

# Estudio de la interacción de la luz con el ultrasonido, a través del fenómeno de difracción descrito por el modelo de Raman y Nath

Luis E. Aragón<sup>1</sup> - Iván A. Davidovich<sup>2</sup>  
Instituto Balseiro  
8400 Bariloche

Fecha de entrega: 11 de marzo de 2009

## Resumen

En este trabajo, se estudió la interacción de la luz con el ultrasonido. En particular, se analizó el fenómeno de difracción producido cuando un rayo de luz atraviesa un medio, en este caso agua o glicerina, en el cual se propagan ondas de ultrasonido, en condiciones acordes al modelo planteado por Raman y Nath. Se observó un buen acuerdo entre los valores medidos, de la intensidad de los máximos de difracción de cada orden en función de la potencia de las ondas sonoras y, las predicciones del modelo. Se encontró a su vez un comportamiento inesperado de uno de los parámetros del modelo, lo cual constituye un tópico interesante a investigar en futuras realizaciones de este experimento.

## Introducción

Las ondas de sonido generan compresiones y rarefacciones del medio en el que se propagan cambiando periódicamente su densidad y por ende su índice de refracción.

Cuando este medio es transparente y sobre él, además inciden perpendicularmente ondas de luz coherente, los cambios periódicos del índice de refracción actúan como una red de difracción de fases. Esto produce un patrón de máximos y mínimos de intensidad cuya descripción fue modelada por Raman y Nath [1]. Esta situación se esquematiza en la **Figura 1**.

La teoría desarrollada por Raman y Nath predice que la amplitud de la onda de luz modulada en fase se puede expandir en término de las funciones de Bessel,  $J_q$ , donde cada orden  $q$  tiene una intensidad  $I_q$  proporcional a  $J_q^2$ , de manera que,

$$I_q = k J_q^2 (m \sqrt{P_s}) \quad (1)$$

---

<sup>1</sup> E-mail: aragonl@ib.cnea.gov.ar

<sup>2</sup> E-mail: davidovi@ib.cnea.gov.ar

donde  $P_s$  es la potencia de la onda sonora y,  $k$  y  $m$  son constantes de ajuste. Según este modelo,  $m$  depende de la disposición general del experimento y no debiera cambiar al variar  $q$ .

En este trabajo, la onda sonora fue emitida por un piezoeléctrico a través del cual circulaban corrientes bajas. Esto indica que la potencia aplicada fue proporcional al cuadrado de la tensión  $V$  sobre el piezoeléctrico. Suponiendo que la transducción de potencia eléctrica a sonora es lineal, se tiene que  $P_s = \alpha V^2/R$ , siendo  $R$  la resistencia del piezoeléctrico y  $\alpha$  una constante de proporcionalidad. Entonces, en este caso particular la ecuación (1) se puede escribir como

$$I_q = k J_q^2 (h V) \quad (2)$$

con  $h = m \sqrt{(\alpha/R)}$ .

Como las funciones de Bessel cumplen con la relación  $J_q^2 = J_{-q}^2$ , se predice que el patrón de difracción debe ser simétrico respecto del rayo central. Además, es interesante notar que a diferencia de la difracción usual de la luz, esta ecuación predice que el máximo central ( $q=0$ ) se puede anular cuando se anule la  $J_0$ .

El objetivo de este trabajo es observar el fenómeno de difracción descrito por este modelo y estudiar el comportamiento de la intensidad de los máximos del patrón al variar la potencia de la onda sonora en agua y en glicerina.

## Método Experimental

Para producir el fenómeno buscado, utilizamos una cubeta metálica de unos **66mm** de ancho, sin tapa y con dos ventanas en sus costados. Llenamos la cubeta con agua para realizar las primeras series de mediciones y con glicerina para las últimas. En la pared más alejada de la cubeta se ubicaba un piezoeléctrico. Este actuaba como transductor, transformando la señal eléctrica que se le entregaba en un movimiento periódico, que producía las ondas de ultrasonido en el fluido. La señal que llegaba al piezoeléctrico provenía de un generador diseñado para trabajar en frecuencias del orden de **1MHz**. La potencia entregada por este se controlaba a través de un autotransformador. La tensión entregada por el generador al piezoeléctrico se midió con un osciloscopio, conectado en paralelo. La introducción de este último en el circuito no introdujo variaciones apreciables a simple vista en el patrón de difracción. Se buscó que las ondas en el fluido viajaran en una dirección paralela a las ventanas de la cubeta y se ubicó una esponja dentro de la misma, en el extremo opuesto al piezoeléctrico, para evitar la formación de ondas estacionarias de ultrasonido. Se hizo entonces incidir, de forma normal a las ventanas, un haz de luz láser previamente ensanchado mediante el uso de un telescopio. Esta luz era roja, de longitud de onda **(632,8±0,1)nm**. Luego de atravesar la cubeta, esta luz era recogida por un segundo telescopio y proyectada en una pared a unos **3,3m**. Se niveló y alineo este sistema óptico, dentro de lo permitido por los telescopios. Estos resultaban muy pesados, lo que dificultaba mantenerlos a nivel, e introducían muchas reflexiones internas. Finalmente, para el relevamiento de los patrones de difracción se usó un fotodetector (cuyo esquema circuital

se muestra en la **Figura 2)** montado sobre el carro de un graficador de tipo xy, que permitía realizar barridos de lectura<sup>3</sup>, y conectado a un multímetro HP3478A que transfería los valores de voltaje medidos a una PC a través de una interfaz de tipo IEEE 488.2. El código fuente del programa utilizado para la adquisición de datos se presenta en el **Apéndice** de este informe. En las **Figuras 3 a 6** puede verse la disposición general del equipo experimental empleado, así como una vista más detallada de distintas partes del mismo.

Para la correcta obtención de mediciones el equipo debió someterse a sucesivos ajustes. Uno de los principales problemas encontrados fue la inestabilidad de los primeros generadores de ultrasonido utilizados. Por otro lado, el piezoeléctrico usado no se encontraba fijo a la pared de una manera que le impidiese rotar sobre su eje vertical, por lo que su dirección no era totalmente paralela a dicha cara de la cubeta. Esto resultó en que la cubeta como conjunto debiese rotarse ligeramente para observar el fenómeno de difracción apropiadamente, saliendo de la perpendicularidad inicial entre esta y el haz láser. Según se observó, la alineación del ultrasonido es un factor importante a tener en cuenta en el armado del experimento.

A partir de aquí se comenzaron a realizar barridos del fotodetector, para relevar patrones, a la vez que se medía la tensión entregada al piezoeléctrico. Con la cubeta llena de agua se midieron pares de patrones con cada una de 12 tensiones distintas aplicadas al piezoeléctrico. Luego se vació la cubeta y se la dejó secar, para entonces llenarla de glicerina. Con este fluido se midió aplicando 11 tensiones distintas al piezoeléctrico. Con cada una de estas tensiones se relevaron 3 espectros. Se trabajó siempre a una frecuencia de **(910±90)kHz**.

De los espectros relevados, se obtuvieron valores relativos de intensidad de cada pico, a través de un proceso de normalización. Dicho proceso consistió en calcular el área de cada pico y dividirla por el área total de su patrón correspondiente. Como herramienta para acelerar el procesamiento de estos datos se utilizó un programa de desarrollo propio cuyo código fuente se presenta en el **Apéndice**.

Una vez procesados todos los datos medidos, se procedió al ajuste de los mismos según **(2)**.

