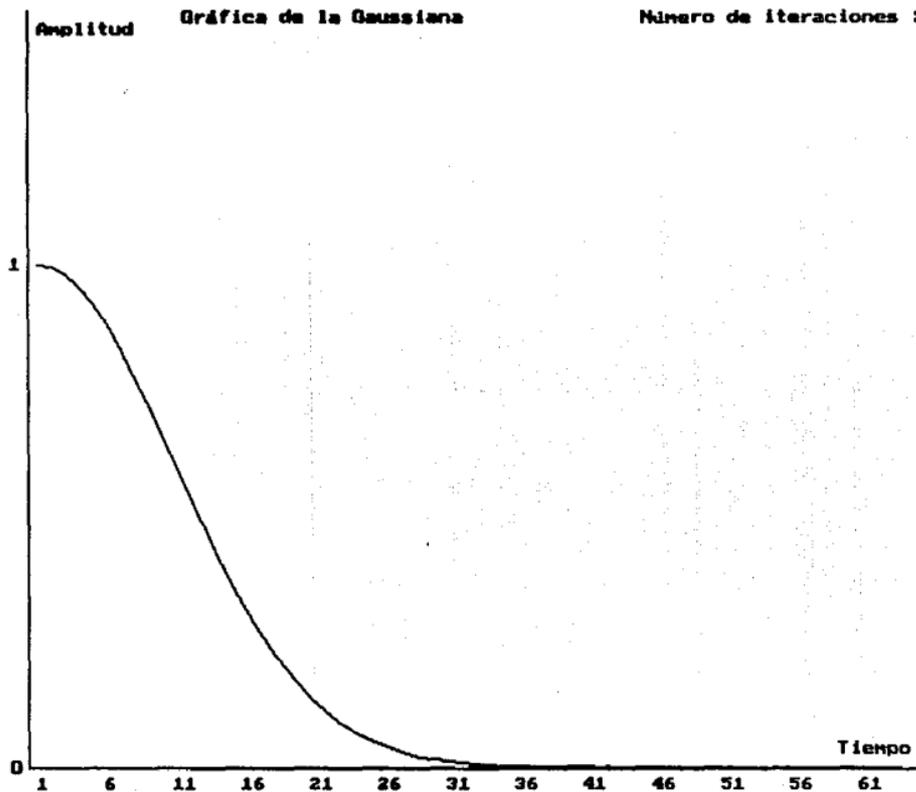
Número de iteraciones : 1

1

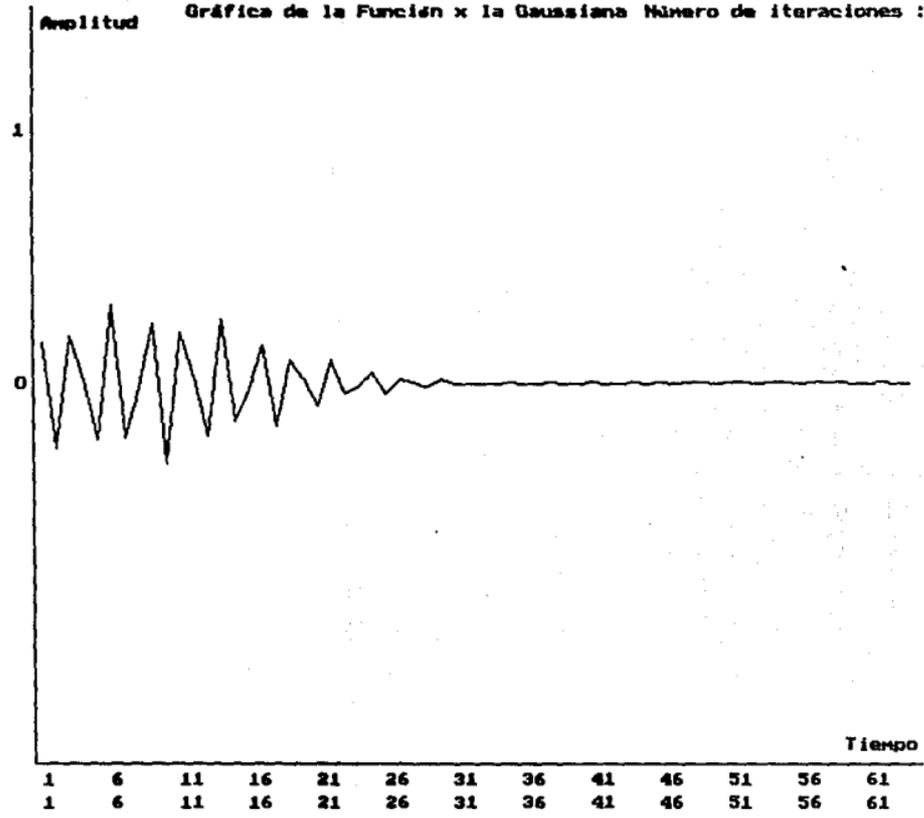
0

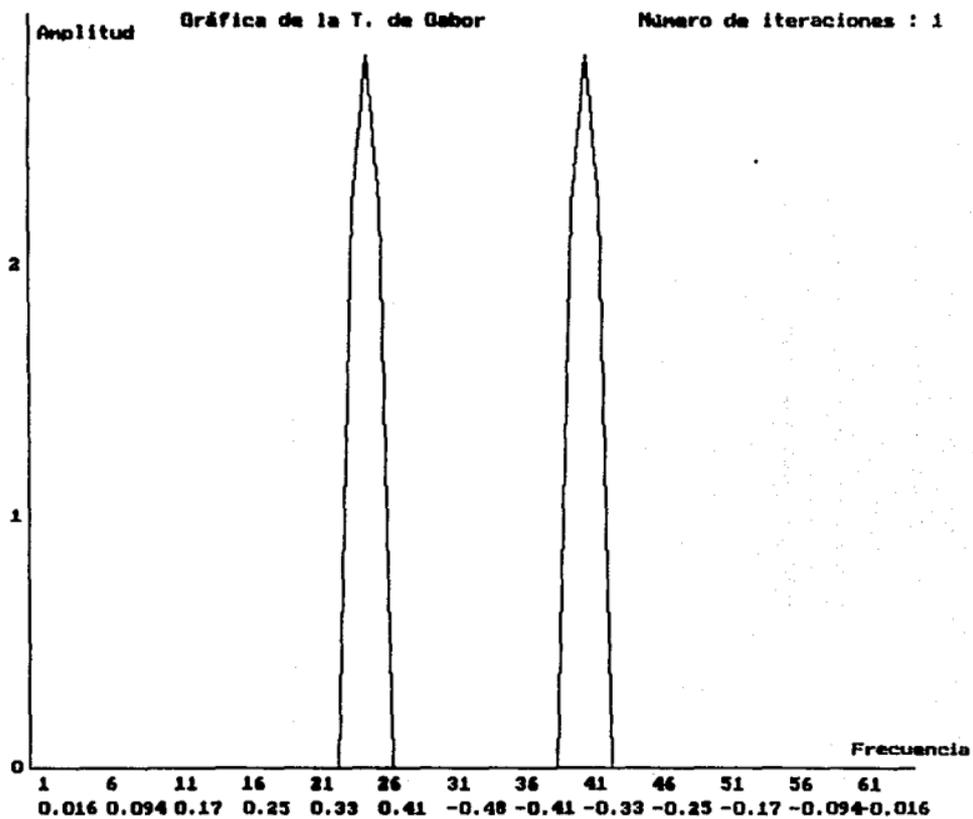
1 6 11 16 21 26 31 36 41 46 51 56 61

Tiempo



Gráfica de la Función x la Gaussiana Número de iteraciones : 1

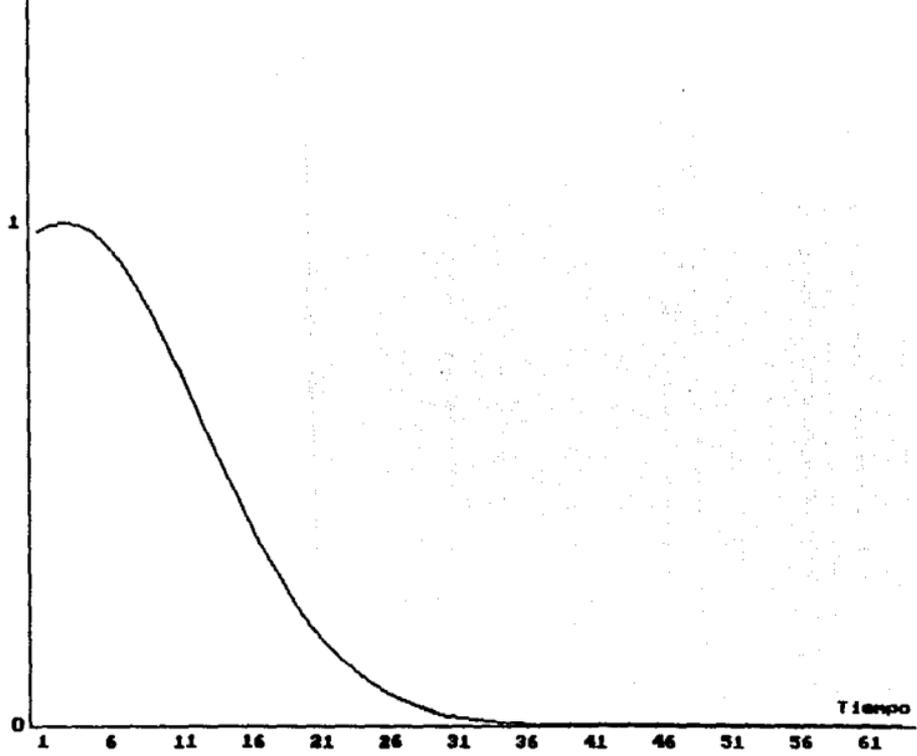




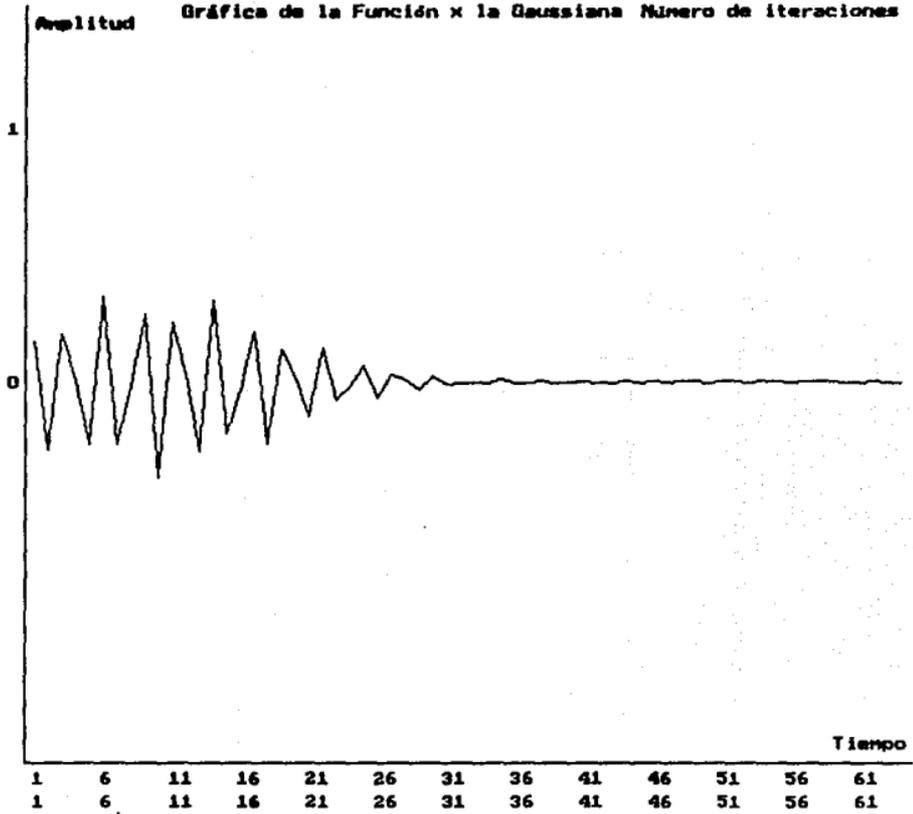
Amplitud

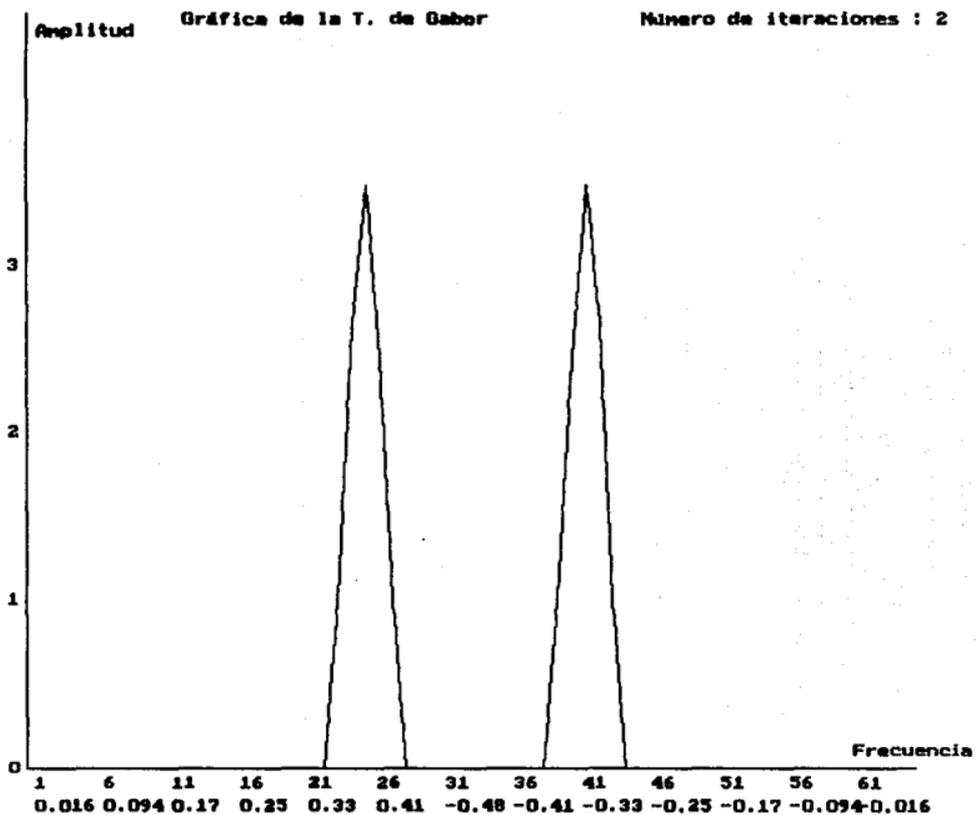
Gráfica de la Gaussiana

Número de iteraciones : 2



Gráfica de la Función x la Gaussiana Número de iteraciones : 2

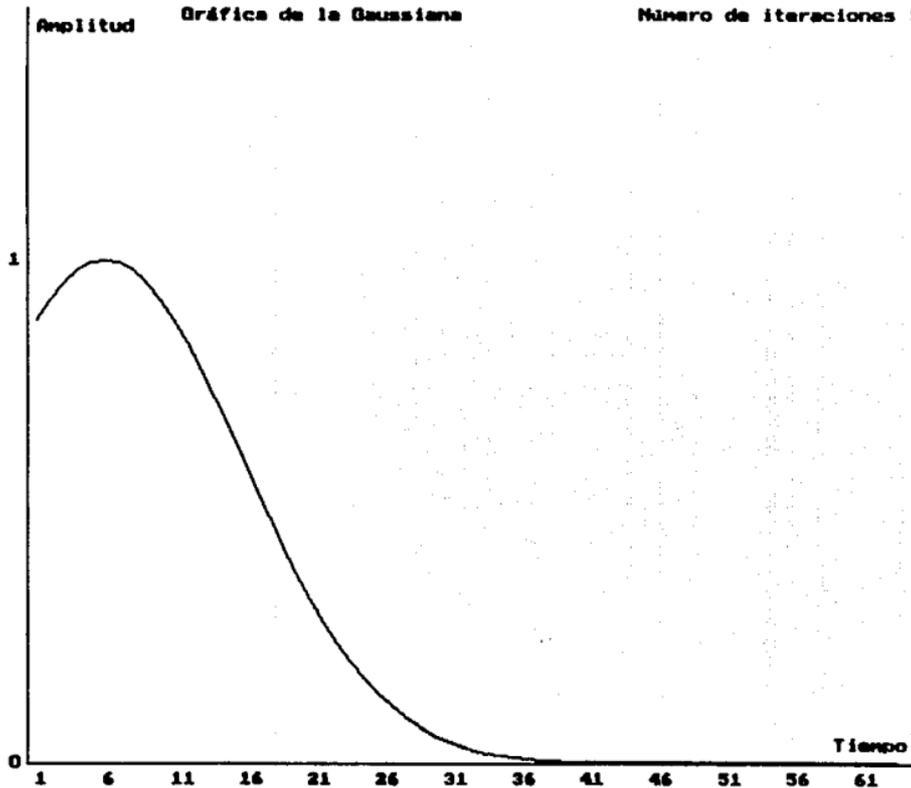




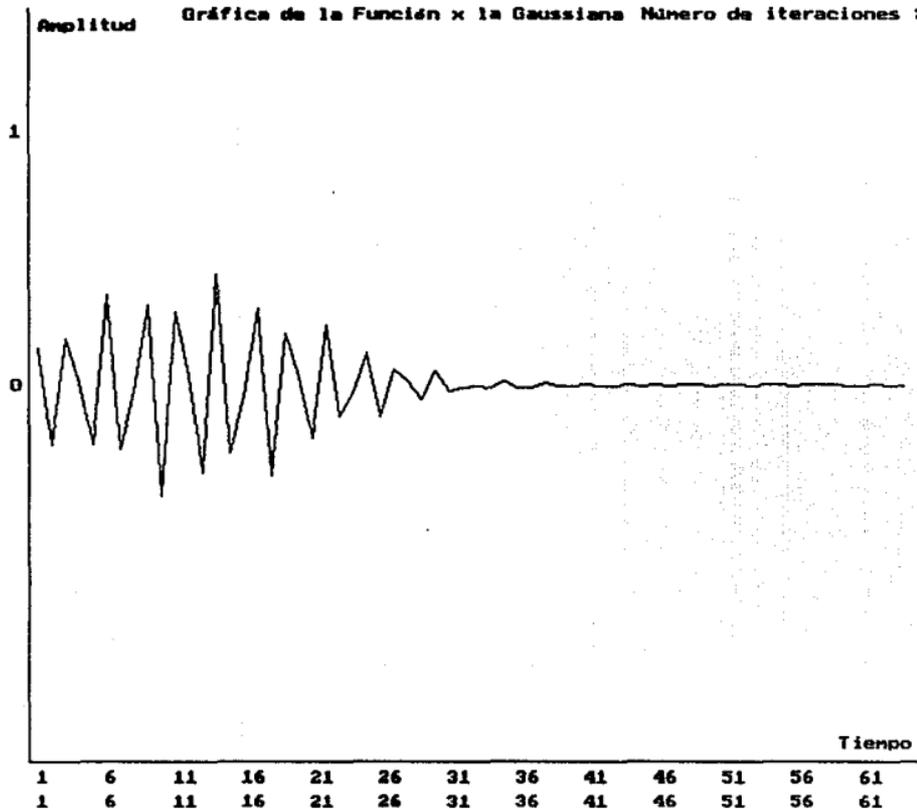
Amplitud

Gráfica de la Gaussiana

Número de iteraciones : 3



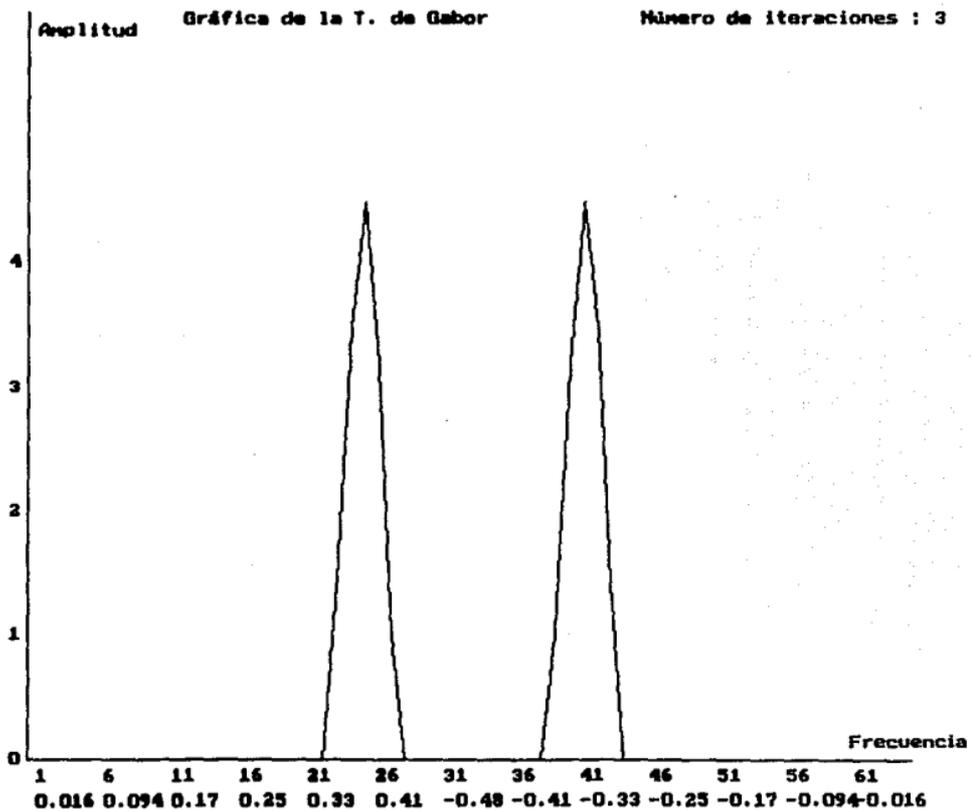
Amplitud Gráfica de la Función x la Gaussiana Número de iteraciones : 3



