



**Universidad
Zaragoza**

Proyecto Fin de Carrera

Desarrollo de un software para la adquisición
de datos de un equipo de difracción
Malvern HS-2600 para determinar el tamaño de
burbujas generadas por cavitación

Autor

María Victoria Ferruz Alba

Directores

Luis Manuel Cerecedo Figueroa

Juan Antonio García Rodríguez

Facultad / Escuela

Escuela de Ingeniería y Arquitectura

Universidad de Zaragoza

Año 2014



Agradecimientos

Gracias a mi familia por estar siempre ahí pase lo que pase, su apoyo ha sido el mejor de los apoyos. A mi hermana, por tener la suerte de poder compartir tantas aficiones y poder disfrutarlas juntas y sobre todo por apoyarme y animarme en los malos momentos de la mejor de la formas, gracias Elena. Por supuesto, gracias a mi madre y a mi padre por enseñarme tantas cosas que tanto me han ayudado a lo largo de mi vida y sobre todo por el apoyo y confianza que siempre me habéis dado.

Agradecer todos los buenos momentos que me han dado y darán mis compañeros de clase, porque sin ellos estudiar esta carrera no habría sido lo mismo; gracias por pintar mis recuerdos del "CPS" con vuestra incomparable compañía.

Gracias a mi director, Luis Manuel, por brindarme la oportunidad de hacer un proyecto, que para mí, ha sido fascinante ya que he podido aprender muchísimo y además, en ámbitos que me apasionan. En especial, agradecerle a mi codirector, Chuan, por enseñarme tantas cosas y hacer que la elaboración de este proyecto haya sido tan agradable; gracias por los ánimos y confianza que siempre me has transmitido. También, agradecer a Esteban, perteneciente al departamento de mecánica de fluidos, el buen humor y ayuda prestada. Todo este párrafo lo puedo resumir simplemente diciendo: gracias Departamento de Mecánica de Fluidos.

Por último, añadir, que hay muchas personas que me han acompañado y apoyado a lo largo de este periodo de mi vida tan enriquecedor por lo que, sin más dilación, GRACIAS.

Resumen general

Una técnica ampliamente utilizada para la determinación de tamaños de partículas es la difracción. En el Área de mecánica de Fluidos de la Universidad de Zaragoza, se cuenta con un equipo apropiado para este fin: el equipo Malvern HS-2600. Dicho equipo, adquirido en 1986, ya no dispone del ordenador que el fabricante suministró y actualmente se controla a través de un ordenador con sistema operativo Windows 3.11, lo que no permite emplear los recientes y potentes programas para el análisis de resultados, y los datos únicamente pueden ser transferidos a través de un disquete de 3½. El desarrollo de un nuevo código permitirá la instalación en equipos con sistemas operativos modernos y, puesto que la nueva tarjeta se comunica con el ordenador a través de USB, permitirá el uso con cualquier ordenador.

El trabajo a realizar, por tanto, consistirá en implementar todo lo necesario para habilitar la adquisición de datos con una nueva tarjeta de adquisición.

La principal actividad es el desarrollo de un programa para la adquisición de los datos proporcionados por un difractor Malvern HS-2600 con una tarjeta de adquisición de National Instrument, NI USB-6501. El código se escribe en C#, en el entorno de programación Visual Studio 2010, puesto que se desea una aplicación que explote el potencial del entorno gráfico de Windows y, por tanto, es idóneo el uso de un lenguaje orientado a la programación con objetos. Se construye también un conector que adapta las entradas-salidas del equipo a la tarjeta de adquisición.

El difractor se quiere utilizar para la medida del tamaño de burbujas producidas por cavitación. Como se comenta en la sección de validación, los resultados obtenidos con un dispositivo en el que se genera cavitación a través de una onda acústica no fueron satisfactorios y se ha validado el código con otras medidas. Será necesario diseñar otro dispositivo que ofrezca señales suficientemente intensas para el estudio de la cavitación, este diseño y realización se salen del marco del presente proyecto.