## **Resultados y Discusión**

En las **Figuras 7 a 14** se muestra una serie de los mejores espectros relevados utilizando glicerina. Los valores de intensidad en función de la tensión aplicada al piezoeléctrico, correspondientes a cada orden, pueden verse junto con sus curvas de mejor ajuste en las **Figuras 15 a 21**. Los parámetros de ajuste **h** y **k** de cada orden y con cada sustancia se presentan en las **Tablas 1 y 2**. Se ajustó pidiendo que, para ordenes con el mismo valor absoluto y de la misma sustancia, **h** fuese el mismo.

Como puede verse, en las gráficas de intensidad como función de la tensión aplicada al piezoeléctrico, hay cierta cantidad de puntos

---

<sup>3</sup> Como resultado de una calibración se obtuvo el valor **(11,942±0,006)s/cm** correspondiente a la inversa de la velocidad de barrido del carro del graficador.

experimentales que no son tocados por su curva de ajuste. Hay dos causas importantes que podrían explicar esto. Una la constituyen las fluctuaciones en la intensidad del láser, que fueron observadas durante la experiencia. Si bien esto fue remediado en gran parte al utilizar un láser de mejor calidad, las fluctuaciones persistieron. Una idea que podría ponerse en práctica en futuras realizaciones de este experimento, a los fines de eliminar el efecto de este fenómeno sobre los datos, sería el incorporar un divisor de haz a la salida del láser y medir la intensidad del nuevo rayo con un segundo detector. Se podría entonces dividir cada lectura del detector encargado de medir el patrón de difracción, por la lectura correspondiente de este segundo detector y, de esa manera, eliminar los problemas causados por fluctuaciones aleatorias en la intensidad del láser. Claro está que, esta idea sería poco efectiva si las fluctuaciones mencionadas fuesen muy rápidas. La segunda posible causa del desajuste de los puntos antes mencionados, podría estar en una cierta inestabilidad del generador de ultrasonido a altas potencias. Esto fue observado durante el experimento con los primeros generadores utilizados, en la forma de pérdidas periódicas del patrón de difracción o de cambios visibles a simple vista en su intensidad. Reemplazando algunas partes del generador se logró que el mismo dejara de manifestar este comportamiento, al menos a simple vista. Puede ser, sin embargo, que estas inestabilidades persistieran a menor escala.

Cuando se midió el voltaje aplicado el piezoeléctrico se encontró que la señal enviada al mismo presentaba un cierto “espesor”, es decir, que mostraba oscilaciones en una segunda frecuencia más baja. La fuente de este ruido no pudo ser identificada, si bien puede estar relacionada con las inestabilidades ya mencionadas en el generador. Este ruido se vio reflejado en los intervalos de error asociadas al voltaje aplicado al piezoeléctrico.

Se observó a su vez cierta asimetría para algunos patrones correspondientes a órdenes de un mismo valor absoluto. Esto se debió con buena probabilidad a reflexiones internas en los telescopios. Sería recomendable, en futuras realizaciones de este experimento, que los telescopios fuesen reemplazados por pares de lentes, montadas sobre la guía de elementos ópticos. Se intentó trabajar de esa manera en este trabajo, pero se volvió finalmente a los telescopios, por falta de tiempo para optimizar el sistema óptico.

Más allá de los distintos problemas mencionados, el modelo propuesto por Raman y Nath ajusta en un 72% de los puntos medidos con agua y en un 97% de los puntos medidos con glicerina. Consideramos que estos resultados representan un buen ajuste. Además, se observó que las características generales del fenómeno concuerdan bien con lo predicho por el modelo.

Cabe hacer un comentario más sobre los ajustes y es en lo referido al parámetro  $h$ . Como puede verse en las **Tablas 1 y 2**,  $h$  presenta un crecimiento monótono con el orden de difracción. Para analizar mejor este crecimiento se decidió graficar este parámetro en función del orden de difracción. Estas representaciones se muestran en las **Figuras 22 y 23**. Según se ve en la primera de ellas,  $h$  muestra un crecimiento que sugiere un comportamiento exponencial con el orden. Esto ya ha sido observado en realizaciones pasadas del experimento [2], lo que da lugar a pensar que  $h$

podría no ser únicamente dependiente de las condiciones generales del experimento, tal y como se creía. Esta es una cuestión sumamente interesante, que podría ser objeto de estudio en futuras realizaciones de este experimento.

## **Conclusiones**

Se estudió el fenómeno de difracción de la luz por un medio en el cual se propagan ondas de ultrasonido, dentro de las condiciones del modelo de Raman y Nath. Se observó un comportamiento de la intensidad de los distintos órdenes como función de la potencia de las ondas sonoras acorde a lo predicho por el modelo. El fenómeno pudo observarse correctamente usando como medio para la propagación del sonido tanto agua como glicerina. Se detectó a su vez un comportamiento inesperado de uno de los parámetros del modelo, que ya había sido observado en el pasado. El origen de este comportamiento es aún desconocido y constituye un interesante objeto de estudio para futuras realizaciones del experimento.

## **Referencias**

- [1] D.T.Pierce y R.L Byer, *Am.J.Phys.* 41, 314 (1973).
- [2] D.D.Blanco y J.M.Lorenzi, Estudio de la interacción de la luz con ultrasonido dentro de la aproximación de Raman-Nath, *Física Experimental II*, Instituto Balseiro, 2008.

## Tablas

Parámetros de ajuste para el Agua				
Orden de difracción	k	$\Delta k$	$h[V^{-1}]$	$\Delta h[V^{-1}]$
0	1,00	0,03	0,0242	0,0006
-1	1,01	0,03	0,0248	0,0005
1	0,88	0,03		
-2	1,1	0,2	0,027	0,002
2	0,8	0,2		
-3	0,33	0,05	0,037	0,003
3	0,20	0,04		

Tabla1. Parámetros de ajuste para los valores medidos de intensidad de los máximos de difracción en función del voltaje aplicado al piezoelectrico, según (2), en agua.

Parámetros de ajuste para la Glicerina				
Orden de difracción	k	$\Delta k$	$h[V^{-1}]$	$\Delta h[V^{-1}]$
0	0,92	0,03	0,0186	0,0005
-1	0,90	0,02	0,0212	0,0004
1	0,98	0,02		
-2	0,52	0,03	0,0256	0,0007
2	0,51	0,03		

Tabla2. Parámetros de ajuste para los valores medidos de intensidad de los máximos de difracción en función del voltaje aplicado al piezoelectrico, según (2), en glicerina.

## Figuras

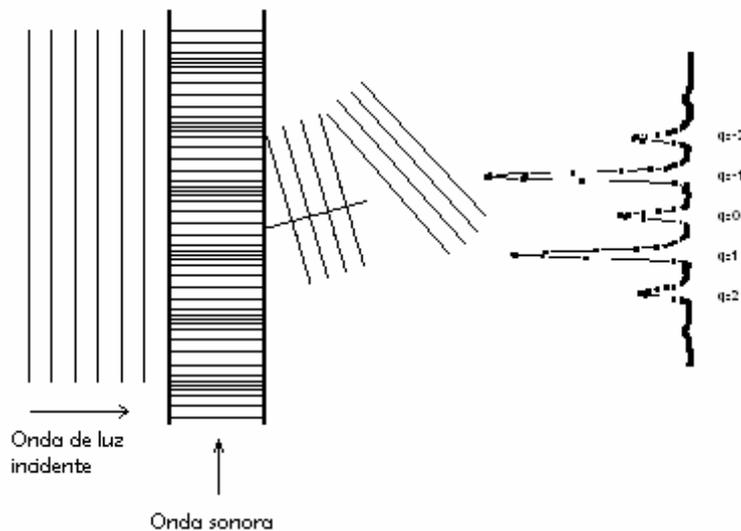


Figura1. Esquema de la red de difracción de fases que se forma cuando una onda de luz coherente incide perpendicularmente sobre ondas sonoras. Se puede apreciar la desviación del frente de ondas de luz y un grafico de los máximos de intensidad de orden q.