Amplitud

Gráfica de la T. de Gabor

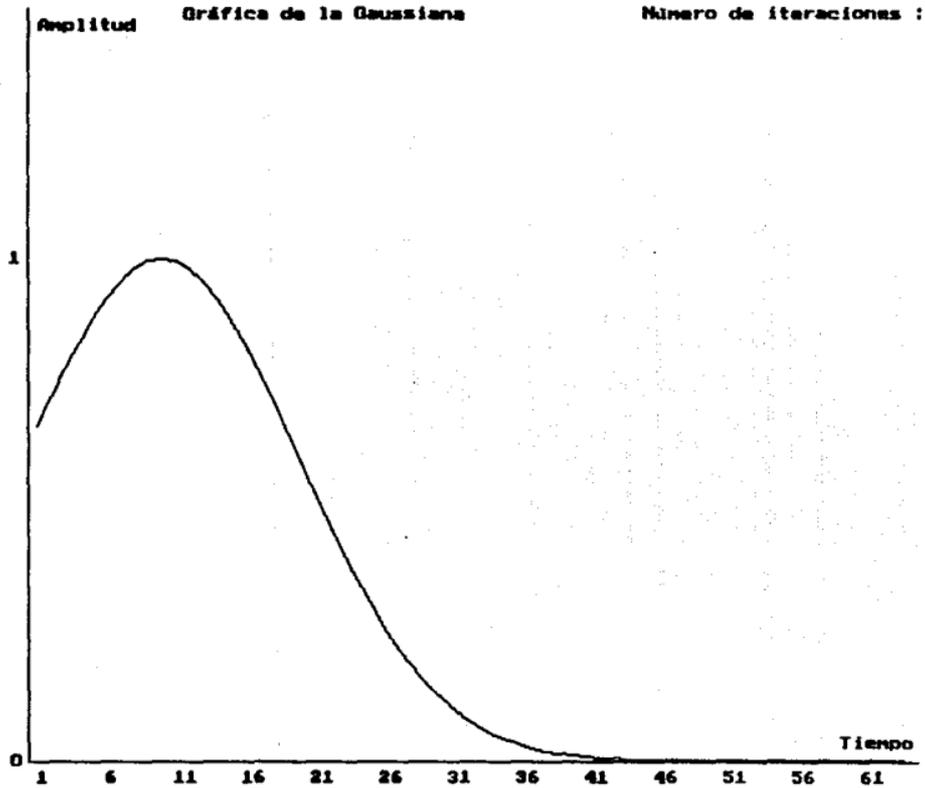
Número de iteraciones : 3

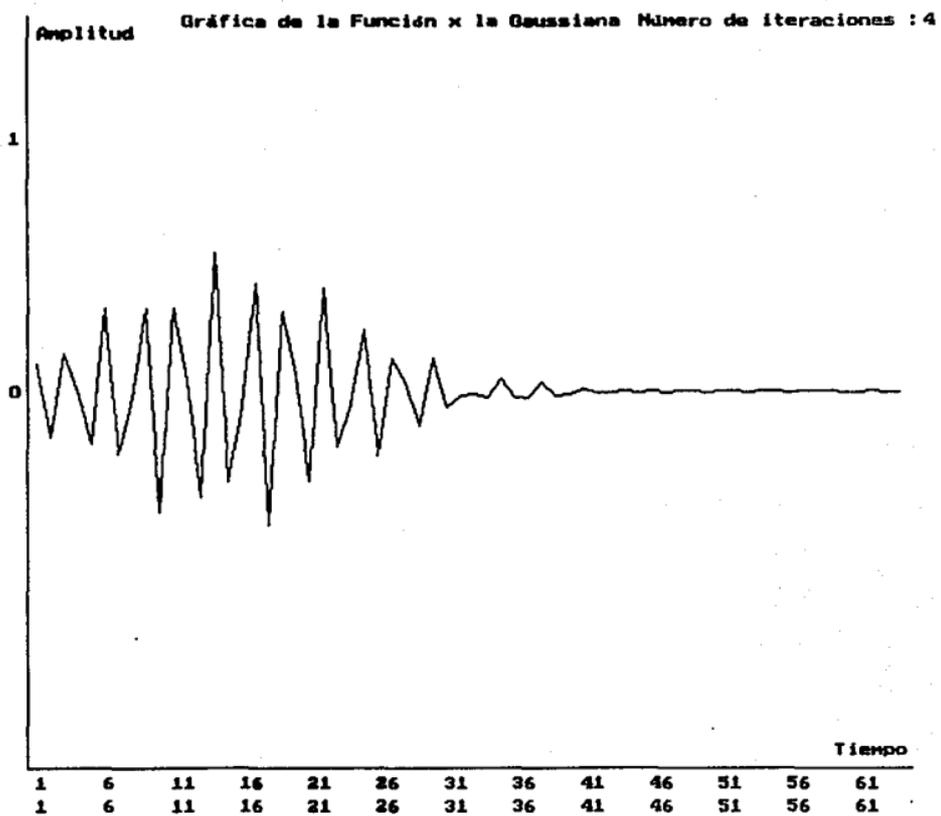


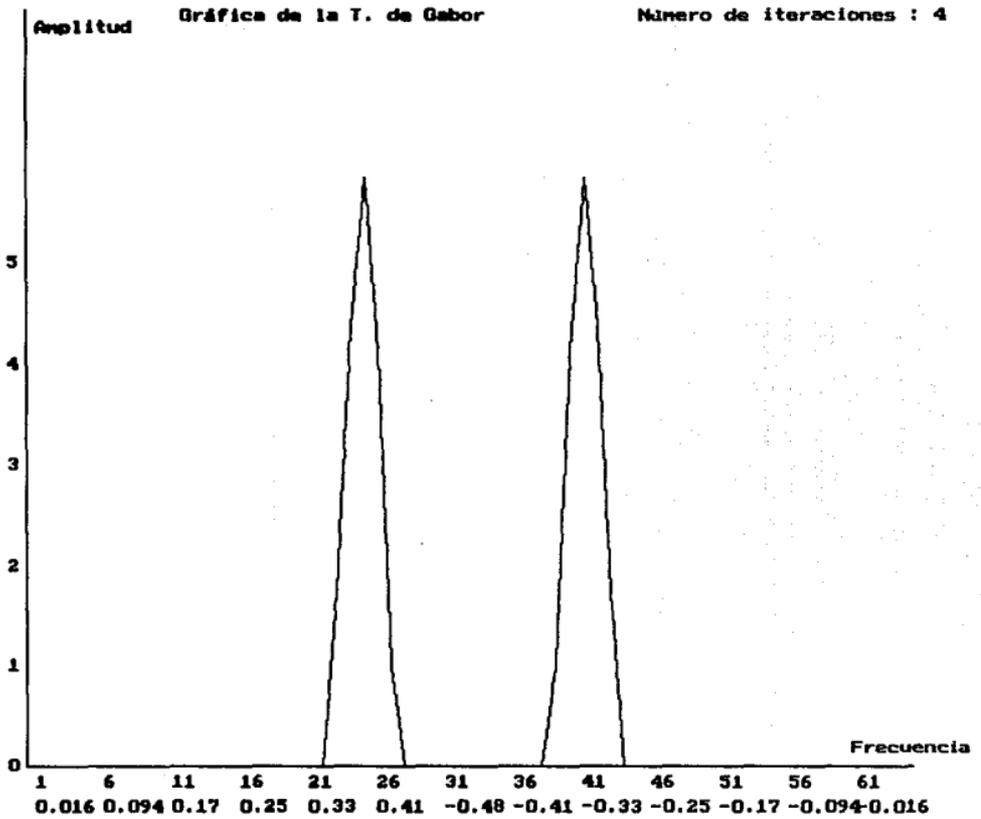
Amplitud

Gráfica de la Gaussiana

Numero de iteraciones : 4



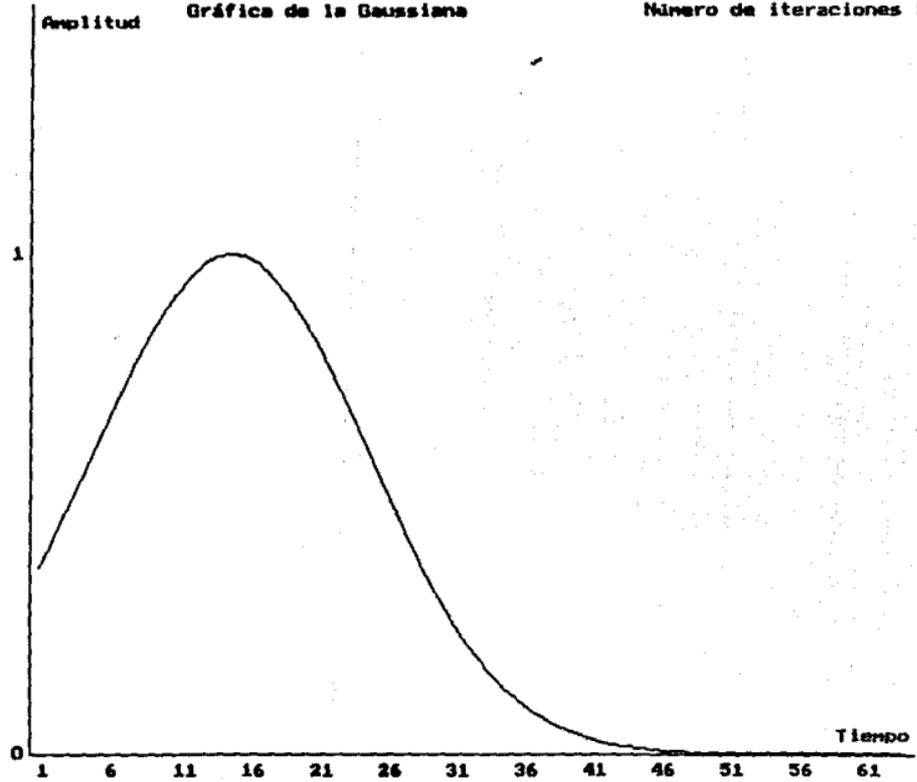


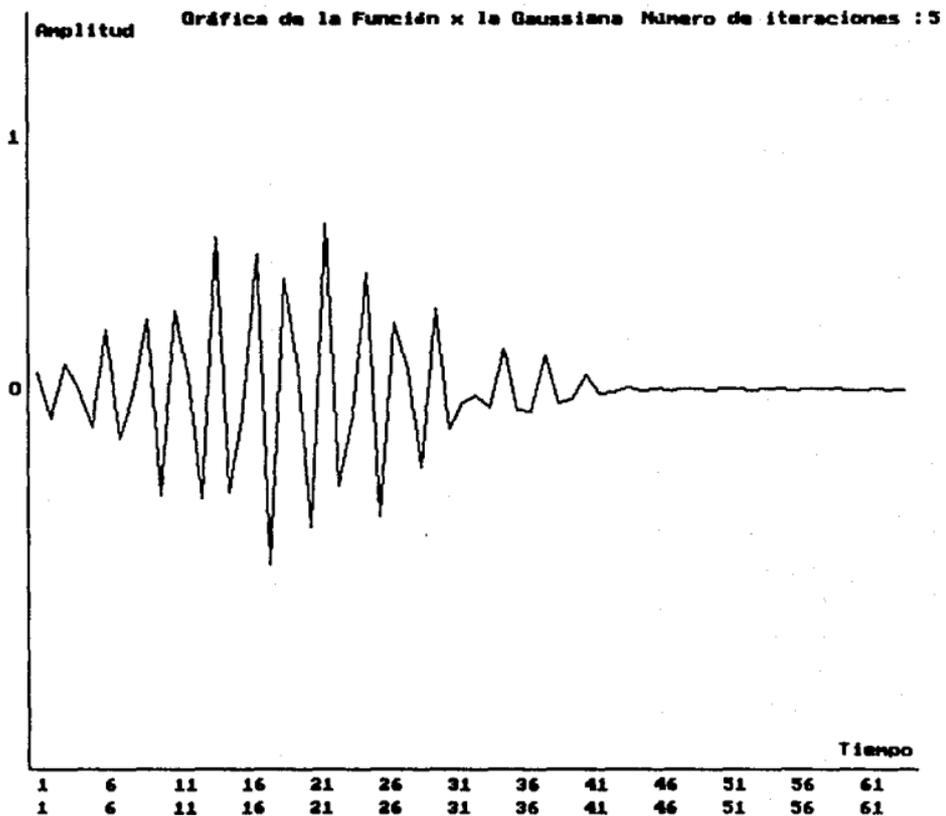


Amplitud

Gráfica de la Gaussiana

Número de iteraciones : 5

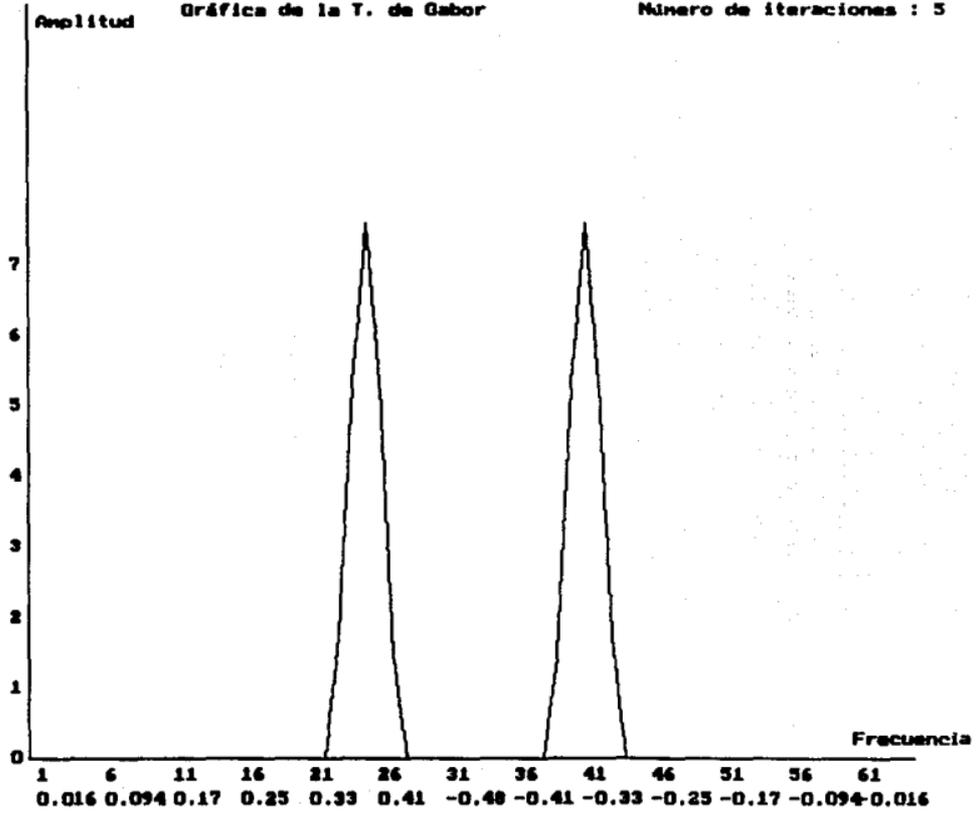




Amplitud

Gráfica de la T. de Gabor

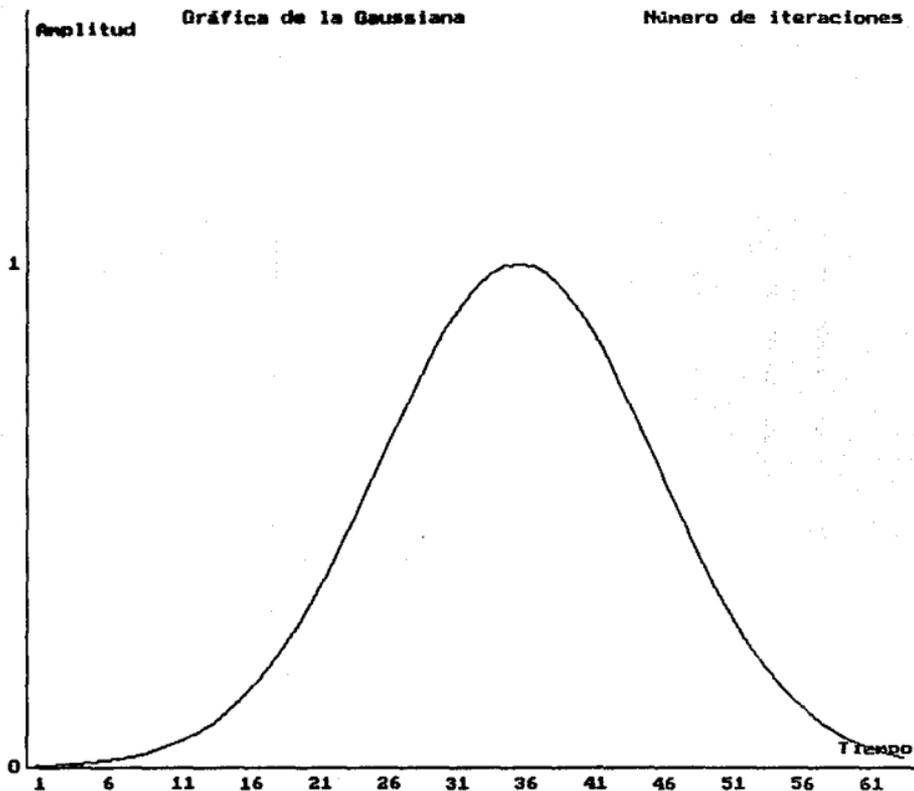
Número de iteraciones : 5



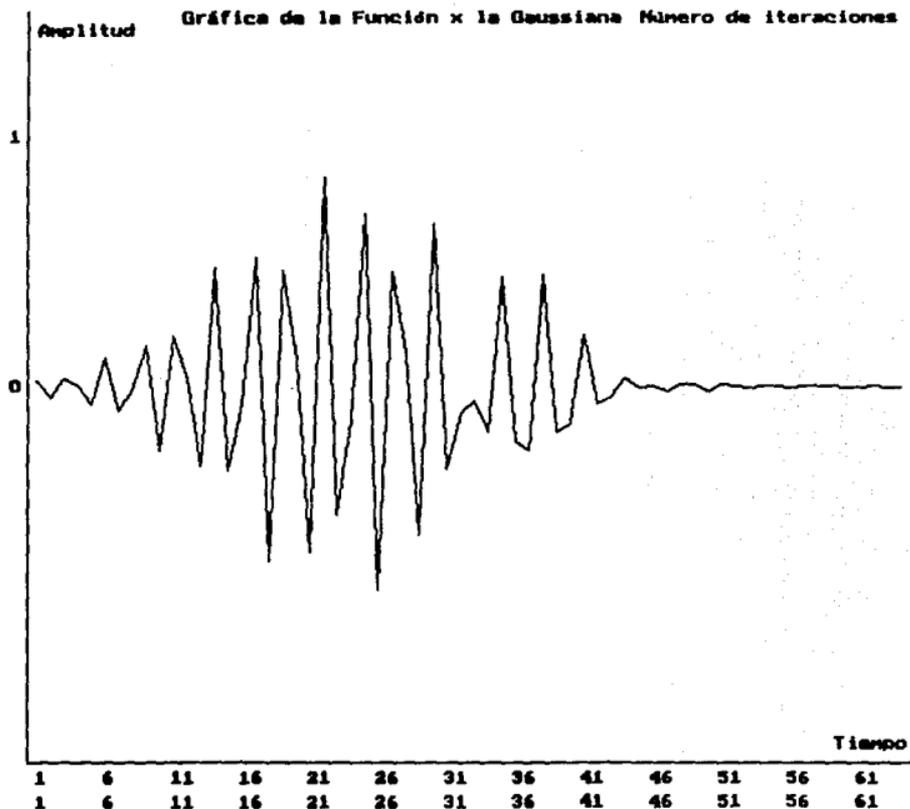
Amplitud

Gráfica de la Gaussiana

Número de iteraciones : 8



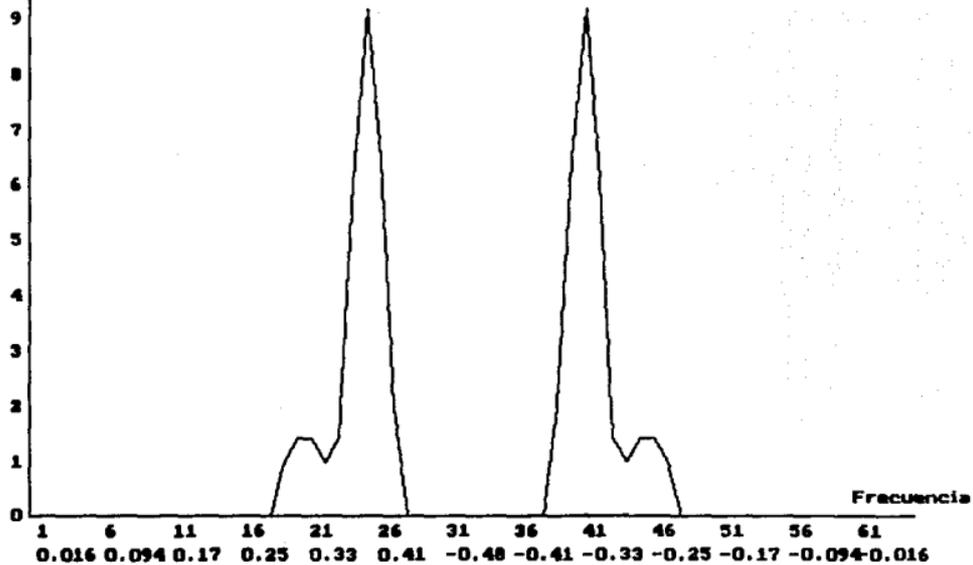
Gráfica de la Función x la Gaussiana Número de iteraciones : 6