Índice

1. Introducción	1
1.1. Objetivo	1
1.2. Antecedentes	1
1.3. Contexto actual	2
1.4. Descripción de las tareas.....	2
1.5. Contenido	3
2. Principio de funcionamiento: difracción.....	4
3. Descripción de cálculos	6
3.1. Cálculo de las clases de diámetro	6
3.2. Cálculo de las matrices de coeficientes.....	9
3.3. Cálculo de la distribución de probabilidad de clases de tamaño	11
3.4. Cálculo de los estadísticos y otros parámetros de interés.....	12
4. Implementación del sistema de adquisición.....	14
4.1. Adaptación del hardware	14
4.1.1. Funcionamiento interno del Malvern HS-2600.....	14
4.1.2. Conexiones y cableado.....	16
4.1.3. Tarjeta de adquisición de datos	18
4.2. Software: código de adquisición y procesamiento de datos.....	20
4.2.1. Lectura de datos del Malvern.....	20
4.2.2. Cálculo de la distribución de probabilidad de clases tamaño y otros parámetros ...	24
4.2.3. Representaciones gráficas.....	25
4.2.4. Guardado y lectura de archivos	26
4.2.5. Descripción y diseño de los formularios	28
5. Validación del software	35
6. Conclusiones.....	40
7. Bibliografía	40
8. Anexos.....	42
Anexo1: Detalles del código del software MalvernStudio	42
1.1. Cálculos	42
1.2. Representaciones gráficas.....	52
1.3. Guardado y lectura de archivos	54
1.4. Clases.....	62
Anexo 2: Vocabulario de programación.....	84

Anexo 3: Complemento sobre la teoría de la difracción.....	85
Anexo 4: Ficha de especificaciones de la tarjeta de adquisición	88

Índice de tablas

Tabla 1. Cambios realizados para la actualización del equipo	2
Tabla 2. Radios internos y externos de los semianillos detectores	7
Tabla 3. Clases de diámetros calculados para cada lente	8
Tabla 4. Líneas de comunicación.....	15
Tabla 5. Leyenda del esquema de la disposición de los cables y conexiones	16
Tabla 6. Correspondencias entre el DB-50 hembra y el Burndy 28 hembra.....	17
Tabla 7. Correspondencia entre el DIN-5P hembra y el Burndy 28 hembra.....	18
Tabla 8. Asignaciones de los terminales digitales	18
Tabla 9. Descripciones de las señales	19
Tabla 10. Mapa de bits de la tarjeta de adquisición de datos	19
Tabla 11. Leyenda del mapa de bits.....	19
Tabla 12. Opciones del software según el tipo de representación.....	26

Índice de figuras

Figura 1. Plano de apertura u objeto difractor y plano de la imagen de acuerdo a la difracción de Fresnel	4
Figura 2. Esquema de difractor.....	9
Figura 3. Aros detectores del difractor.....	10
Figura 4. Esquema de la disposición de los cables y conexiones	16
Figura 5. Vista frontal del conector Burndy 28 hembra.....	17
Figura 6. Dimensiones de la NI USB-6501	18
Figura 7. Opciones de la representación de datos.....	26
Figura 8. Ventana inicial del software	28
Figura 9. Jerarquía de formularios	29
Figura 10. Formulario F1_Alin durante la alineación	30
Figura 11. Formulario F1_Backg tras la medida del Background	30
Figura 12. Formulario F1_Medir tras realizar una serie de medidas	31
Figura 13. Formulario F1_Leer tras la lectura de un archivo	31
Figura 14. Formulario F2_MedirF1_TratDat	32
Figura 15. Formulario F2_RepresentarEner	33
Figura 16. Formulario F2_RepresentarDiam	33
Figura 17. Formulario F3_EstadisYParam.....	34
Figura 18. Plantilla de partículas de 130 micras de diámetro (patrón en transparencia).....	35
Figura 19. Representación gráfica de las energías leídas en el experimento de patrón en transparencia	36
Figura 20 Distribución de frecuencia de clases de diámetros obtenida en el experimento de patrón en transparencia.....	36

Figura 21. Distribución de frecuencias de clases de diámetros obtenida mediante el SprayTec del patrón en transparencia.....	37
Figura 22 Estadísticos y otros parámetros de interés obtenidos en el experimento de patrón en transparencia	37
Figura 23. Representación gráfica de las energías leídas en el experimento del aerosol comercial.....	38
Figura 24. Distribución de frecuencia de clases de diámetros obtenida en el experimento del aerosol comercial	38
Figura 25. Distribución de frecuencias de clases de diámetros obtenida mediante el SprayTec del aerosol comercial	38
Figura 26. Diagramas de la clase "basica" y de la clase "AlgoritmoComunicacion"	62
Figura 27. Diagrama de la clase "datos".....	63
Figura 28. Diagrama de la clase "medida"	63
Figura 29. Experimento de la doble rendija de Young.....	85
Figura 30. Evolución de una onda electromagnética a lo largo del tiempo.....	86
Figura 31. a) Patrón de difracción de Fraunhofer de una única rendija unidimensional. b) Fotografía de un patrón de difracción de Fraunhofer de una sola rendija.....	87