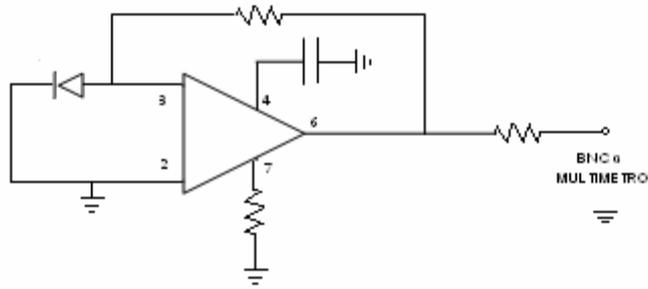


Figura 2. Esquema circuital del fotodiodo utilizado para detectar el haz de luz difractado.



Figura 3. Fotografía de la configuración experimental donde se puede apreciar los elementos utilizados: PC, Detector, Multímetro, Telescopios, Cubeta, Láser, Autotransformador y Osciloscopio.



Figura 4. Fotografía del patrón de difracción y del fotodetector montado sobre el graficador xy.

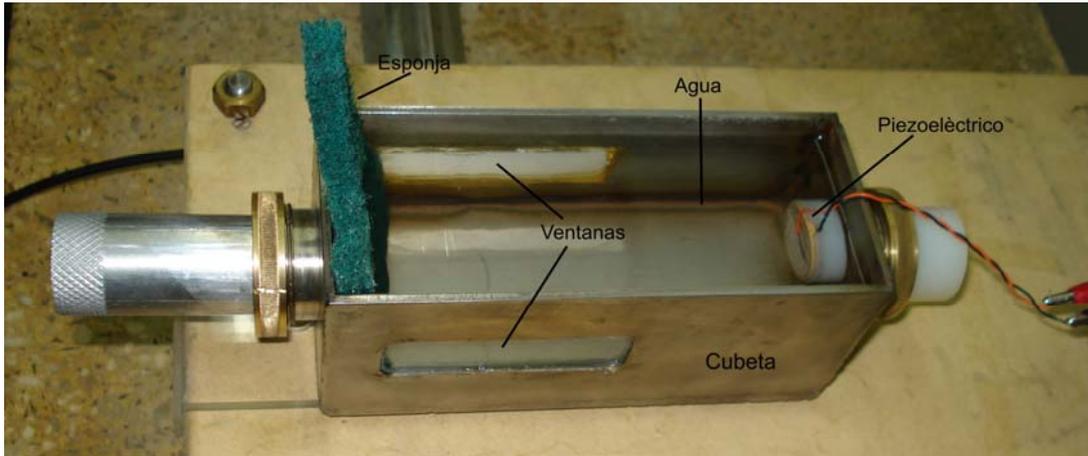
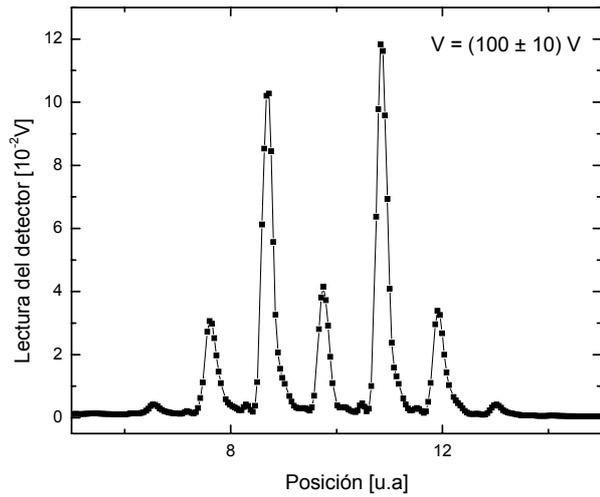


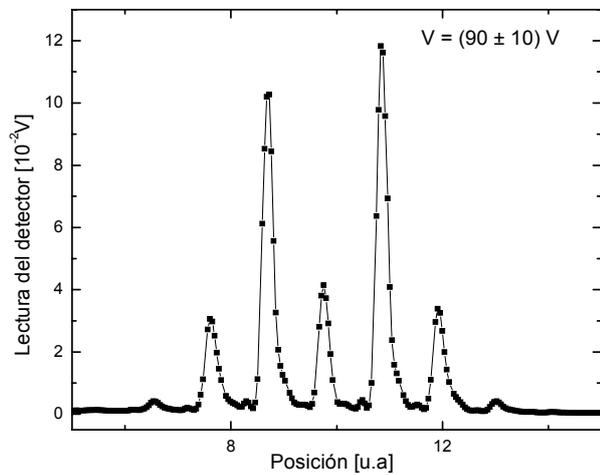
Figura 5. Detalles de la cubeta llena de agua. Se pueden apreciar las ventanas por donde atravesaba el haz de luz, el piezoeléctrico que transmitía las ondas sonoras, y la esponja utilizada para evitar ondas estacionarias.



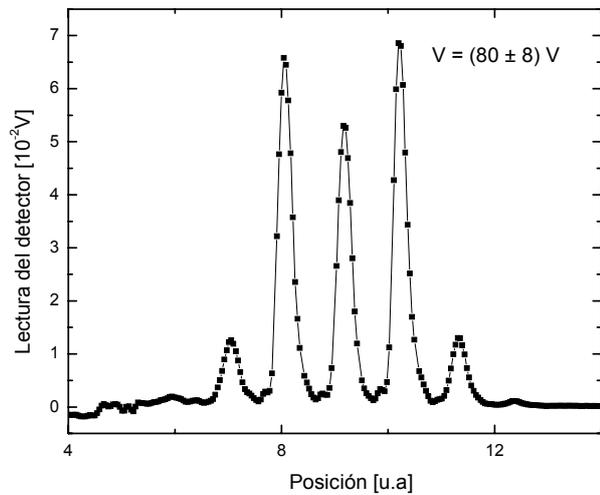
Figura 6. Vista detallada de la configuración experimental donde se puede apreciar el autotransformador, el láser, la fuente de ultrasonido, el telescopio y la cubeta.



**Figura 7.** Lectura del fotodetector en función de la posición del mismo con un voltaje aplicado al piezoeléctrico de  $(100 \pm 10) V$  en glicerina.



**Figura 8.** Lectura del fotodetector en función de la posición del mismo con un voltaje aplicado al piezoeléctrico de  $(90 \pm 10) V$  en glicerina.



**Figura 9.** Lectura del fotodetector en función de la posición del mismo con un voltaje aplicado al piezoeléctrico de  $(80 \pm 10) V$  en glicerina.

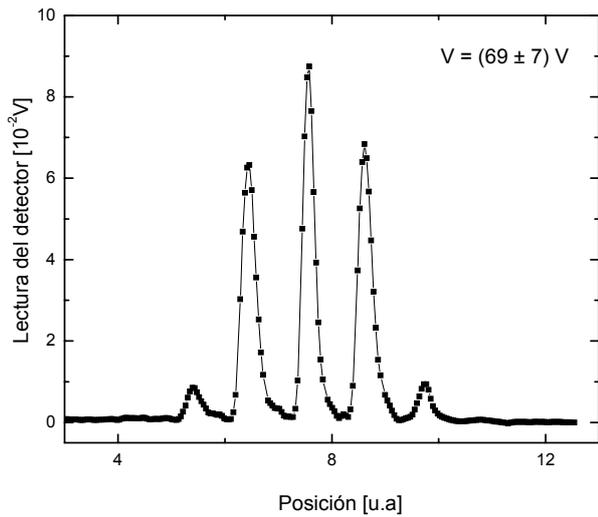


Figura 10. Lectura del fotodetector en función de la posición del mismo con un voltaje aplicado al piezoeléctrico de  $(69 \pm 7) V$  en glicerina.