Amplitud

Gráfica de la T. de Gabor

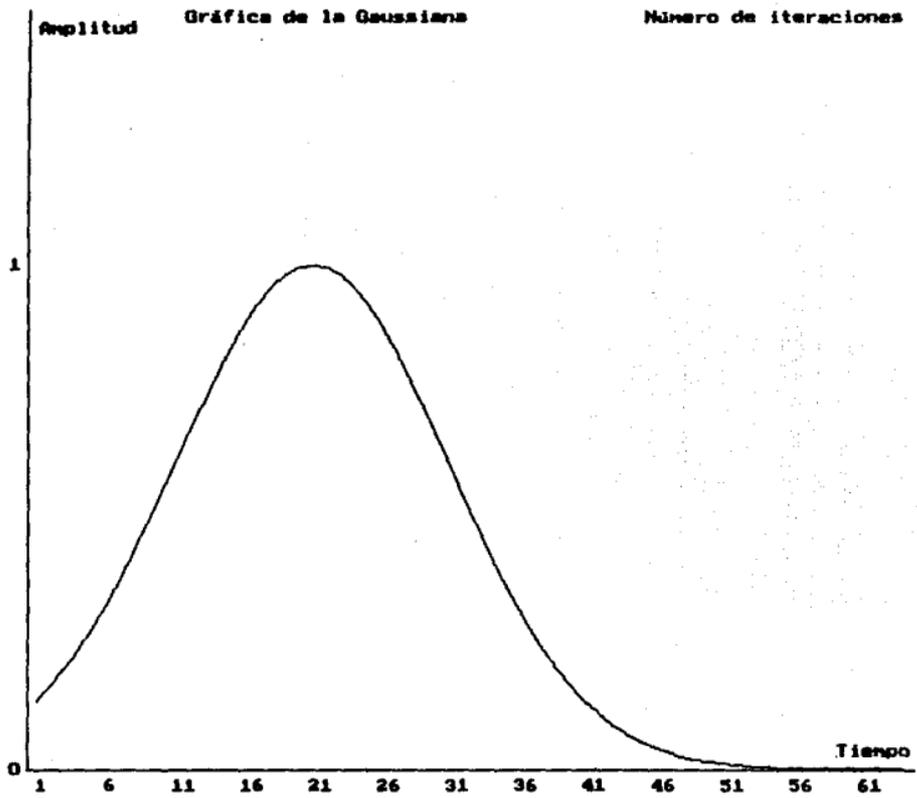
Número de iteraciones : 6



Amplitud

Gráfica de la Gaussiana

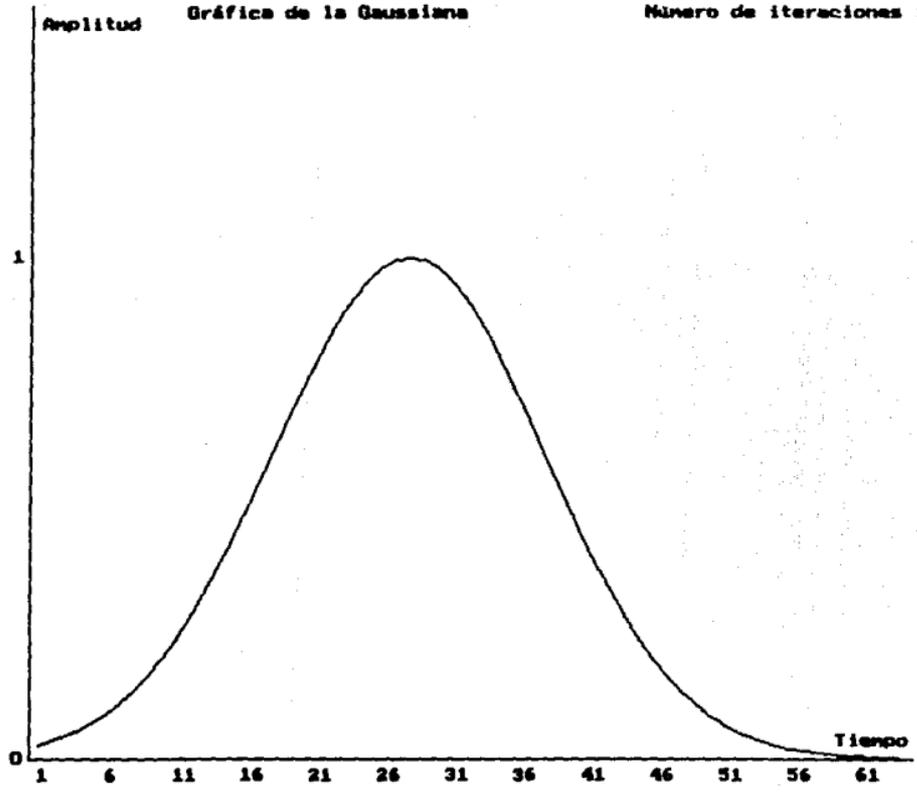
Número de iteraciones : 6



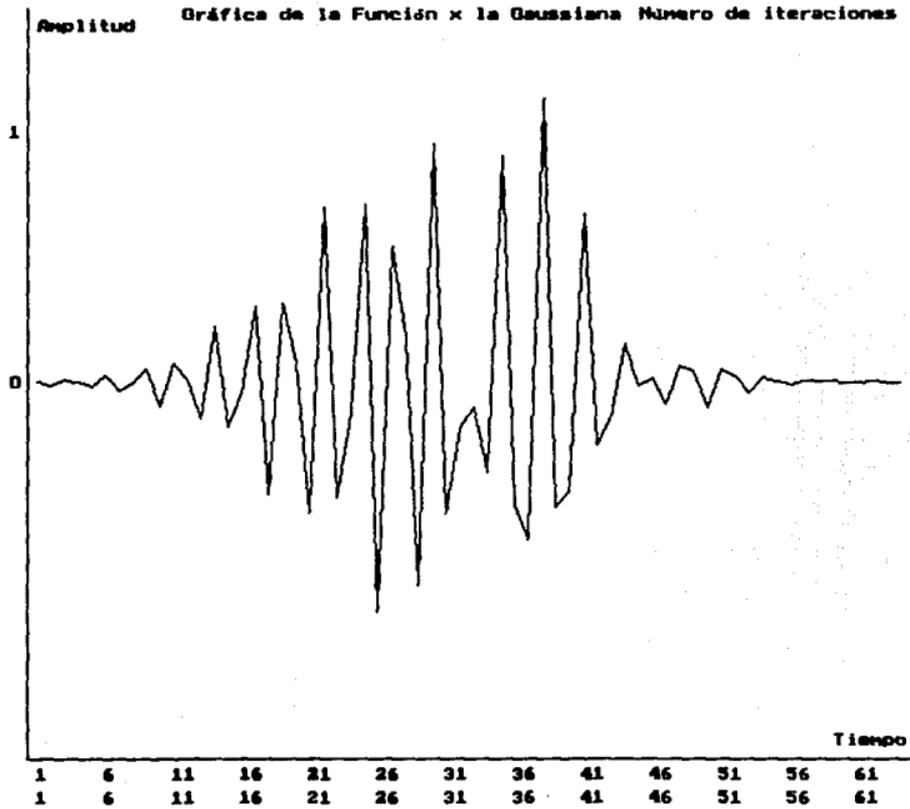
Amplitud

Gráfica de la Gaussiana

Número de iteraciones : ?



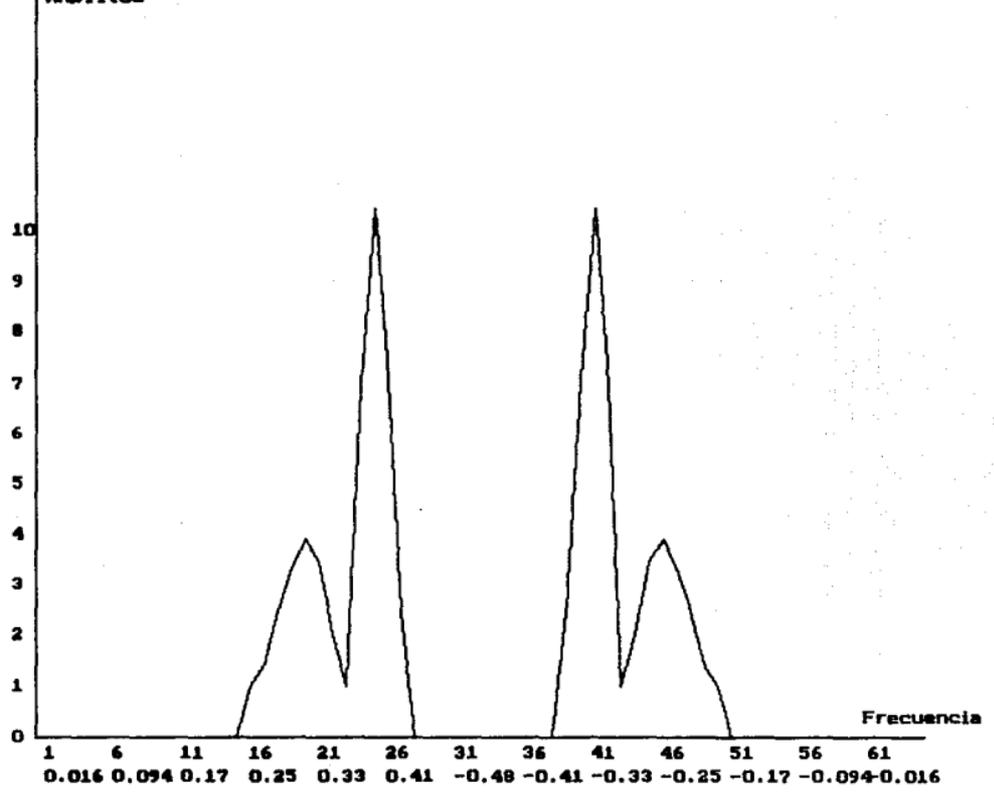
Gráfica de la Función x la Gaussiana Número de iteraciones : 7