Índice de ecuaciones

[Ecuación 1]. Campo eléctrico en el plano z ($z=cte$).....	4
[Ecuación 2]. Simplificación del campo eléctrico en el plano (x,y,z)	5
[Ecuación 3]. Número de Fresnel	5
[Ecuación 4]. Relación entre la distribución de probabilidad de clases de diámetro con las energías	6
[Ecuación 5]. Fórmula utilizada para el cálculo de los diámetros.....	6
[Ecuación 6]. Intensidad obtenida para el caso de una apertura circular	9
[Ecuación 7]. Funciones de Bessel.....	9
[Ecuación 8]. Energía difundida por una partícula de diámetro D en una semicorona	10
[Ecuación 9]. Integral a través de la cual se calculan los términos de la matriz de coeficientes	10
[Ecuación 10]. Ecuación utilizada para el cálculo de los términos de la matriz de coeficientes.	10
[Ecuación 11]. Ecuación de los estadísticos D_{nm}	12
[Ecuación 12]. Ecuación del Span	13
[Ecuación 13]. Ecuación de la transmisión.....	13

1. Introducción

El trabajo a desarrollar consiste en implementar la adquisición de datos de un equipo Malvern HS-2600 para aprovechar las tecnologías informáticas actuales. El equipo obtiene la distribución de tamaños de un conjunto de gotas, partículas o burbujas esféricas, explotando las características de la luz difractada por dichas gotas. Para la integración con el sistema de adquisición se realizará el cableado de la tarjeta de adquisición de datos adquirida para este propósito y se construirá un programa de adquisición de datos para dicha tarjeta. El desarrollo del código de dicho programa es la parte más costosa del presente proyecto e incluye, además de la adquisición, el cálculo de las distribuciones de tamaño y de varios parámetros de interés.

La adquisición de datos y los cálculos se validarán para comprobar el buen funcionamiento del sistema de adquisición.

1.1. Objetivo

El principal objetivo de este proyecto es desarrollar un software de acuerdo con la tecnología actual, para la obtención de datos de un equipo de difracción a través de una tarjeta de adquisición de datos.

1.2. Antecedentes

El software que se había utilizado hasta ahora para controlar el equipo en Windows 3.11 fue desarrollado en el Proyecto Fin de Carrera de Agustín Esbec Albalad [referencia 7, ver bibliografía], dirigido por José Ignacio García Palacín, y se comunicaba con una tarjeta de adquisición de datos PCL-720 que se conectaba a través de un bus EISA y que por tanto, iba asociada al ordenador que se dispuso para el manejo del Malvern HS-2600.

El Proyecto Fin de Carrera de Agustín Esbec Albalad parte del riguroso conocimiento del equipo de medida que se obtuvo en el trabajo correspondiente al Proyecto Fin de Carrera de Jorge Lagranja Berges y Javier Vecino Soler [referencia 5, ver bibliografía] y el de José Ignacio García Palacín [referencia 6, ver bibliografía].

1.3. Contexto actual

Para actualizar el equipo se decidió, en primer lugar, cambiar la tarjeta de adquisición de datos. La nueva tarjeta de adquisición de datos se comunica por vía USB (Universal Serial Bus), por lo tanto, puede conectarse a cualquier equipo que disponga de los ampliamente usados puertos USB y, de esta forma, se gana una gran versatilidad de uso.

Este cambio de tarjeta vino ligado a un desarrollo de software completamente nuevo. Dicho software se ha programado en el entorno Microsoft Visual Studio 2010, el cual soporta múltiples lenguajes de programación. En este caso, el lenguaje utilizado en el desarrollo del nuevo código es C#, ya que su sintaxis es sencilla e intuitiva desde el punto de vista del programador. El conjunto de todo ello ofrece la posibilidad de hacer uso de la programación orientada a objetos, la cual permite el desarrollo y diseño del software de acuerdo a lo deseado, es decir, un programa que interactúe fácilmente con la tarjeta y a la par, ofrezca un entorno con ventanas y múltiples posibilidades, entre las que destacan, las opciones de representación gráfica y de cálculo.

A continuación, se muestra una tabla resumen en la que se indican los cambios realizados para la actualización del equipo.

Tarjeta de adquisición de datos	NI-USB-6501
Entorno de programación	Microsoft Visual Studio 2010
Lenguaje de programación	C#

Tabla 1. Cambios realizados para la actualización del equipo

1.4. Descripción de las tareas

El proyecto ha contemplado todas las tareas necesarias para la implementación del nuevo sistema de adquisición y la realización de medidas para la comprobación de su correcto funcionamiento. La tarea principal ha sido la escritura del código MalvernStudio, pero para su funcionamiento ha sido también necesaria la construcción de un cable para adaptar las líneas de comunicación a las conexiones de la tarjeta NI-USB-6501. El trabajo desarrollado para la implementación de dicho sistema de adquisición se describe en detalle en el capítulo 4.

Para la correcta programación de los cálculos efectuados ha sido necesario revisar los principios físicos de funcionamiento del difractómetro. El principio de funcionamiento y la descripción de estos cálculos se exponen en los capítulos 2 y 3.