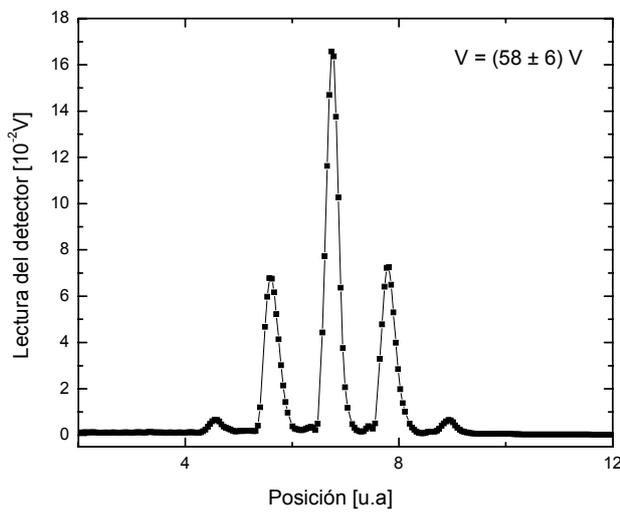


Figura 11. Lectura del fotodetector en función de la posición del mismo con un voltaje aplicado al piezoeléctrico de  $(58 \pm 6) V$  en glicerina.

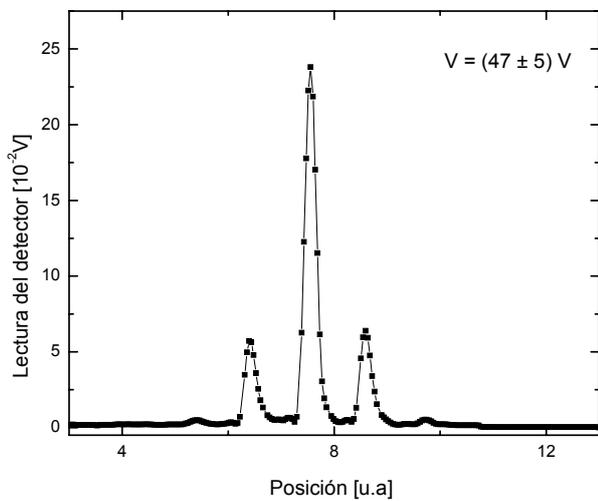


Figura 12. Lectura del fotodetector en función de la posición del mismo con un voltaje aplicado al piezoeléctrico de  $(47 \pm 5) V$  en glicerina.

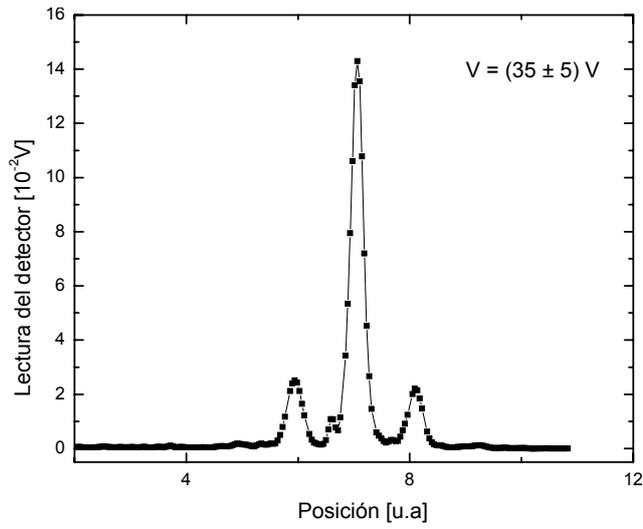


Figura 13. Lectura del fotodetector en función de la posición del mismo con un voltaje aplicado al piezoeléctrico de  $(35 \pm 5) V$  en glicerina.

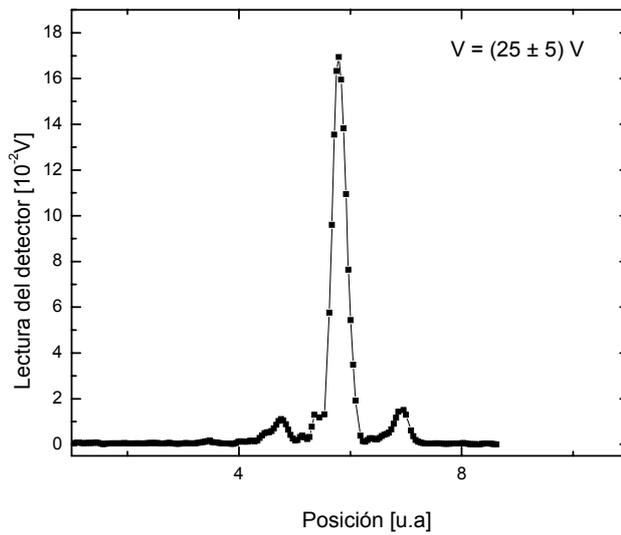


Figura 14. Lectura del fotodetector en función de la posición del mismo con un voltaje aplicado al piezoeléctrico de  $(25 \pm 5) V$  en glicerina.

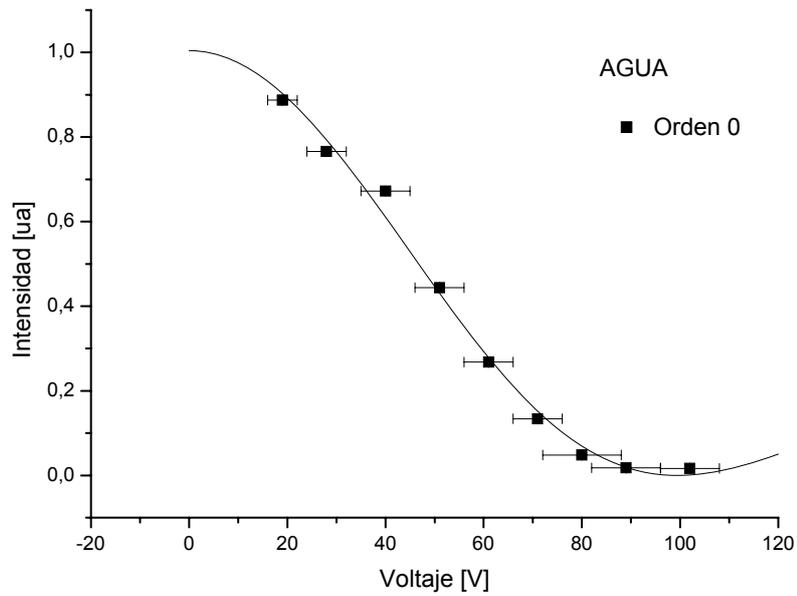


Figura 15. Intensidad del máximo central de difracción de orden  $q=0$  ( $I_0$ ) en función del voltaje (V) aplicado al piezoeléctrico en agua. La curva es un ajuste teniendo en cuenta la ecuación  $I_0 = k J_0^2 (h \sqrt{V})$ .  $h_0 = (0.0242 \pm 0.0006) V^{-1}$ ;  $k_0 = 1.00 \pm 0.03$ .