Amplitud

Gráfica de la T. de Gabor

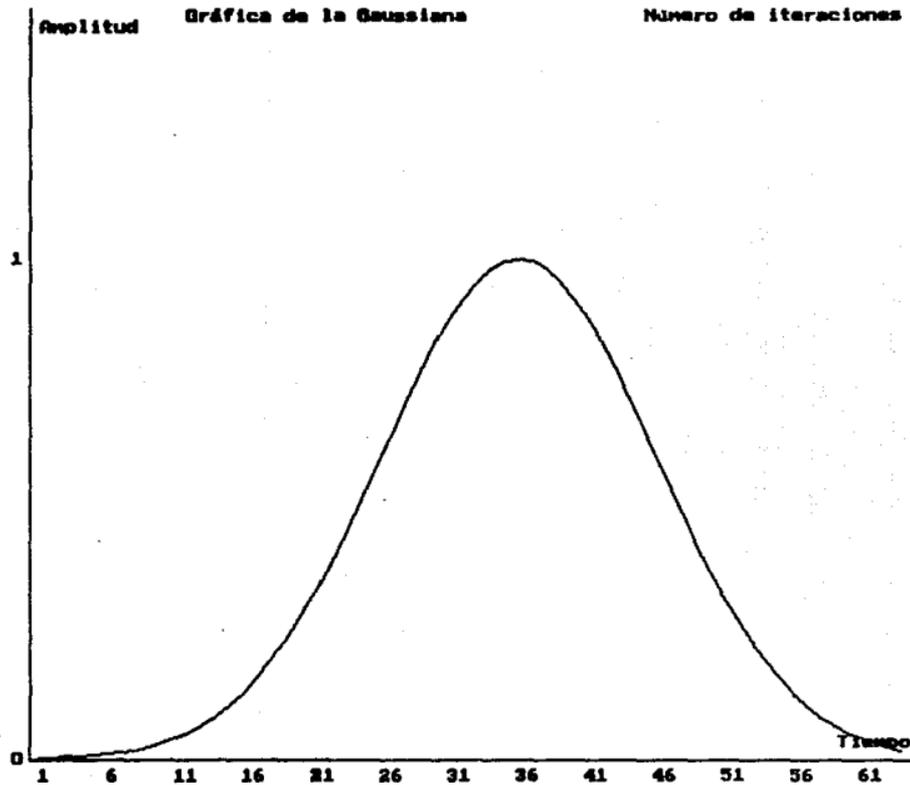
Número de iteraciones : 7

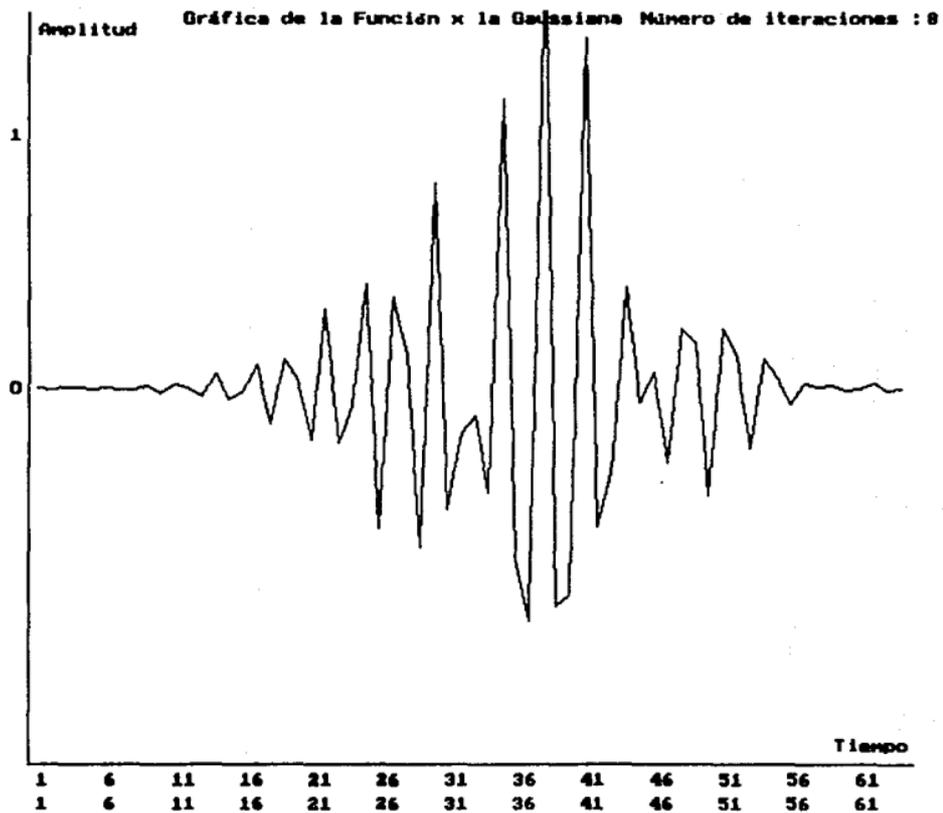


Amplitud

Gráfica de la Gaussiana

Número de iteraciones : 8

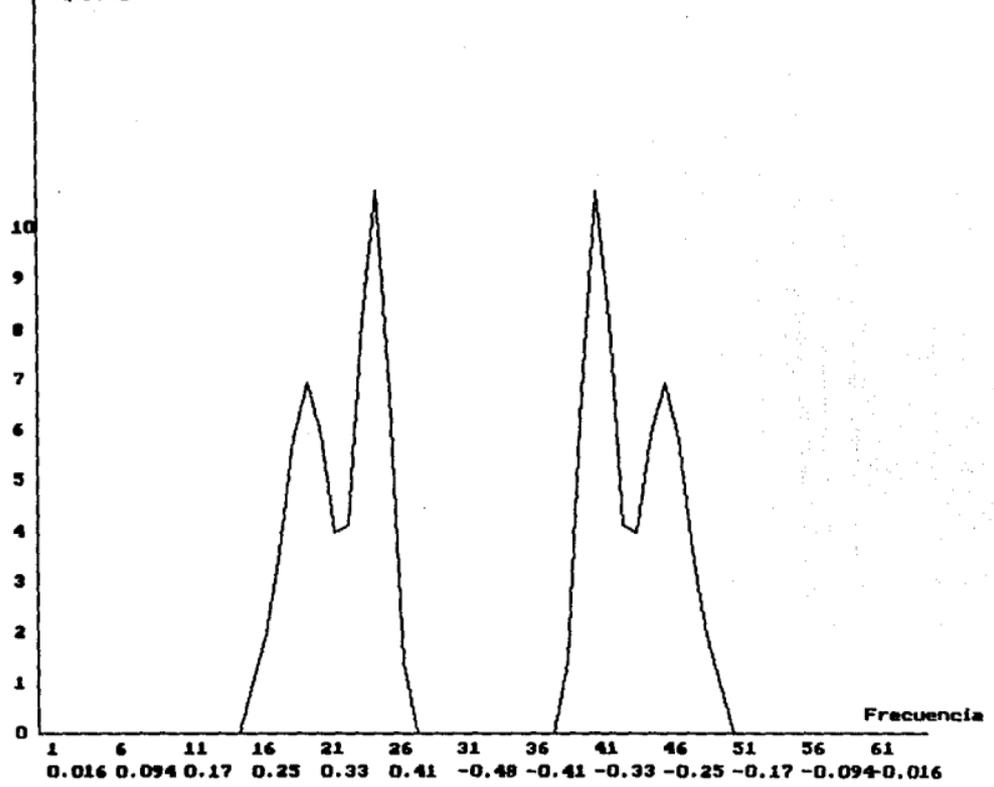




Amplitud

Gráfica de la T. de Gabor

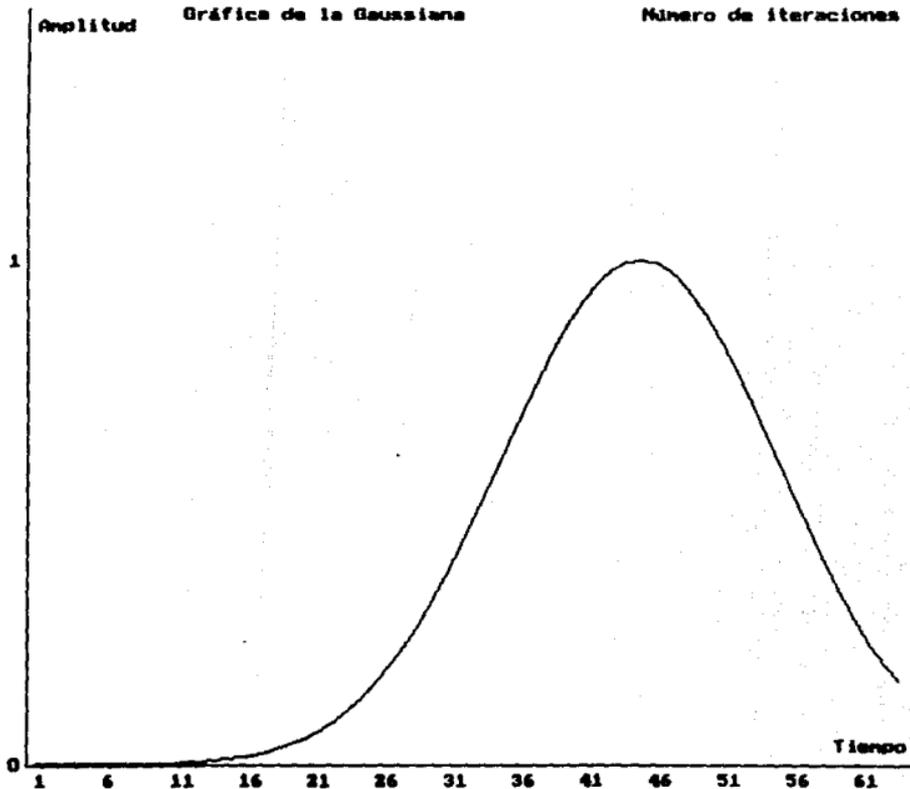
Número de iteraciones : 8

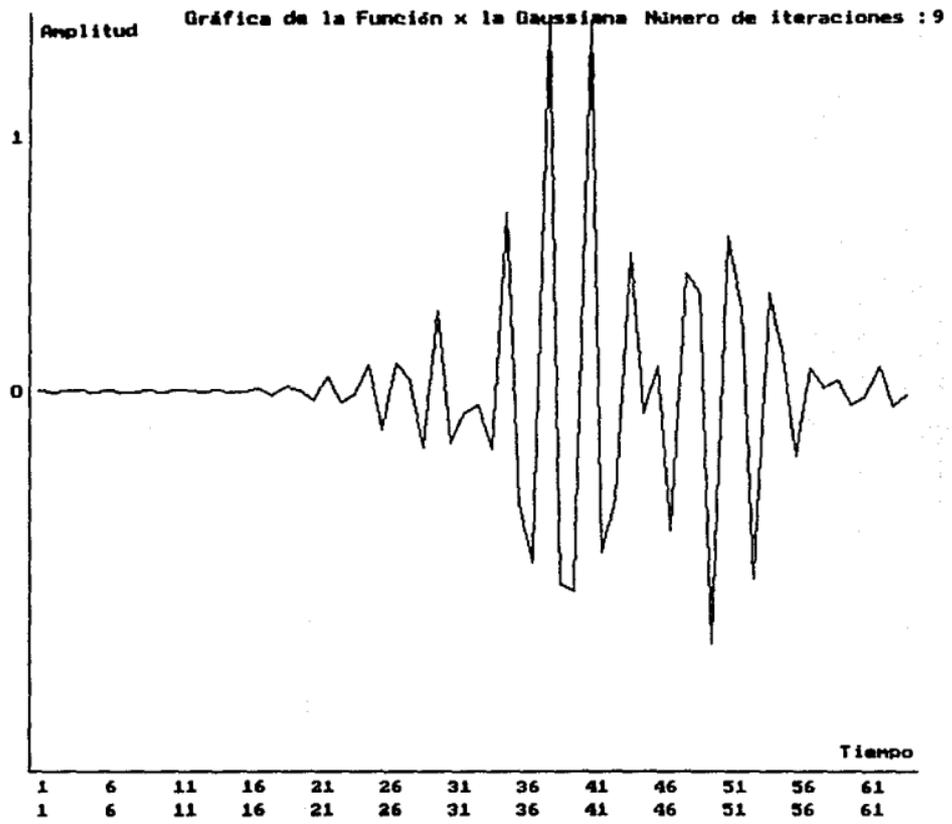


Amplitud

Gráfica de la Gaussiana

Número de iteraciones : 9

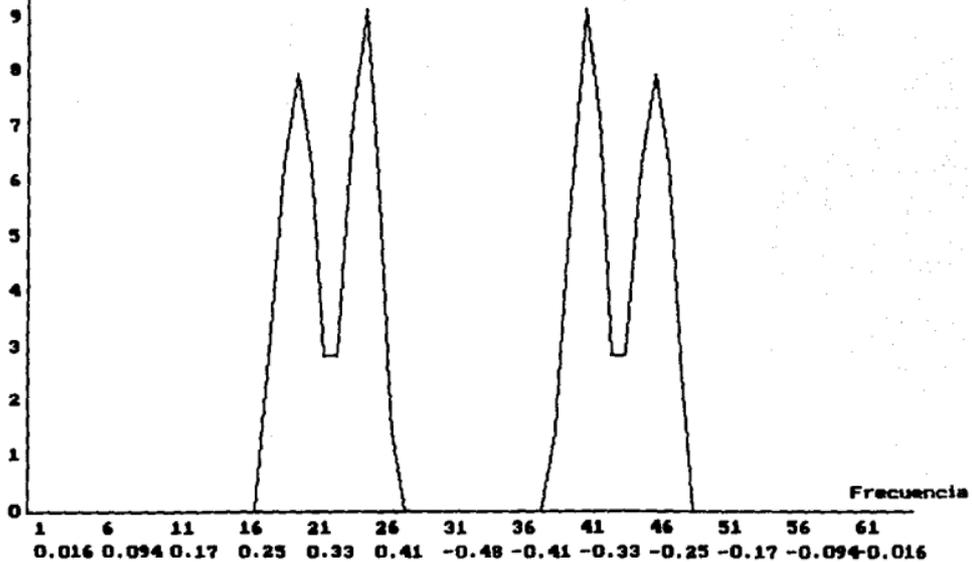