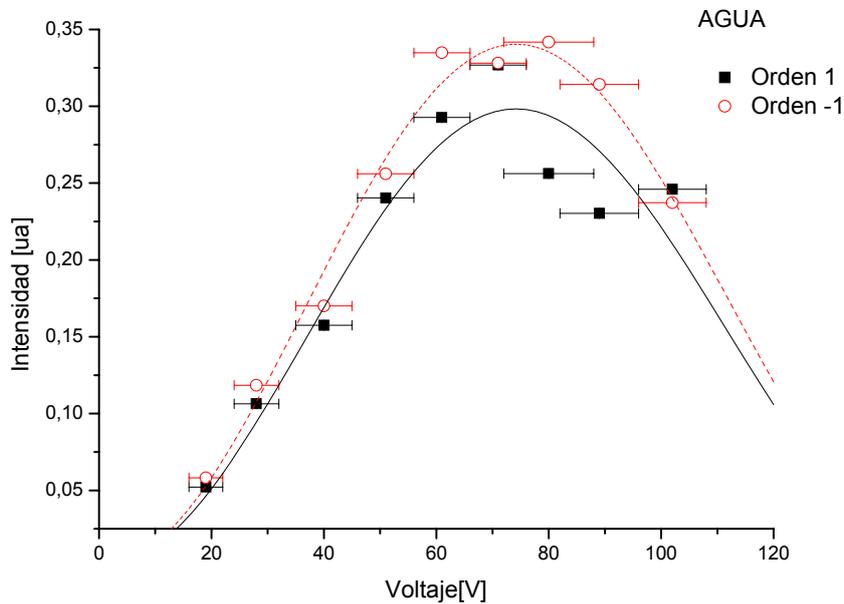


Figura 16. Intensidad de los máximos de difracción de orden  $q=1$  y  $q=-1$  ( $I_1$  y  $I_{-1}$ ) en función del voltaje (V) aplicado al piezoeléctrico en agua. Los círculos vacíos son los valores medidos para  $q=-1$  y los cuadrados negros son los valores medidos para  $q=1$ . La curva de trazos y la curva continua son un ajuste teniendo en cuenta la ecuación  $I_q = k J_q^2 (h \sqrt{V})$  con  $q=-1$  y  $q=1$  respectivamente.  $h_1 = h_{-1} = (0.0242 \pm 0.0006) V^{-1}$ ;  $k_1 = 0,88 \pm 0.03$ ;  $k_{-1} = 1.01 \pm 0.03$ .

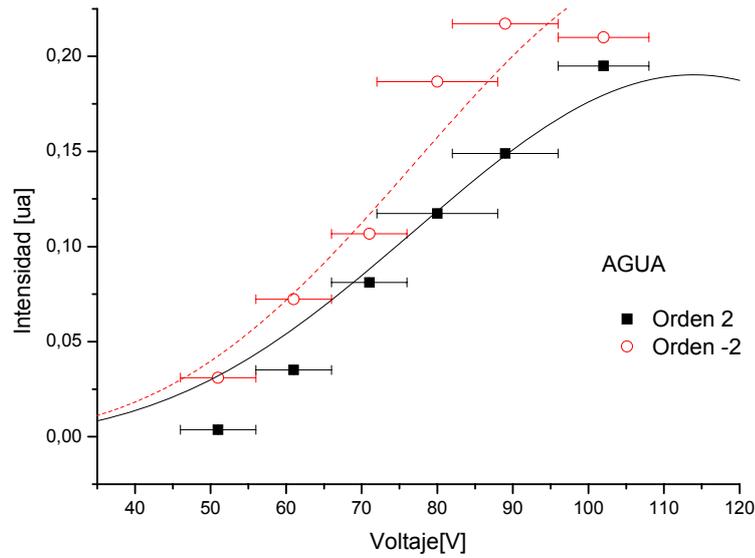


Figura 17. Intensidad de los máximos de difracción de orden  $q=2$  y  $q=-2$  ( $I_2$  y  $I_{-2}$ ) en función del voltaje (V) aplicado al piezoeléctrico en agua. Los círculos vacíos son los valores medidos para  $q=-2$  y los cuadrados negros son los valores medidos para  $q=2$ . La curva de trazos y la curva continua son un ajuste teniendo en cuenta la ecuación  $I_q = k J_q^2 (h \sqrt{V})$  con  $q=-2$  y  $q=2$  respectivamente.  $h_2 = h_{-2} = (0.027 \pm 0.002) V^{-1}$ ;  $k_2 = 0,8 \pm 0.2$ ;  $k_{-2} = 1.1 \pm 0.2$ .

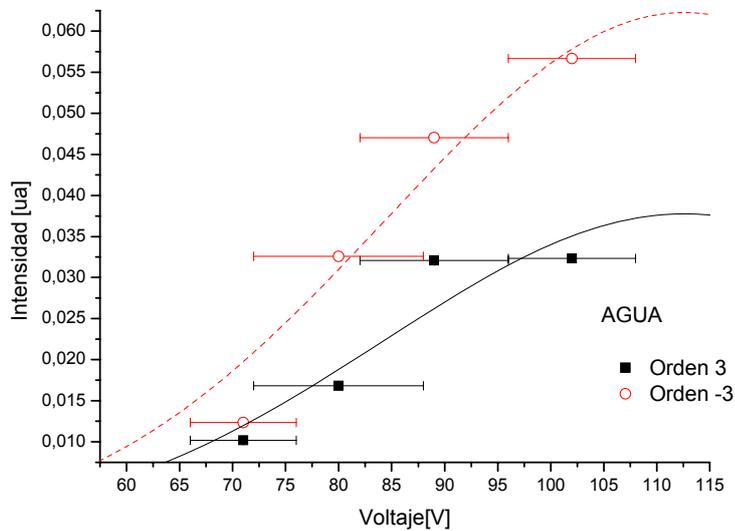


Figura 18. Intensidad de los máximos de difracción de orden  $q=3$  y  $q=-3$  ( $I_3$  y  $I_{-3}$ ) en función del voltaje (V) aplicado al piezoeléctrico en agua. Los círculos vacíos son los valores medidos para  $q=-3$  y los cuadrados negros son los valores medidos para  $q=3$ . La curva de trazos y la curva continua son un ajuste teniendo en cuenta la ecuación  $I_q = k J_q^2 (h \sqrt{V})$  con  $q=-3$  y  $q=3$  respectivamente.  $h_3 = h_{-3} = (0.037 \pm 0.003) V^{-1}$ ;  $k_3 = 0,20 \pm 0.05$ ;  $k_{-3} = 0,33 \pm 0.04$ .

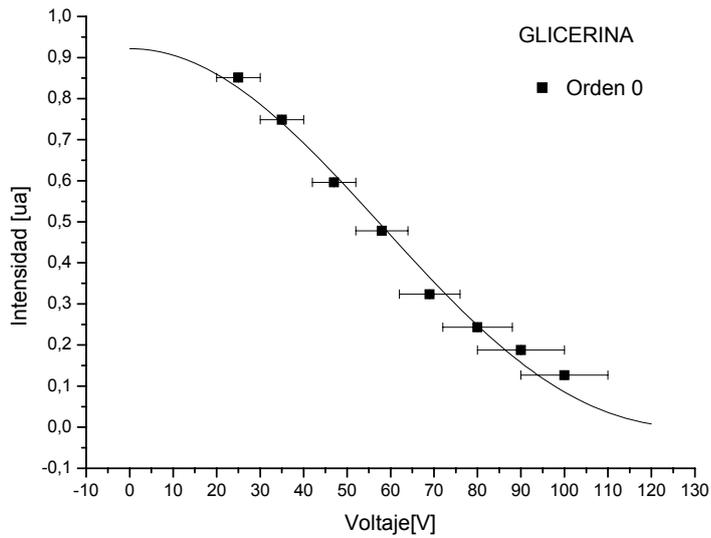


Figura 19. Intensidad del máximo central de difracción de orden  $q=0$  ( $I_0$ ) en función del voltaje (V) aplicado al piezoeléctrico en glicerina. La curva es un ajuste teniendo en cuenta la ecuación  $I_0 = k J_0^2 (h \sqrt{V})$ .  $h_0 = (0,0186 \pm 0,0005) V^{-1}$ ;  $k_0 = 0,92 \pm 0,03$ .