Amplitud

Gráfica de la T. de Gabor

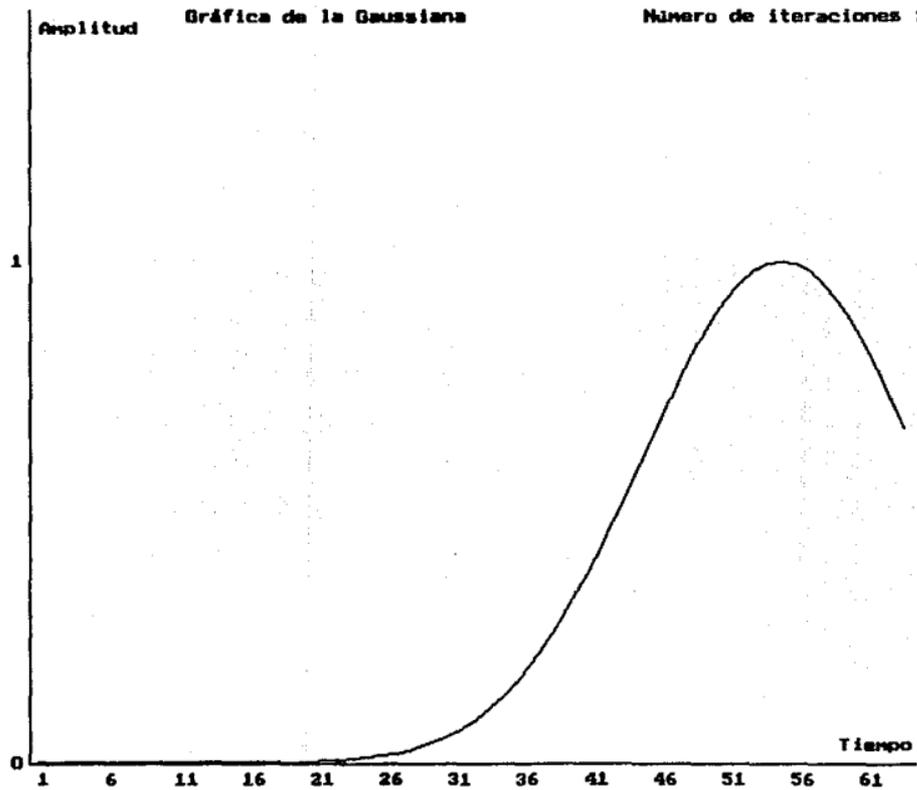
Número de iteraciones : 9

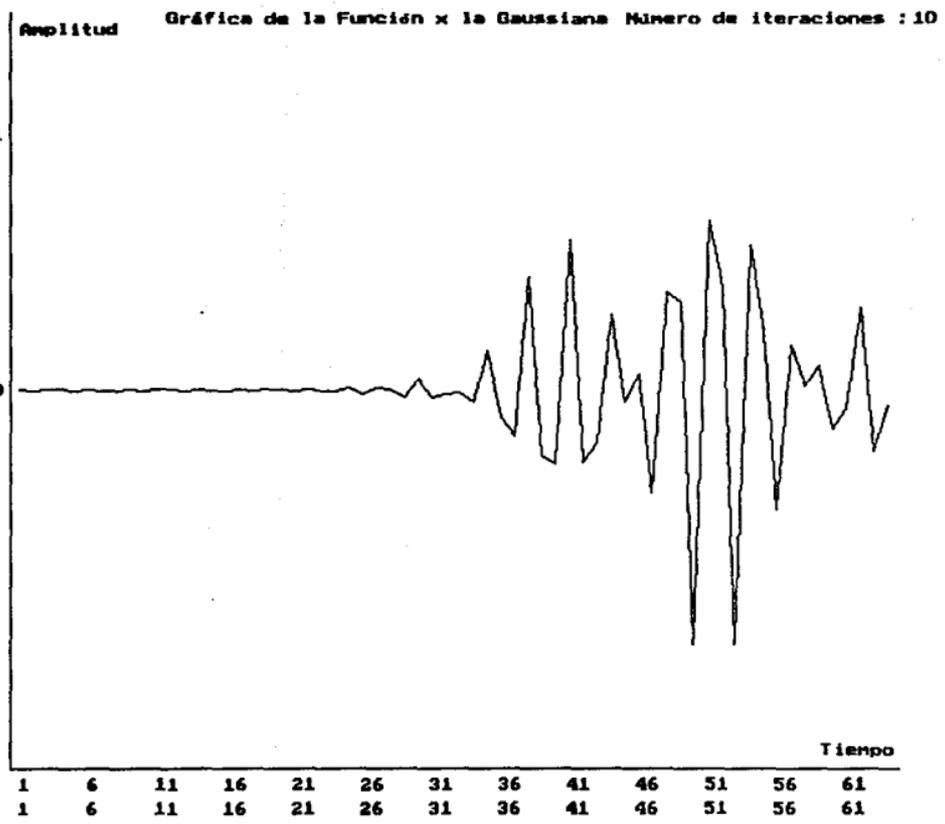


Amplitud

Gráfica de la Gaussiana

Número de iteraciones : 10

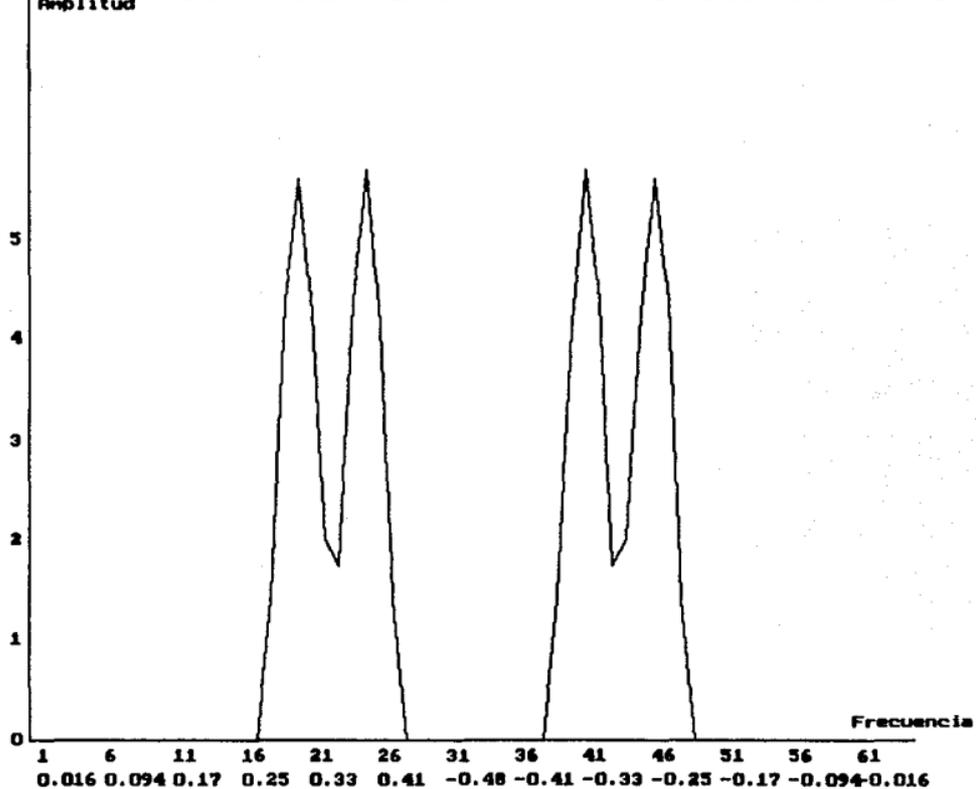




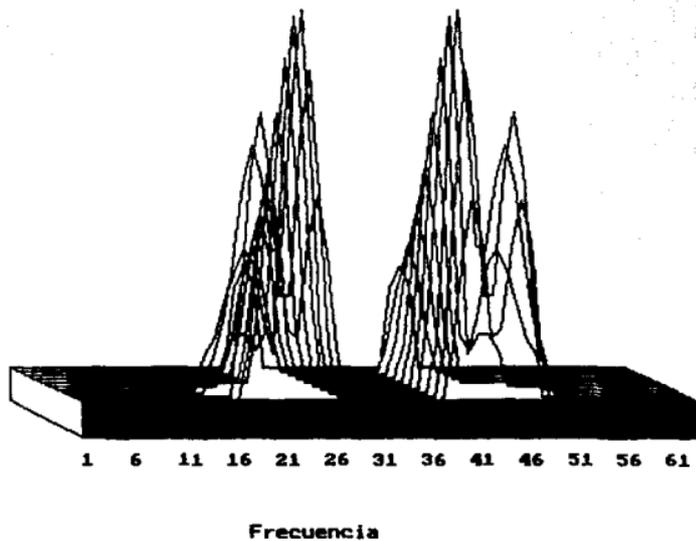
Amplitud

Gráfica de la T. de Gabor

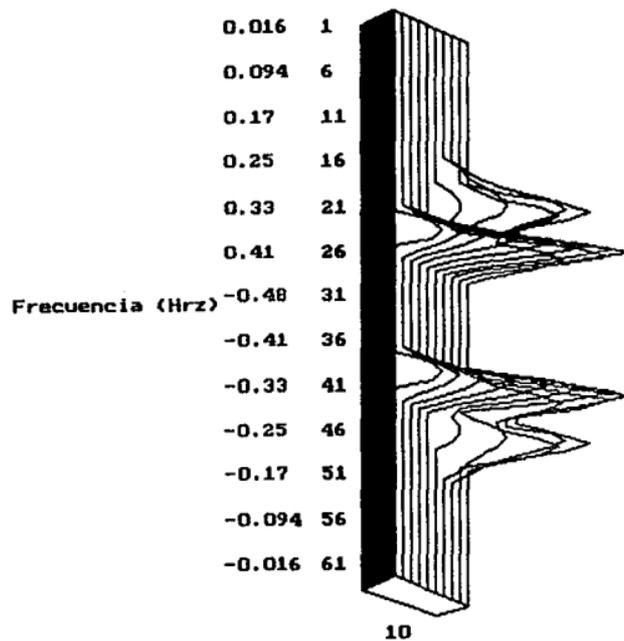
Número de iteraciones : 10



Gráfica de la T. de Gabor en 3-D



Rotación de la Gráfica en 3-D



Análisis del segundo ejemplo.

La señal son dos senos, para cada uno de ellos se calcula su frecuencia con el valor de  $h$ . Nos podemos dar cuenta que el cálculo es igual al del primer ejemplo.

Revisando las gráficas, se puede observar que las frecuencias y su posición fueron localizadas. En la última gráfica es donde se puede comparar esta localización en ambos dominios (tiempo-frecuencia).

## CONCLUSIONES

Después de haber analizado la Transformada de Gabor, y de ejecutar algunos ejemplos, podemos concluir que :

1) La Transformada de Gabor es útil para analizar señales, cuando todas las muestras aparecen aproximadamente con la misma escala.

2) La localización de la señal en el dominio de tiempo-frecuencia, es adecuado siempre y cuando se tome en cuenta a 1).

### CONCLUSION DE LOS EJEMPLOS

Uno de los objetivos de la tesis es la localización de las señales en el dominio de tiempo-frecuencia. Los ejemplos que se incluyeron trataron de exponer señales en las cuales su localización fuera un poco difícil. Esto es porque las señales contenían diferentes frecuencias y se encontraban translapadas en el tiempo. Aunque las diferencias de frecuencias no son muy significativas, con una Transformada de Fourier normal se puede lograr su separación, pero lo importante es que se incluyeron diferentes frecuencias con la finalidad de saber si con la técnica de la Transformada de Gabor se podían separar correctamente las frecuencias.

La señal empezaba con una frecuencia dada, después de ciertos intervalos de tiempo la señal se traslapaba tanto en frecuencias como en tiempos , es decir, casi a la mitad de la señal se tenía este traslape.

Se ejecutó el programa, en el cuál se obtuvieron los resultados ya mostrados, y de los cuales se puede observar lo siguiente. En sí la gráfica en la cuál vamos a dar nuestras observaciones finales es en la última (Rotación de la Gráfica en 3-D).

En esta gráfica se puede ver como las primeras Gaussianas que se encuentran son relativamente pequeñas, esto es porque la Gaussiana del desplazamiento se encuentra al principio de la señal, en el desarrollo del cálculo la Gaussianas empiezan a crecer ya que la Gaussiana del desplazamiento ventanea mayor superficie de la señal.

La localización de la segunda frecuencia empieza en la iteración 6 , en esta iteración se puede observar que la segunda Gaussiana que aparece es también relativamente pequeña, esto también sucede debido a que la frecuencia se encuentra en su inicio. Con la evolución del cálculo las Gaussianas llegan a tener casi el mismo tamaño, es exactamente en la iteración 10 cuando sucede esto.

Las frecuencias que se calcularon con la Transformada de Fourier, (ver la tercera gráfica), concordaron con las frecuencias calculadas de la Transformada de Gabor (Rotación de la Gráfica en 3-D).

Esto es importante, ya que si no hubieran concordado las frecuencias de una con otra, se podría suponer que no se estaba localizando bien el dominio de las frecuencias.

Se había dicho que la señal tiene un traslape en frecuencias y en el tiempo casi a la mitad, esto es, que se esperaba que la técnica de la Transformada de Gabor localizara en que momento la señal tenía este traslape. Y en el dominio del tiempo se encontró con que la localización de las dos frecuencias empezó en el tiempo 6, lo cual también concuerda con lo supuesto. Se puede decir que la localización en el dominio del tiempo se pudo realizar.

Entonces podemos concluir que la localización de la señal en el espacio fase frecuencia - tiempo se pudo realizar. Tenemos que si nosotros estamos escuchando nuestro concierto favorito, decidimos analizarlo, queremos saber que notas y en que momento fueron emitidas, tendríamos que analizarlo por la Transformada de Gabor, que nos dirá su localización en frecuencia - tiempo.

## LISTADO DEL PROGRAMA

El programa que a continuación se lista fue realizado en Turbo C version 2.0.

```
#include <stdio.h>
#include <math.h>
#include <graphics.h>
#include <conio.h>
#include <stdlib.h>
#include <ctype.h>
#include <string.h>
#include <fcntl.h>
#define NUM 100
int n,nu,m;
double x[NUM], y[NUM],ma[15][NUM];
char nombre[12];
char titulo[35];
float dx,ga;

main ()
{
    int ar;
    char opcion = '';
    while(opcion != '4')
    {
        opcion = '';
        ponico(1);

        do {
            gotoxy(45,18);opcion = getch();
        } while( opcion < '1' || opcion > '4');
        switch(opcion){
            case '1':
                closegraph();
                nombre_ar();
                ar = leer_archivo();
                if(ar != 0)
                {
```

```

preguntar();
calcula(1);
    }
    break;
    case '2' :
        closegraph();
        teclado();
        preguntar();
        calcula(1);
        break;
    case '3':
        closegraph();
        ejemplos();
        calcula(0);
        break;
    case '4':
        closegraph();
        exit(0);
    }
}
calcula(int ban)
{
double pi = 3.1416;

    if(ban == 0)
        normal();
    imp_res(x,y);
    strcpy(titulo,"Gráfica de la Función");
    dibuja(x,n,2,0);
    gabor();
    dib_tri();
    dib_con(pi/2.0);
}

nombre_ar()
{
    ventana();
    gotoxy(3,3);
    printf("Introduzca el nombre del archivo :");
    scanf("%s", nombre);
}

preguntar()
{
    char preg;

    do
    {
        ventana();
        gotoxy(3,3);

```

```

printf("Numero de puntos      N      : %d",n);
gotoxy(3,4);
printf("Paso de muestreo      dt     :");
scanf("%f",&dx);
gotoxy(3,5);
printf("Dilatación de la gaussina a :");
scanf("%f",&ga);
gotoxy(3,6);

printf("Introduzca el valor de  n     :");
scanf("%d",&m);
gotoxy(3,11);
printf("Estan correctos los datos  :");
preg=getch();
) while(tolower(preg) == 'n' );
)

leer_archivo()
(
float tem,tem1;
int m;
FILE *fp;
if((fp = fopen(nombre,"r"))==NULL)
(
gotoxy(3,5);
printf("No puede abrirse el archivo o no existe el archivo");
getch();
return(0);
)
m = 1;
n = 0;
while(!feof(fp))
(
fscanf(fp,"%f %f",&tem,&tem1);
x[m] = tem;
y[m] = tem1;
n = m++;
)
n--;
fclose(fp);
nu = (int)(log(n)/log(2));
)

esc_archivo()
(
FILE *fp;
int t;
nombre_ar();
if((fp = fopen(nombre,"w"))==NULL)
(
gotoxy(3,5);

```

```

    printf("No puede abrirse el archivo ");
    return;
}
for(t=1; t<=n; t++)
    fprintf(fp, "%f %f", x[t], y[t]);
fclose(fp);
}

ejemplos(void)
{
    char opc;

    ponico(2);
    do{
        gotoxy(45,20);opc = getch();
    } while( opc < '1' || opc > '4');
    switch(opc){
        case '1':
            closegraph();
            ejemplo1();
            break;
        case '2':
            closegraph();
            ejemplo2();
            break;

        case '3':
            closegraph();
            ejemplo3();
            break;
        case '4':
            closegraph();
            ejemplo4();
            break;
    }
}

ejemplo1(void)
{
    int t;
    char preg;
    double tem[NUM];
    float pi, fo, fi;
    pi = 3.1416;

    do {
        ventana();
        gotoxy(3,3);
        printf("Numero de puntos          N          :");
        scanf("%d", &n);
        gotoxy(3,4);

```

```

printf("Paso de muestreo      dt      :");
scanf("%f",&dx);
gotoxy(3,5);
printf("Introduzca la 1a frecuencia f      :");
scanf("%f",&fo);
gotoxy(3,6);
printf("Introduzca la 2a frecuencia f      :");
scanf("%f",&fi);
gotoxy(3,7);
printf("Dilatación de la gaussina a      :");
scanf("%f",&ga);
gotoxy(3,8);
printf("Introduzca el valor de n      :");
scanf("%d",&m);
gotoxy(3,11);
printf("Estan correctos los datos      :");
preg=getch();
} while(tolower(preg) == 'n' );
nu = (int)(log(n)/log(2));
for(t=1; t<=n; t++)
{
    x[t] = cos(2.0*pi*t*dx*fo);
    y[t] = 0.0;
}
for(t=1;t<=30;t++) tem[t] = 0.0;
for(t=30; t<=n; t++) tem[t] = cos(2.0*pi*t*dx*fi);
for(t=1;t<=n;t++)
    x[t] = x[t] + tem[t];
}

```

ejemplo2(void)

```

{
    int t;
    char preg;
    float pi,fo,fi;
    pi = 3.1416;
do {
    ventana();
    gotoxy(3,3);
    printf("Numero de puntos      N      :");
    scanf("%d",&n);
    gotoxy(3,4);
    printf("Paso de muestreo      dt      :");
    scanf("%f",&dx);
    gotoxy(3,5);
    printf("Introduzca la frecuencia f      :");
    scanf("%f",&fo);
    gotoxy(3,6);
    printf("Introduzca la 2a frecuencia f      :");
    scanf("%f",&fi);
}

```

```

gotoxy(3,7);
printf("Dilatación de la gaussina a      :");
scanf("%f",&ga);
gotoxy(3,8);
printf("Introduzca el valor de n          :");
scanf("%d",&m);
gotoxy(3,11);
printf("Estan correctos los datos        :");
preg=getch();
) while(tolower(preg) == 'n' );
nu = (int)(log(n)/log(2));
for(t=1; t<=n; t++)
(
    x[t] = sin(2.0*pi*t*dx*fo)+sin(2.0*pi*t*dx*fi);;
    y[t] = 0.0;
)
)

```

ejemplo3(void)

```

(
    int t;
    double tem[NUM];
    char preg;
    float pi,fo,fi;

    pi = 3.1416;
do {
    ventana();
    gotoxy(3,3);
    printf("Numero de puntos      N      :");
    scanf("%d",&n);
    gotoxy(3,4);
    printf("Paso de muestreo      dt      :");
    scanf("%f",&dx);
    gotoxy(3,5);
    printf("Introduzca la frecuencia f      :");
    scanf("%f",&fo);
    gotoxy(3,6);
    printf("Introduzca la 2a frecuencia f      :");
    scanf("%f",&fi);
    gotoxy(3,7);
    printf("Dilatación de la gaussina a      :");
    scanf("%f",&ga);
    gotoxy(3,8);
    printf("Introduzca el valor de n          :");
    scanf("%d",&m);
    gotoxy(3,11);
    printf("Estan correctos los datos        :");
    preg=getch();
) while(tolower(preg) == 'n' );
nu = (int)(log(n)/log(2));

```

```

for(t=1; t<=n; t++)
{
    x[t] = sin(2.0*pi*t*dx*fo);
    y[t] = 0.0;
}
for(t=1;t<=30;t++) tem[t] = 0.0;
for(t=30; t<=n; t++) tem[t] = sin(2.0*pi*t*dx*fi);
for(t=1;t<=n;t++)
    x[t] = x[t] + tem[t];
}
ejemplo4(void)
{
    int t;
    float h,hi;
    double pi,fo,fi;
    double tem[NUM];
    pi = 3.1416;
    n = 64;
    dx = 1.0;
    ventana();
    gotoxy(3,3);
    printf("Ejemplo de frecuencia de musica 2^(h/12) ");
    gotoxy(3,5);
    printf("Introduzca el valor de 1 - 12 :");
    scanf("%f",&h);
    gotoxy(3,6);
    printf("Introduzca el valor de 1 - 12 :");
    scanf("%f",&hi);
    gotoxy(3,7);
    printf("Dilatación de la gaussina a :");
    scanf("%f",&ga);
    gotoxy(3,8);
    printf("Introduzca el valor de n :");
    scanf("%d",&m);
    nu = (int)(log(n)/log(2));
    fo = exp((h/12.0)*log(2.0));
    fi = exp((hi/12.0)*log(2.0));
    for(t=1; t<=n;t++)
    {
        x[t] = sin(2.0*pi*t*dx*fo);
        y[t] = 0.0;
    }
    for(t=1; t<=n;t++)
        tem[t] = 0.0;

    for(t=(n/2);t<=n; t++)
        tem[t] = sin(2.0*pi*t*dx*fi);
    for(t=1;t<=n; t++)
        x[t] = x[t]+tem[t];
}

```

```

teclado(void)
(
  int t,m,k;
  float tem,tem1;

  ventana();
  gotoxy(3,3);
  printf("Introduzca la cantidad de puntos N :");
  scanf("%d",&n);
  nu = (int)(log(n)/log(2));
  m =2;
  k =5;
  for(t=1; t<=n; t++)
  (
    gotoxy(m,++k);
    printf("Real(%d) = ",t);
    scanf("%f",&tem);
    x[t] = tem;
    gotoxy(m+40,k);
    printf("Imag(%d) = ",t);
    scanf("%f",&tem1);
    y[t] = tem1;
    if(k >=22)
    (
      k= 2;
      gotoxy(23,23);
      getch();
      ventana();
    )
  )
)

ventana()
(
  int i;

  clrscr();
  gotoxy(1,1); putchar('F ');
  gotoxy(2,1);
  for(i=1;i<=79;i++) putchar('- ');
  for(i=2;i<=23;i++)
  (
    gotoxy(1,i); putchar('I');
    gotoxy(80,i); putchar('I');
  )
  for(i=1;i<=79;i++) putchar('- ');
  gotoxy(80,1); putchar('I');
  gotoxy(1,24); putchar('I');
  gotoxy(80,24); putchar('I');
)