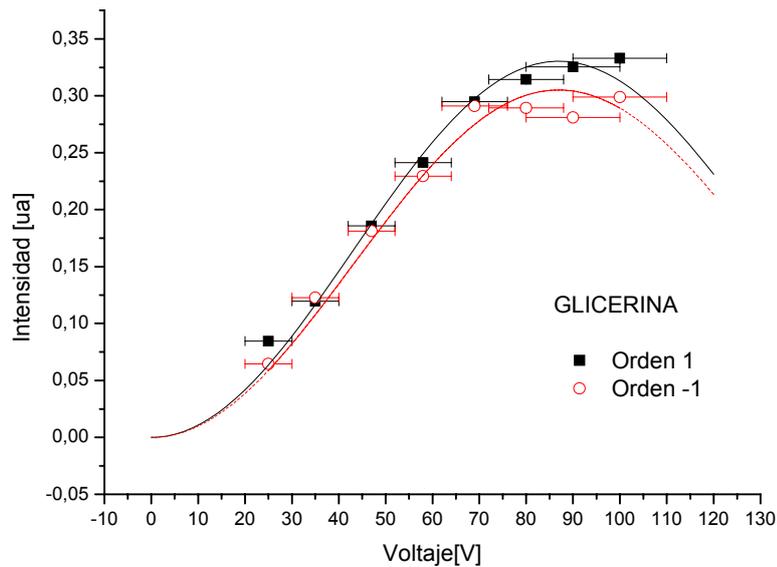


Figura 20. Intensidad de los máximos de difracción de orden  $q=1$  y  $q=-1$  ( $I_1$  y  $I_{-1}$ ) en función del voltaje (V) aplicado al piezoeléctrico en glicerina. Los círculos vacíos son los valores medidos para  $q=-1$  y los cuadrados negros son los valores medidos para  $q=1$ . La curva de trazos y la curva continua son un ajuste teniendo en cuenta la ecuación  $I_q = k J_q^2 (h \sqrt{V})$  con  $q=-1$  y  $q=1$  respectivamente.  $h_1 = h_{-1} = (0,0212 \pm 0,0004) V^{-1}$ ;  $k_1 = 0,98 \pm 0,02$ ;  $k_{-1} = 0,90 \pm 0,02$ .

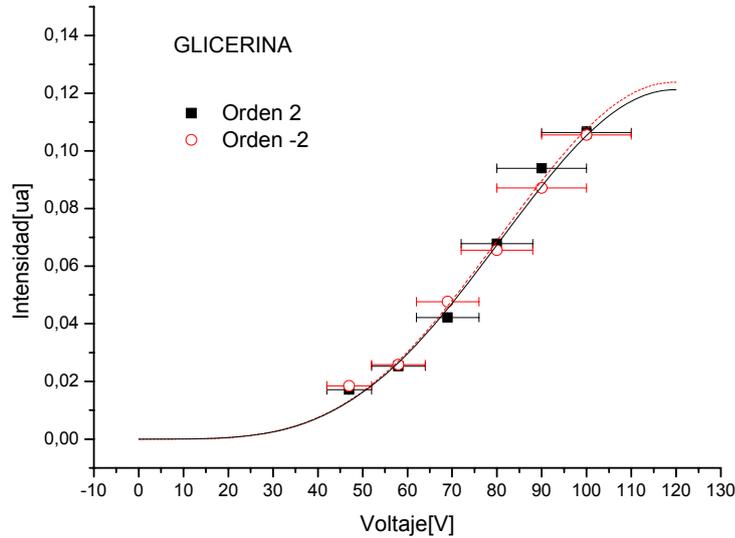


Figura 21. Intensidad de los máximos de difracción de orden  $q=2$  y  $q=-2$  ( $I_2$  y  $I_{-2}$ ) en función del voltaje (V) aplicado al piezoeléctrico en glicerina. Los círculos vacíos son los valores medidos para  $q=-2$  y los cuadrados negros son los valores medidos para  $q=2$ . La curva de trazos y la curva continua son un ajuste teniendo en cuenta la ecuación  $I_q = k J_q^2 (h \sqrt{V})$  con  $q=-2$  y  $q=2$  respectivamente.  $h_2 = h_{-2} = (0,0256 \pm 0,007) V^{-1}$ ;  $k_2 = 0,51 \pm 0,03$ ;  $k_{-2} = 0,52 \pm 0,03$ .

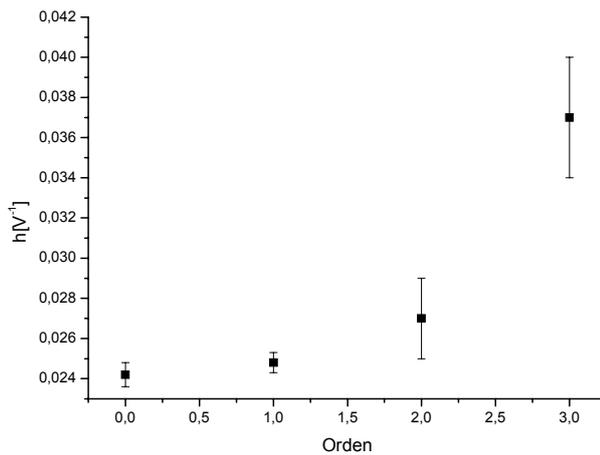


Figura 22. Parámetro de ajuste  $h$  de la ecuación  $I_q = k J_q^2 (h \sqrt{V})$  en función del valor absoluto del orden del máximo de difracción  $q$  en agua.

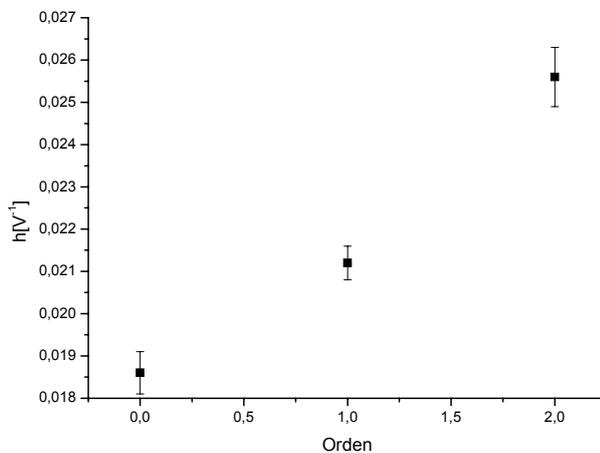


Figura 23. Parámetro de ajuste  $h$  de la ecuación  $I_q = k J_q^2 (h \sqrt{V})$  en función del valor absoluto del orden del máximo de difracción  $q$  en glicerina.