```

```

dibuja(double xreal[], int n,int ft,int t)
{
    double es,es1;
    int driver=DETECT, mode;
    int i,j,escala_i,escala_x,nvga,mvga,y_res,ymax,max;
    int menos,y_res1;
    char *ap = " ";
    char cad_num[18];
    mvga = 300;
    nvga = 600;
    y_res = 450;
    ymax = 0;

    for(i=1;i<=n-1;i++)
    {
        for(j=2;j<=n;j++)
            (int)max = (xreal[i] > xreal[j]) ? xreal[i] : xreal[j];
        ymax = (max > ymax) ? max : ymax;
    }
    menos = 0;
    for(j=1;j<=n;j++)
        if (xreal[j] < 0) menos = 1;
    escala_i = (int)nvga/n;
    if(menos >= 1){
        y_res1 = y_res/2;
        mvga = 150;
        nvga = 300;
        escala_i = (int)nvga/(n/2);
    }
    else
        y_res1 = y_res;
    if(ymax <= 0) ymax = 1;
    escala_x = (int)mvga/ymax;
    initgraph(&driver,&mode,ap);
    for(i=1;i<=n;i+=5) {
        itoa(i,cad_num,10);
        outtextxy(i*escala_i+15,y_res+6,cad_num);
    }
    if(ft == 1)
    {
        for(i=1;i<=(n/2)-2;i+=5) {
            es = (double)i/(double)(n*dx);
            gcvt(es,2,cad_num);
            outtextxy(i*escala_i+15,y_res+20,cad_num);
        }
        j = (n/2)-1;
        for(i=(n/2)-1;i>=1;i-=5) {
            es = (double)-i/(double)(n*dx);
            gcvt(es,2,cad_num);
            outtextxy(j*escala_i+15,y_res+20,cad_num);
            j += 5;
        }
    }
}

```

```

if(ft == 2)
{
for(i=1;i<=n;i+=5) {
esl = i*dx;
itoa(esl,cad_num,10);
outtextxy(i*escala_i+15,y_res+20,cad_num);
}
}

settextjustify(0,1);
for(j=y_max;j>=0;j--) {
itoa(j,cad_num,10);
outtextxy(2,y_res1-j*escala_x,cad_num);
}
line(18,y_res,18,1);
line(18,y_res,600,y_res);
setcolor(YELLOW);
moveto(escala_i+15,y_res1-xreal[1]*escala_x);
for(i=2;i<=n;i++) lineto(i*escala_i+15,y_res1-xreal[i]*escala_x);
setcolor(WHITE);
settextstyle(1,0,2);
outtextxy(120,5,titulo);
if(t != 0)
{
strcpy(titulo,"Numero de iteraciones : ");
itoa(t,cad_num,10);
strcat(titulo,cad_num);
outtextxy(420,5,titulo);
}
getch();
closegraph();
}

dib_tri()
{
double h[NUM],di,d2;
char cad_num[18];
int t,k,esx,esy,val;
int driver=DETECT, mode;
int i,escala_i,escala_x,nvga,mvga,y_res;
char *ap = "";

mvga = 200;
nvga = 400;
y_res = 350;
val=40;
escala_i = (int)nvga/n;
escala_x = (int)mvga/ga;

initgraph(&driver,&mode,ap);
esx = 0; esy = 0;
setcolor(YELLOW);

```

```

di = ma[1][1];
d2 = ma[1][n];
for(t=1;t<=m;t++)
(
for(k=1;k<=n;k++) h[k] = ma[t][k];
esx += 5;
esy += 2;
moveto(escala_i+val+esx,y_res-h[1]*escala_x+esy);
for(i=2;i<=n;i++) lineto(i*escala_i+val+esx,y_res-h[i]*escala_x+esy);
)
line(escala_i+val+esx,y_res-di*escala_x+esy,n*escala_i+val+esx,
y_res-h[1]*escala_x+esy);
line(escala_i+val+esx,y_res-di*escala_x+esy,escala_i+val+5,
y_res-di*escala_x+2);
line(escala_i+val+5,y_res-di*escala_x+2,escala_i+val+5,
y_res-di*escala_x+esy);
line(n*escala_i+val+esx,y_res-h[1]*escala_x+esy,n*escala_i+val+esx,
y_res-h[1]*escala_x+esy+esy);
line(escala_i+val+esx,y_res-di*escala_x+esy,escala_i+val+esx,
y_res-di*escala_x+esy+esy);
line(escala_i+val+5,y_res-di*escala_x+esy,escala_i+val+esx,
y_res-di*escala_x+esy+esy);
line(escala_i+val+esx,y_res-di*escala_x+esy+esy,n*escala_i+val+esx,
y_res-h[1]*escala_x+esy+esy);
line(n*escala_i+val+esx,y_res-h[1]*escala_x+esy,n*escala_i+val+5,
y_res-d2*escala_x+2);
setfillstyle(SOLID_FILL, YELLOW);
floodfill(escala_i+val+esx+esx,y_res-di*escala_x+esy+2,YELLOW);
setcolor(WHITE);
for(i=1;i<=n;i+=5) {
itoa(i,cad_num,10);
outtextxy(i*escala_i+90,400,cad_num);
}
settextstyle(3,0,2);
strcpy(titulo,"Grafica de la T. de Gabor en 3-D");
outtextxy(120,5,titulo);
strcpy(titulo,"Frecuencia");
outtextxy(200,450,titulo);
getch();
closegraph();
)

dib_con(double teta)
(
double hx[NUM],hy[NUM],es;
char cad_num[18];
int y_res,escala_i,escala_x;
int i,j,t,k,esx,esy,val;
int driver=DETECT, mode;
char *ap = "";
double pi = 3.1416;
initgraph(&driver,&mode,ap);

```

```

esx = 0; esy = 0;
if(teta == pi)
{
escala_i = 5;
escala_x = 10;
val = 400;
y_res = 50;
}
if(teta == pi/2.0)
{
escala_i = 10;
escala_x = 5;
val = 250;
y_res = 40;
}
setcolor(YELLOW);
for(t=1;t<=m;t++)
{
for(k=1;k<=n;k++) hx[k] = ma[t][k];
for(k=1;k<=n;k++)
{
hy[k] = hx[k]*sin(teta) + k*cos(teta);
hx[k] = hx[k]*cos(teta) - k*sin(teta);
}
esx += 5;
esy += 2;
moveto(hy[1]*escala_i+val+esx,y_res-hx[1]*escala_x+esy);
for(i=2;i<=n;i++) lineto(hy[i]*escala_i+val+esx,
y_res-hx[i]*escala_x+esy);
}
line(hy[1]*escala_i+val+5,y_res-hx[1]*escala_x+2,hy[n]*escala_i+val+5,
y_res-hx[n]*escala_x+2);
line(hy[1]*escala_i+val+5,y_res-hx[1]*escala_x+2,hy[1]*escala_i+val+es
y_res-hx[1]*escala_x+esy);
line(hy[n]*escala_i+val+5,y_res-hx[n]*escala_x+2,hy[n]*escala_i+val+es
y_res-hx[n]*escala_x+esy);
line(hy[n]*escala_i+val+5,y_res-hx[n]*escala_x+2,hy[n]*escala_i+val-15
y_res-hx[n]*escala_x+10);
line(hy[n]*escala_i+val+esx,y_res-hx[n]*escala_x+esy,hy[n]*escala_i+va
y_res-hx[n]*escala_x+esy+5);
line(hy[n]*escala_i+val-15,y_res-hx[n]*escala_x+10,hy[n]*escala_i+val+
y_res-hx[n]*escala_x+esy+5);
line(hy[1]*escala_i+val+5,y_res-hx[1]*escala_x+2,hy[1]*escala_i+val-15
y_res-hx[1]*escala_x+10);
line(hy[1]*escala_i+val-15,y_res-hx[1]*escala_x+10,hy[n]*escala_i+val-
y_res-hx[n]*escala_x+10);
setfillstyle(SOLID_FILL, YELLOW);
floodfill(hy[n]*escala_i+val-5,y_res-hx[n]*escala_x,YELLOW);
setcolor(WHITE);
for(i=1;i<=m;i+=9) {
itoa(i,cad_num,10);
outtextxy(hy[i]*escala_i+val,390,cad_num);
}

```

```

}
settextjustify(0,1);
for(j=1;j<=n;j+=5)
{
itoa(j,cad_num,10);
outtextxy(210,y_res+j*escala_x+10,cad_num);
}
for(j=1;j<=(n/2)-2;j+=5)
{
es = (double)j/(double)(n*dx);
gcvt(es,2,cad_num);
outtextxy(150,y_res+j*escala_x+10,cad_num);
}
i = (n/2)-1;
for(j=(n/2)-1;j>=1;j-=5)
{
es = (double)-j/(double)(n*dx);
gcvt(es,2,cad_num);
outtextxy(150,y_res+i*escala_x+10,cad_num);
}
i = (n/2)-1;
for(j=(n/2)-1;j>=1;j-=5)
{
es = (double)-j/(double)(n*dx);
gcvt(es,2,cad_num);
outtextxy(150,y_res+i*escala_x+10,cad_num);
i +=5;
}
settextstyle(3,0,2);
strcpy(titulo,"Rotacion de la Grafica en 3-D");
outtextxy(120,5,titulo);
strcpy(titulo,"Frecuencia (Hrz)");
outtextxy(20,210,titulo);
strcpy(titulo,"Tiempo(Posicion)");
outtextxy(120,460,titulo);
getch();
closegraph();
}

imp_res(double xreal[],double ximag[])
{
int t,m,k;
char preg;
ventana();
m =2;
k =2;
for(t=1; t<=n; t++)
{
gotoxy(m,++k);
printf("%(d) = %f + i%f ",t,xreal[t],ximag[t]);
if(k >=22)
{

```

```

        k= 2;
        gotoxy(23,23);
        getch();
        ventana();
    }
    gotoxy(23,23);
    printf("Desea salvar esta informacion :");
    prag=getch();
    if(tolower(prag) == 's') esc_archivo();
}

ibit(j,nu)
int j,nu;
{
    int i,j1,j2,k;

    j1=j;
    k=0;
    for(i=1; i <= nu; i++)
    {
        j2 = j1/2;
        k = k*2+(j1-2*j2);
        j1 = j2;
    }
    return(k);
}

pote(int nui)
{
    int f;
    /* f = (int)((double)exp(nui * (double)log(2.0)))/
    if (nui >= 1)
        f = (int)pow(2.0, (double)nui);
    else f = 1;
    return(f);
}

poten(double b)
{
    double f;
    f = b*b;
    return(f);
}

normal()
{
    int i;
    double d,s,med,w[NUM],gas[NUM];
    for(i=1;i<=n;i++)

```

```

    w[i] = i*dx;
    med = 0.0;
    for(i=1;i<=n;i++) med = med + w[i];
    med = med/n;
    s = 0.0;
    for(i=1;i<=n;i++) s = s+poten(med - w[i]);
    if (s < 0.0) s = (-1.0)*s;
    s = sqrt(s/n);
    d = poten(s);
    for(i=1;i<=n;i++)
        gas[i] = exp(-poten(w[i]-med)/(2.0*d));
    for(i=1; i<=n; i++) x[i] = x[i]*gas[i];
}

gauss(double z[],double g[],double a)
{
    int i;
    double b;
    b = poten(a);
    for(i=1;i<=n;i++)
        g[i] = exp(-poten(z[i])/(2.0*b));
}

gabor()
{
    double g[NUM],w[NUM],h[NUM],hl[NUM];
    int i,j,t;
    double a;
    a = ga;
    for(j=1;j<=n;j++)
    {
        h[j] = x[j];
        hl[j] = y[j];
    }
    fft(h,hl);
    for(j=1;j<=n;j++) h[j] = sqrt(poten(h[j]) + poten(hl[j]));
    strcpy(titulo,"F.F.T. de la Función");
    dibuja(h,n,1,0);
    for(i=1;i<=n;i++)
        w[i] = i*dx;
    for(t=1;t<=m;t++)
    {
        for(j=1;j<=n;j++) w[j] = w[j]-t*dx;
        gauss(w,g,a);
        strcpy(titulo,"Grafica de la Gaussiana");
        dibuja(g,n,0,t);
        for(j=1;j<=n;j++) h[j] = g[j]*x[j];
        for(j=1;j<=n;j++) hl[j] = 0.0;
        strcpy(titulo,"Grafica de la Funcion x la Gaussiana");
        dibuja(h,n,0,t);
        fft(h,hl);
        for(j=1;j<=n;j++) h[j] = sqrt(poten(h[j]) + poten(hl[j]));
    }
}

```

```

strcpy(titulo,"Grafica de la T. de Gabor");
dibuja(h,n,1,t);
for(j=1;j<=n;j++) ma[t][j] = h[j];
}
fft(double xreal[],double ximag[])
{
int n2,nul,i,l,k,kl,kln2,m1,pote(),ibit();
double treal,timag,p,arg,c,s,pi;
n2 = n/2;
nul = nu-1;
pi = 3.1416;
k = 0;
for(l=1; l <= nu; l++)
{
while(k < n)
{
for(i=1; i <= n2; i++)
{
m1 = k/pote(nul);
p = ibit(m1,nu);
arg =(double) (2.0*pi*p)/n;
c = cos(arg);
s = sin(arg);
kl = k +1;
kln2 = kl + n2;
treal = xreal[kln2]*c + ximag[kln2]*s;
timag = ximag[kln2]*c - xreal[kln2]*s;
xreal[kln2] = xreal[kl] - treal;
ximag[kln2] = ximag[kl] - timag;
xreal[kl] = xreal[kl] + treal;
ximag[kl] = ximag[kl] + timag;
k = k+1;
}
k = k + n2;
}
k = 0;
nul = nul - 1;
n2 = n2/2;
}
for(k = 1; k <= n; k++)
{
m1 = k -1;
l = ibit(m1,nu);
i = l + 1;
if(i > k)
{
treal = xreal[k];
timag = ximag[k];
xreal[k] = xreal[i];
ximag[k] = ximag[i];
}
}
}

```



```
outtextxy(180,166,titulo2);
BitMap = malloc(SizeBitMap);
HandleFile = open(path3,O_RDONLY);
read(HandleFile,BitMap,SizeBitMap);
rectangle(90,190,142,242);
putimage(100,200,BitMap,COPY_PUT);
close(HandleFile);
free(BitMap);
outtextxy(180,216,titulo3);
BitMap = malloc(SizeBitMap);
HandleFile = open(path4,O_RDONLY);
read(HandleFile,BitMap,SizeBitMap);
rectangle(90,240,142,292);
putimage(100,250,BitMap,COPY_PUT);
close(HandleFile);
free(BitMap);
outtextxy(180,266,titulo4);
outtextxy(180,300,"Introduzca la opcion :");
BitMap = NULL;
)
```

## APENDICE A

### A.1

#### LA TRANSFORMADA DE FOURIER

La transformada de Fourier  $\hat{h}(f)$  de una función  $h(t)$  está definida como :

$$\hat{h}(f) = \int_{-\infty}^{+\infty} h(t) e^{-j2\pi ft} dt \quad (i)$$

$t$  es tiempo y  $f$  es frecuencia.

La Transformada de Fourier es una función compleja y por tanto se puede escribir en la forma :

$$\hat{h}(f) = R(f) + jI(f) = |H(f)| e^{j\theta(f)} \quad (ii)$$

donde

$R(f)$  es la parte real .

$I(f)$  es la parte imaginaria, donde  $j = \sqrt{-1}$

$|\hat{h}(f)|$  es el modulo

$$[R^2(f) + I^2(f)]^{1/2}.$$

y el ángulo está dado por  $\theta(f) = \tan^{-1} [I(f)/R(f)]$ .