## Apéndice

Programa empleado para la adquisición de los datos medidos por el voltímetro HP3478A, a través de una interfaz de tipo IEEE488.2 (programado en Basic)

**Nota:** Incluir módulo iee488w.bas

```
Dim X(2000) As Double, Y(2000) As Double, iniT As Double, Ndata As Integer,  
Midiendo As Boolean
```

---

```
Private Function ReadV(Volt%) As Double
```

```
    'Genera buffer  
    pp$ = Space$(40)  
    'Maximo numero de c\bytes a leer  
    cnt% = 40  
    'Lectura  
    ret% = IeEnter(Volt%, 0, pp$, cnt%)  
    If ret% <> 0 Then ret% = MsgBox("Error GPIB " & Format(ret%)): End  
    'Convierte a valor numerico  
    ReadV = Val(Left$(pp$, 11))
```

```
End Function
```

---

```
Private Sub cmdGuardar_Click()
```

```
    With cmdIlgSave  
        .DefaultExt = ".dat"  
        .Filter = "Data Files (*.dat)|*.dat|Text Files (*.txt)|*.txt|All Files (*.*)|*.*"  
        .ShowSave  
        If .FileName <> "" Then  
            Open .FileName For Output As #1  
            For i = 1 To Ndata  
                Print #1, X(i), Y(i)  
            Next i  
            Close #1  
            ret% = MsgBox("Datos guardados!!!")  
        End If  
    End With
```

```
End Sub
```

---

```
Private Sub cmdMedir_Click()
```

```
    iniT = Timer  
  
    Midiendo = Not (Midiendo)  
    If Midiendo Then  
        cmdMedir.Caption = "Parar de Adquirir"  
        tmrMedir.Enabled = True  
    Else
```

```
cmdMedir.Caption = "Empezar a Adquirir"  
tmrMedir.Enabled = False  
End If
```

```
End Sub
```

---

```
Private Sub cmdReset_Click()
```

```
ret% = MsgBox("Va a perder los datos!!! Esta seguro?", vbOKCancel)  
If ret% = 1 Then  
Ndata = 0  
NTGraph.ClearGraph  
End If
```

```
End Sub
```

---

```
Private Sub Form_Load()
```

```
'Inicializa la placa IEEE  
ret% = IeInit(0, 0, 0, 0)  
  
'Lee un valor de cada voltmetro  
' V = ReadV(702)  
V = ReadV(707)  
  
Midiendo = False  
tmrMedir.Enabled = False  
tmrMedir.Interval = 300
```

```
End Sub
```

---

```
Private Sub NTGraph_CursorPosition(ByVal X As Double, ByVal Y As Double)
```

```
lblCoord.Caption = "X = " & Format(X) & Space$(5) & "Y = " & Format(Y)
```

```
End Sub
```

---

```
Private Sub tmrMedir_Timer()
```

```
'Lee un valor de Voltaje  
  
'Vx = ReadV(702)  
  
Vy = 100 * ReadV(707)  
' Vx = Ndata  
Vx = (Timer - iniT) / Val(txtSweep.Text)  
  
Ndata = Ndata + 1  
X(Ndata) = Vx  
Y(Ndata) = Vy  
lblData.Caption = Format(Ndata) & Space$(10) & Format(Vx) & Space$(10) &  
Format(Vy)  
'Reconstruye el grafico  
With NTGraph
```

```

.ClearGraph
.ElementLineColor = vbGreen
.ElementPointColor = vbGreen
.ElementPointSymbol = Dots
.ElementLinetype = 5
.ElementWidth = 2

'Grafica en funcion de V en microvolts
For i = 1 To Ndata
    .PlotXY X(i), Y(i), 0
Next i
'Hace autorango para las escalas
.AutoRange
End With

End Sub

```

### Programa empleado para el procesamiento de datos (programado en C++)

**Nota:** Este programa sufrió importantes modificaciones durante su programación y es la última versión de un proceso de desarrollo. Es debido a los cambios sufridos que puedan tal vez encontrarse en él líneas cuyo contenido es obsoleto. Se pudo haber eliminado dichas partes del código por estética, optimización o claridad, pero se optó en cambio por presentar el código del programa tal y como fue utilizado para el procesamiento de los datos.

```

// ProcesRamanu.cpp : Defines the entry point for the console
application.
//

#include "stdafx.h"
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

struct Espectro {
    char *ofilename;
    double voltaje;
    double errvoltaje;
    int peaknum;
    double totarea;
    double errtotarea;
    struct Pico *peak;
};
struct Pico {
    int order;
    double starting;
    double ending;
    double abspeakarea;
    double errabspeakarea;
    double relpeakarea;
    double errrelpeakarea;
};

```

```

};

double **AllocMat(int nFilas, int nColumnas);
void ImprimeMatriz(double **Mat, int N);
void LiberaMat(double **Mat, int i);
double ** FLoad(char filename[], int *n, struct Espectro *spectre);
int FProcess(struct Espectro *spectre, double **datos, int n);
int CalculaArea(struct Espectro *spectre, double **datos, int ini, int
end, int peakn);
int FSave(struct Espectro *spectre, char filename[]);
int Promediar(struct Espectro **spectre, int current, int count, char
**filename);
//void Ordena(struct Espectro *spectre);

int main(int argc, char *argv[])
{
    int *n,i,count,j=0;
    double **datos;
    struct Espectro **spectre;
    FILE *g;

    /*argc=11;
    argv[1]="stest.dat";
    argv[2]="OrdenM3.dat";
    argv[3]="OrdenM2.dat";
    argv[4]="OrdenM1.dat";
    argv[5]="Orden0.dat";
    argv[6]="Orden1.dat";
    argv[7]="Orden2.dat";
    argv[8]="Orden3.dat";
    argv[9]="V8.dat";
    argv[10]="V3.dat";*/

    n = (int *)malloc(sizeof(int));
    if ((spectre = (struct Espectro**)malloc((argc-2)*sizeof(struct
Espectro*)))==NULL)
    {
        perror ("Error al aloca el Espectro\n");
        return -1;
    }

    for (i=0;i<argc-9;i++)
    {
        spectre[i] = (struct Espectro*)malloc(sizeof(struct
Espectro));
        if (spectre == NULL)
        {
            perror ("Error al aloca Espectro\n");
            return -1;
        }
    }

    g=fopen(argv[1],"a+");
    fprintf(g,"Voltaje\t\tIntensidad\tErrorV\t\tErrorI\t\tNombre\t\t
Tolerancia\tInicio\t\tFin\t\tOrden\n");
    fclose(g);

    for (i=2;i<9;i++)
    {

```

```

        g=fopen(argv[i], "a+");
        fprintf(g, "Voltaje\t\tIntensidad\tErrorV\t\tErrorI\n");
        fclose(g);
    }

    for (i=9;i<argc;i++)
    {

        if ((datos = FLoad(argv[i],n,spectre[i-9]))==NULL)//Cargo
los valores en datos y de paso chequeo que ande todo bien
        {
            perror ("Error al cargar el archivo\n");
            return -2;
        }

        spectre[i-9]->ofilename = argv[i];
        spectre[i-9]->peaknum = 0;

        if (FProcess(spectre[i-9],datos,*n)//Proceso los datos y
veo que esté todo bien
        {
            perror ("Error de procesamiento\n");
            return -3;
        }

        if (FSave(spectre[i-9],argv[1])//Se guardan los datos
        {
            perror("Error al salvar los resultados\n");
            return -4;
        }
        if (Promediar(spectre,i-9,j,argv))
            j=0;
        else j++;
    }
    LiberaMat(datos,2);
    for (i=0;i<argc-9;i++)
        free(spectre[i]);
    free(spectre);
    free(n);
    getchar();
    getchar();
    return 0;
}

double** FLoad(char filename[], int *n, struct Espectro *spectre)
{
    int i;
    double **datos;
    FILE *f;

    printf ("Iniciando conteo de datos en %s\n",filename);
    if((f=fopen(filename,"r")) == NULL)//Se abre el archivo y se
verifica que la operación se haya llevado a cabo correctamente. En mi
caso el archivo está en la misma carpeta de proyecto que este .c
    {
        perror("No pudo allocarse el puntero para abrir el
archivo\n");
        return NULL;
    }
}

```

```

        fscanf(f, "%lf\t%lf");

        for (i=0;fscanf(f, "%lf\t%lf")!=EOF;i++);//Hago una pasada por el
archivo para contar el número de líneas

        rewind(f);

        datos = AllocMat(2,i);

        *n = i;

        printf ("Iniciando copiado de datos de %s\n",filename);

        fscanf(f, "%lf\t%lf",&spectre->voltaje,&spectre->errvoltaje);

        for
(i=0;fscanf(f, "%lf\t%lf",&datos[0][i],&datos[1][i])!=EOF;i++); //Cargo
los valores

//    ImprimeMatriz(datos,*n);

        fclose(f);

        printf ("Finalizado copiado de datos de %s\n",filename);

        return datos;
}

double **AllocMat(int nFilas, int nColumnas)
{
    double **Mat;
    int i;

    if((Mat = (double**) malloc(nFilas * sizeof(double *))) == NULL)
        return NULL;

    for(i=0; i<nFilas; i++)
        if((Mat[i] = (double *) malloc(nColumnas *
sizeof(double)))==NULL) {
            LiberaMat(Mat,i);
            return NULL;
        }
    return Mat;
}

void ImprimeMatriz(double **Mat,int N)
{
    int j;
    for (j=0;j<N;j++)
    {
        printf ("%lf\t",Mat[0][j]);
        printf ("%lf\n",Mat[1][j]);
    }
}

void LiberaMat(double **Mat,int i)
{
    int k;
    for (k=0;k<i;k++)
        free(Mat[k]);
    free(Mat);
}

```

```

}

int FProcess(struct Espectro *spectre, double **datos, int n)
{
    int i,j,k;
    double deltax=0;

reingreso:
    printf ("Cual es el numero de picos a medir?\n");
    scanf ("%d",&spectre->peaknum);

    if (spectre->peaknum<0)
    {
        printf("El número de picos en %s debe ser mayor a
cero\n",spectre->ofilename);
        goto reingreso;
    }

    printf("Numero de picos = %d\n",spectre->peaknum);
    spectre->peak = (struct Pico *)malloc((spectre-
>peaknum)*sizeof(struct Pico));

    for (i=0;i<spectre->peaknum;i++)
    {
badbeg:
        printf("Ingresar el numero de dato en el que COMIENZA el
pico numero %d\n",i+1);
        scanf ("%d",&j);
        j--;
        if (j>n)
        {
            perror("El numero ingresado es invalido\n");
            goto badbeg;
        }
        spectre->peak[i].starting=datos[0][j];

badend:
        printf("Ingresar el numero de dato en el que TERMINA el
pico numero %d\n",i+1);
        scanf ("%d",&k);
        k--;
        if (k>n)
        {
            perror("El numero ingresado es invalido\n");
            goto badend;
        }
        spectre->peak[i].ending=datos[0][k];
        printf("Cual es el orden de este pico?\n");
        scanf ("%d",&spectre->peak[i].order);
        printf("Calculando area del pico\n");
        if (CalculaArea(spectre,datos,j,k,i))
        {
            printf("Error al calcular el area del pico %d de
izquierda a derecha\n",i+1);
            return 2;
        }
    }

    spectre->totarea=0;
    spectre->errrtotarea=0;
    for (i=0;i<spectre->peaknum;i++)
    {

```

```

        spectre->totarea += spectre->peak[i].abspeakarea;
        spectre->errtotarea += spectre->peak[i].errabspeakarea;
    }

    for (i=0;i<spectre->peaknum;i++)
    {
        spectre->peak[i].relpeakarea = spectre->peak[i].abspeakarea
/ spectre->totarea;
        spectre->peak[i].errrelpeakarea = spectre-
>peak[i].relpeakarea * (spectre->peak[i].errabspeakarea/spectre-
>peak[i].abspeakarea + spectre->errtotarea/spectre->totarea);
    }

// Ordena(spectre);

    return 0;
}

int CalculaArea(struct Espectro *spectre, double **datos, int ini, int
end, int peakn)
{
    spectre->peak[peakn].abspeakarea = datos[1][end]-datos[1][ini];
    if (spectre->peak[peakn].abspeakarea <= 0)
        return 1;
    spectre->peak[peakn].errabspeakarea = ((datos[1][end+1]-
datos[1][ini-1])-(datos[1][end-1]-datos[1][ini+1]))/2;
    return 0;
}

void Ordena(struct Espectro *spectre)
{
    int i;

    for (i=0;i<spectre->peaknum;i++)
    {
        printf("Se encontro un pico que inicia en x=%lf , y termina
en x=%lf\n",spectre->peak[i].starting,spectre->peak[i].ending);
        printf("Que orden se le debe asignar?\n");
        scanf("%d",&spectre->peak[i].order);
    }
}

int FSave(struct Espectro *spectre, char filename[])
{
    int i;
    FILE *f;

    if ((f=fopen(filename,"a+"))==NULL)
    {
        perror("No se pudo allocar el puntero para el archivo de
salida\n");
        return 1;
    }

    for (i=0;i<spectre->peaknum;i++)

        fprintf(f,"%lf\t%lf\t%lf\t%lf\t%s\t\t%lf\t%lf\t%d\n",spectre-
>voltaje,spectre->peak[i].relpeakarea,spectre->errvoltaje,spectre-
>peak[i].errrelpeakarea,spectre->ofilename,spectre-
>peak[i].starting,spectre->peak[i].ending,spectre->peak[i].order);
}

```

```

        fclose(f);

        return 0;
    }

int Promediar(struct Espectro **spectre, int current, int count, char
**filename)
{
    int i,j,orden,digit;
    double prom,errprom,check=0.;
    FILE *f;

pifia:
    printf("Desea promediar los ultimos espectros
ingresados?(0/1)\n");
    scanf("%d",&digit);
    if (digit!=0 & digit!=1)
    {
        printf("Ingrese 1 para promediar o 0 para no hacerlo\n");
        goto pifia;
    }
    if (digit==0)
        return 0;

    /*for (i=current-count;i<current+1;i++)
        check+=spectre[i]->voltaje;
    if (spectre[current]->voltaje!=check/(count+1))
    {
        printf("Los patrones no fueron medidos con una misma Vs,
por lo que no seran promediados\n");
        return 0;
    }

    check=0;

    for (i=current-count;i<current+1;i++)
        check+=spectre[i]->peaknum;
    if (spectre[current]->peaknum!=check/(count+1))
    {
        printf("Los patrones no tienen el mismo numero de picos,
por lo que no seran promediados\n");
        return 0;
    }*/

    for (i=0;i<spectre[current]->peaknum;i++)
    {
        orden = spectre[current]->peak[i].order;
        prom = spectre[current]-
>peak[i].relpeakarea/(spectre[current]-
>peak[i].errrelpeakarea*spectre[current]->peak[i].errrelpeakarea);
        errprom = 1/(spectre[current]-
>peak[i].errrelpeakarea*spectre[current]->peak[i].errrelpeakarea);
        for (j=current-count;j<current;j++)
        {
            prom += spectre[j]->peak[i].relpeakarea/(spectre[j]-
>peak[i].errrelpeakarea*spectre[j]->peak[i].errrelpeakarea);
            errprom += 1/(spectre[j]-
>peak[i].errrelpeakarea*spectre[j]->peak[i].errrelpeakarea);
        }
        prom = prom/errprom;
    }
}

```

```

errprom = 1/sqrt(errprom);

    if((f=fopen(filename[orden+5],"a+")) == NULL)//Se abre el
archivo y se verifica que la operación se haya llevado a cabo
correctamente. En mi caso el archivo está en la misma carpeta de
proyecto que este .c
    {
        perror("No pudo allocarse el puntero para abrir el
archivo\n");
        return NULL;
    }
    fprintf(f,"%lf\t%lf\t%lf\t%lf\n",spectre[current]-
>voltaje,prom,spectre[current]->errvoltaje,errprom);
    fclose(f);
}
return 1;
}

```