La transformada inversa está dada por :

$$h(t) = \int_{-\infty}^{+\infty} \hat{h}(f) e^{j2\pi ft} df \quad (iii)$$

las funciones (i) y (iii) están relacionadas mediante la notación

$$T[h(t)] = \hat{h}(f).$$

## A.2

### PROPIEDADES DE LA TRANSFORMADA DE FOURIER

Linealidad.

Si  $x(t)$  y  $y(t)$  tienen las Transformadas de Fourier  $\hat{x}(f)$  y  $\hat{y}(f)$ , respectivamente, entonces la suma de  $x(t) + y(t)$  tiene como Transformada de Fourier la suma de sus Transformadas  $\hat{x}(f) + \hat{y}(f)$ . Esta propiedad es establecida como sigue:

$$\begin{aligned} \int_{-\infty}^{+\infty} [x(t)+y(t)]e^{-j2\pi ft} dt &= \int_{-\infty}^{+\infty} x(t)e^{-j2\pi ft} dt + \int_{-\infty}^{+\infty} y(t)e^{-j2\pi ft} dt \\ &= \hat{x}(f) + \hat{y}(f) \end{aligned} \quad (\text{vi})$$

Simetría.

Si  $h(t)$  es una función y  $\hat{h}(f)$  es la de Transformadas de Fourier, entonces:

$$T[h(-f)] = \hat{h}(t) \quad (\text{vii})$$

volviendo a escribir la ecuación tenemos:

$$h(-t) = \int_{-\infty}^{+\infty} \hat{h}(f) e^{-j2\pi f t} df \quad (\text{viii})$$

e intercambiando los parámetros  $t$  y  $f$  tenemos:

$$h(-f) = \int_{-\infty}^{+\infty} \hat{h}(t) e^{-j2\pi f t} dt \quad (\text{ix})$$

Escalamiento en el Tiempo.

Si la Transformada de Fourier de  $h(t)$  es  $\hat{h}(f)$ , entonces la Transformada de Fourier de  $h(kt)$  es  $\frac{1}{|k|} \hat{h}(f/k)$ , donde  $k$  es una constante real mayor que cero, se determina por sustituir  $t' = kt$  en la ecuación de la Transformada de Fourier. Por lo tanto tenemos:

$$\int_{-\infty}^{+\infty} h(kt) e^{-j2\pi f t} dt = \int_{-\infty}^{+\infty} h(t') e^{-j2\pi t' (f/k)} \frac{dt'}{k} = \frac{1}{|k|} \hat{h}(f/k) \quad (\text{x})$$

Escalamiento en la frecuencia.

Si la Transformada de Fourier de  $h(t)$  es  $\hat{h}(f)$ , la Transformada inversa de Fourier de  $\hat{h}(kf)$  es  $\frac{1}{|k|} h(t/k)$ ;  $k$  es mayor que cero. La relación es establecida por sustituir  $f' = kf$  en la fórmula inversa:

$$\int_{-\infty}^{+\infty} \hat{h}(kf) e^{j2\pi f t} df = \int_{-\infty}^{+\infty} \hat{h}(f') e^{j2\pi f' (t/k)} \frac{df'}{k} = \frac{1}{|k|} h(t/k) \quad (\text{xi})$$

el escalamiento en la frecuencia de las funciones impulso está dada por

$$\delta(af) = \frac{1}{|a|} \delta(f) \quad (\text{xii})$$

Traslación en el tiempo.

Si  $h(t)$  es trasladada por una constante  $t_0$ , entonces la Transformada de Fourier de  $h(t)$  es :

$$\begin{aligned} \int_{-\infty}^{+\infty} h(t - t_0) e^{-j2\pi ft} dt &= \int_{-\infty}^{+\infty} h(s) e^{-j2\pi f(s + t_0)} ds \\ &= e^{-j2\pi ft_0} \int_{-\infty}^{+\infty} h(s) e^{-j2\pi fs} ds \\ &= e^{-j2\pi ft_0} H(f) \quad (\text{xiii}) \end{aligned}$$

∴ la Transformada de Fourier es:

$$T[h(t - t_0)] = \hat{h}(f) e^{-j2\pi ft_0} \quad (\text{xiv})$$

Traslación en la frecuencia.

Si  $\hat{h}(f)$  es trasladada por una constante  $f_0$ , entonces la función es multiplicada por  $e^{j2\pi t f_0}$ , tenemos:

$$T[h(t) e^{j2\pi t f_0}] = \hat{h}(f - f_0) \quad (\text{xvi})$$

La Transformada de Fourier está establecida por la sustitución  $s = f - f_0$  en la Transformada inversa de Fourier, definiendo la relación:

$$\int_{-\infty}^{+\infty} H(f - f_0) e^{j2\pi ft} dt = \int_{-\infty}^{+\infty} H(s) e^{j2\pi t(s+f_0)} ds$$

$$\begin{aligned}
 &= e^{j2\pi t f_0} \int_{-\infty}^{+\infty} H(s) e^{j2\pi t s} ds \\
 &= e^{j2t f_0} h(t) \quad (\text{xvii}).
 \end{aligned}$$

### A.3

#### CONVOLUCION Y CORRELACION

La convolución de 2 funciones está dada por la integral :

sea  $h(t)$  y  $x(t)$

entonces

$$Y(t) = \int_{-\infty}^{+\infty} x(\tau) h(t - \tau) dt \quad (\text{xxv})$$

entonces  $Y(t) = h(t) * x(t)$  (xxvi)

que es lo mismo que

$$Y(t) = \int_{-\infty}^{+\infty} h(\tau) x(t - \tau) dt \quad (\text{xxvii})$$

La integral de la correlación es :

$$Z(t) = \int_{-\infty}^{+\infty} x(\tau) h(t + \tau) d\tau \quad (\text{xxviii})$$

Teorema de convolución en el tiempo.

Si  $h(t)$  tiene la T.F.  $\hat{h}(f)$  y  $x(t)$  tiene la T.F.  $\hat{x}(f)$ , entonces la convolución de las funciones  $h(t) * x(t)$  tienen como T.F. la multiplicación de sus T.F. respectivamente, esto es :

$$F[h(t)*x(t)] = \hat{h}(f)\hat{x}(f) \quad (\text{xxviii})$$

Desarrollo

$$\begin{aligned} \hat{h}(f)\hat{x}(f) &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x(\tau) h(t - \tau) d\tau e^{-j\pi 2ft} dt \\ &= \int_{-\infty}^{+\infty} x(\tau) \int_{-\infty}^{+\infty} h(t - \tau) e^{-j\pi 2ft} dt d\tau \end{aligned}$$

por el teorema de translacion en el tiempo ,

$$= \int_{-\infty}^{+\infty} x(\tau) e^{-j\pi 2f\tau} \hat{h}(f) d\tau = \hat{h}(f)\hat{x}(f)$$

Teorema de convolución en la frecuencia.

La Transformada de Fourier del producto de dos funciones  $h(t)x(t)$  , es igual a la convolucion de sus Transformadas.

$$F[h(t)x(t)] = \hat{h}(f)*\hat{x}(f) \quad (\text{xxix})$$

$$\begin{aligned} \hat{h}(f)*\hat{x}(f) &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \hat{h}(\tau) \hat{x}(t - \tau) d\tau e^{j\pi 2ft} df \\ &= \int_{-\infty}^{+\infty} \hat{h}(\tau) \int_{-\infty}^{+\infty} \hat{x}(t - \tau) e^{j\pi 2ft} df d\tau \end{aligned}$$

por el teorema de translación en la frecuencia,

$$= \int_{-\infty}^{+\infty} \hat{h}(\tau) e^{j\pi 2f\tau} x(f) df = x(f) h(f)$$

## A.4

## TEOREMA DE PARSEVAL

La energía calculada en el dominio del tiempo, debe ser igual a la energía calculada en el dominio de la frecuencia.

$$\int_{-\infty}^{+\infty} h^2(t) dt = \int_{-\infty}^{+\infty} |\hat{h}(f)|^2 df \quad (\text{xxx})$$

Desarrollo

$$\begin{aligned} \int_{-\infty}^{+\infty} h^2(t) dt &= \int_{-\infty}^{+\infty} h(t) \overline{h(t)} dt \\ &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \hat{h}(f) e^{j\pi 2ft} \overline{\hat{h}(f') e^{j\pi 2f't}} dt df df' \\ &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \hat{h}(f) \overline{\hat{h}(f')} e^{j\pi 2ft} e^{-j\pi 2f't} dt df df' \\ &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \hat{h}(f) \overline{\hat{h}(f')} \delta(f - f') df df' \\ &= \int_{-\infty}^{+\infty} \hat{h}(f) \overline{\hat{h}(f)} df = \int_{-\infty}^{+\infty} |\hat{h}(f)|^2 df \end{aligned}$$

## A.5

## TRANSFORMADA DISCRETA DE FOURIER

La Transformada Discreta de Fourier está definida de la siguiente forma

$$D(n/NT) = T \sum_{k=0}^{N-1} d(kT) e^{-j2\pi nk/N} \quad (\text{xxx1})$$

para  $n = 0, 1, \dots, N-1$

## Desarrollo

La Transformada de Fourier es

$$D(n/NT) = \int_{-\infty}^{+\infty} d(kT) e^{-j2\pi nk/N} dt$$

por la Integral de Riemann tenemos que

$$\begin{aligned} &= \int_{-\infty}^{+\infty} d(kT) e^{-j2\pi nk/N} dkT \\ &= \lim_{N \rightarrow \infty} \sum_{k=0}^{N-1} d(kT) e^{-j2\pi nk/N} \Delta kT \\ &= T \sum_{k=0}^{N-1} d(kT) e^{-j2\pi nk/N} \end{aligned}$$

Transformada Discreta Inversa de Fourier.

La transformada inversa está dada por :

$$d(kT) = 1/N \sum_{n=0}^{N-1} D(n/NT) e^{j2\pi nk/N} \quad (\text{xxxii})$$

para  $k = 0, 1, \dots, N-1$

Desarrollo.

$$\begin{aligned} D(n/NT) &= T \sum_{k=0}^{N-1} 1/N \sum_{r=0}^{N-1} D(r/NT) e^{j2\pi rk/N} e^{-j2\pi nk/N} \\ &= T/N \sum_{r=0}^{N-1} D(r/NT) e^{j2\pi rk/N} e^{-j2\pi nk/N} \\ &= T \sum_{r=0}^{N-1} D(r/NT) \\ &= D(n/NT). \end{aligned}$$

La Transformada Discreta de Fourier tiene las mismas propiedades de la Transformada de Fourier.

## A.6

### LA TRANSFORMADA RAPIDA DE FOURIER (FFT)

Como vimos anteriormente la Transformada discreta, se puede realizar mediante un programa, la cantidad de operaciones (multiplicaciones) que se necesitan son de  $N^2$ . La FFT solamente necesita  $N \ln N$  multiplicaciones. Por ejemplo para  $N = 1024$  tenemos que

$$\begin{aligned} N &= 1024 && = 2^{10} \\ N^2 &= 1\,048\,576 = 2^{20} && \text{operaciones en DFT.} \\ N \ln N &= 10\,240 = (2^{10})(10) && \text{operaciones en FFT.} \end{aligned}$$

Matemáticamente la FFT involucra combinaciones de punto  $X_n$  en pares e impares. Se utiliza la periodicidad de la exponencial para eliminar las operaciones redundantes.

Sea

$$D(n/NT) = T \sum_{k=0}^{N-1} d(kT) e^{-j2\pi nk/N} \quad (1)$$

para  $n = 0, 1, \dots, N-1$

$$Y_k = X_{2k} \quad \text{para } k = 0, 1, \dots, (N/2-1)$$

$$Z_k = X_{2k-1}$$

La ecuación (1) se puede escribir como

$$A_n = \sum_{k=0}^{(N/2)-1} ( Y_k e^{-j4\pi nk/N} + Z_k e^{-j2\pi n(2k+1)/N} ) \quad (2)$$

con  $n = 0, 1, \dots, N - 1$ .

y se puede volver a escribir sacando el factor de  $Z_k$

$$A_n = \sum_{k=0}^{(N/2)-1} Y_k e^{-j4\pi nk/N} + e^{-j2\pi n/N} \sum_{k=0}^{(N/2)-1} Z_k e^{-j4\pi nk/N} \quad (3)$$

Para simplificarlo lo escribimos como

$$A_n = B_n + W^n C_n \quad \text{para } 0 \leq n < N/2$$

donde

$$B_n = Y_k e^{-j4\pi nk/N}$$

$$C_n = Z_k e^{-j4\pi nk/N}$$

$$W^n = e^{-j2\pi n/N}$$

Pero como  $B_n$  y  $C_n$  tiene un periodo de medio intervalo, se puede generar  $A_n$  para la siguiente mitad, utilizando las mismos valores de  $B_n$  y  $C_n$

$$A_{n+N/2} = B_n - W^n C_n \quad \text{para } 0 \leq n < N/2$$

El cálculo de  $N$  puntos de la transformada  $A$ , ha sido reducido a  $N/2$  multiplicaciones, utilizando las transformadas de  $B_n$ ,  $C_n$  y el factor de  $W^n$ . Esto ilustra como trabaja la FFT.

## A.7

## LA FUNCION DELTA DE DIRAC

La Funcion Delta se ocupa mucho en el análisis de la Transformada de Fourier, tanto discreta como continua, es usada para simplificar muchas derivadas así como argumentos complejos. Usualmente tiene la representación de :

$$\delta(x) = \lim_{a \rightarrow \infty} \int_{-a}^{+a} e^{\pm j2\pi xy} dy$$

Propiedades de la Funcion Delta.

$$\int_{-\infty}^{+\infty} \delta(t) f(t) dt = f(0)$$

$$\int_{-\infty}^{+\infty} \delta(t - t_0) f(t) dt = f(t_0)$$

Usualmente  $\delta(t - t_0)$  representa un impulso en la unidad de área de  $t = t_0$ .

Una Función Delta con posiciones  $x_n = nx_0$  con  $n = 0, \pm 1, \pm 2, \dots$  puede ser representada por la serie ,

$$\sum_{n=-\infty}^{+\infty} \delta(x - x_0) = 1/x_0 \sum_{n=-\infty}^{+\infty} e^{\pm j2\pi nx/x_0}$$

ESTA TESIS NO DEBE  
SALIR DE LA BIBLIOTECA

BIBLIOGRAFIA

E. Oran Brigham, The Fast Fourier Transform and its Applications, Prentice-Hall Signal Processing Serie, paginas 448 ,1988.

Ingrid Daubechies, The Wavelet Transform, Time-Frequency Localization and Signal Analisis, Revista IEEE Transactions on Information Theory, volumen 36 No 5 paginas 961-1005, 1990.

Herbert Schildt, Programación en Turbo C, Segunda Edición, Borland-Osborne/McGraw-Hill, paginas 447 Febrero de 1991.

Herbert Schildt, C Guía para usuarios expertos, Osborne/McGraw-Hill, paginas 358 Julio de 1990.

Setphane G. Mallat, Multifrequency chanel decompositions of images and wavelets models, Revista IEEE Transactions on Acustics, Speech and Signal Processing volumen 37 No 12 paginas 2091-2110 Diciembre de 1989.

R.J. Higgins, Fast Fourier transform: An introduction with some minicomputer experiments, Revista American Journal of Physics volumen 44 No 8 paginas 766-774 Agosto de 1976.