Por último, se ha realizado la validación del software mediante la toma de medidas de:

- Una transparencia en la que se han impreso partículas de 130 micras
- Un aerosol comercial para limpiar cristales

De esta manera, se comprueba el correcto funcionamiento del software; desde la lectura de datos hasta el cálculo de las distribuciones de probabilidad de clases de diámetro. Todo ello queda recogido en el capítulo 5.

1.5. Contenido

El presente proyecto se compone de los siguientes capítulos:

- Principio de funcionamiento: difracción
- Descripción de cálculos
- Implementación de sistema de adquisición
- Validación del software

El capítulo 2, correspondiente al principio de funcionamiento, expone el principio físico en el que está basado el Malvern HS-2600: la difracción. Además, también se explica dicho fenómeno matemáticamente.

En el capítulo 3, se describe todo lo relacionado con los cálculos que se han tenido que realizar para la obtención de los datos deseados. Los datos que se quieren calcular son los siguientes:

- Distribución de probabilidad de clases de tamaño
- Estadísticos (D_{mn})
- Percentiles
- Span
- transmisión

Para ello se necesitan realizar una serie de cálculos previos, lo cuales se exponen con detalle en dicho capítulo.

En cuanto a la implementación del sistema de adquisición, expuesta en el capítulo 4, se compone de dos secciones: la adaptación del hardware y la creación del software.

Por último, en el capítulo 5 se recogen las medidas y cálculos realizados para la validación del software. Dicha validación garantiza el correcto funcionamiento, tanto en la lectura de datos como en los cálculos de distribuciones de diámetro.

2. Principio de funcionamiento: difracción

La luz se puede entender, de acuerdo con la física clásica, como ondas electromagnéticas que cumplen las ecuaciones de Maxwell. Así, en los fenómenos ópticos se manifiestan los comportamientos ondulatorios que dan lugar al efecto Doppler, al fenómeno de interferencia y a la difracción, entre otros.

La medida del tamaño de gotas, burbujas... por difracción aprovecha el fenómeno de la difracción para deducir la distribución de tamaños de los obstáculos que un haz de luz atraviesa a partir del análisis de la energía luminosa que llega a un conjunto de anillos detectores. La relación entre los diámetros y las energías se calcula gracias a la teoría de la difracción.

Cuando las ondas atraviesan una pequeña abertura o un obstáculo, se observa un patrón de interferencia ya que la luz no viaja en línea recta sino que manifiesta su carácter ondulatorio.

La integral de Fresnel permite obtener el patrón de campo eléctrico en un plano, en el que se eligen coordenadas (x, y) a partir del patrón de campo eléctrico en otro plano, en el que se eligen unas coordenadas (x', y') .

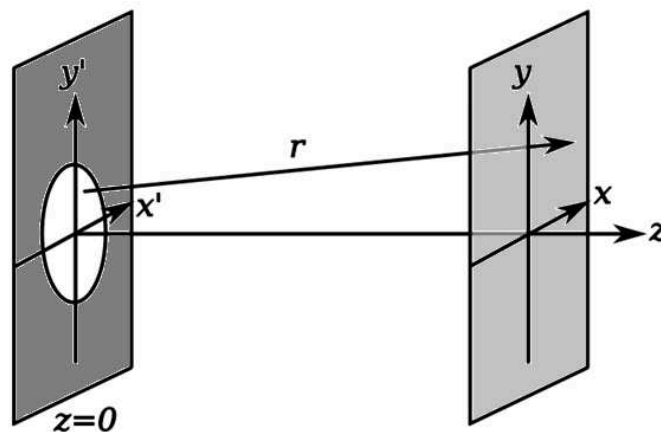


Figura 1. Plano de apertura u objeto difractor y plano de la imagen de acuerdo a la difracción de Fresnel

La integral de Fresnel expresa el campo eléctrico en el plano con coordenadas (x, y, z) :

$$E(x, y, z) = \frac{z}{i\lambda} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} E(x', y', 0) \frac{e^{-i\vec{k}\vec{r}}}{r^2} dx' dy'$$

[Ecuación 1]. Campo eléctrico en el plano z ($z=\text{cte}$)

Con: $r = \sqrt{(x - x')^2 + (y - y')^2 + z^2}$, λ es la longitud de onda.

Cuando la separación entre los planos es suficientemente grande o, lo que es lo mismo, el número de Fresnel es mucho menor que 1, definido en la Ecuación 3, la integral anterior se puede aproximar por la integral de Fraunhofer:

$$E(x, y, z) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} E(x', y', 0) e^{-\frac{2\pi}{\lambda z}(xx' + yy')} dx' dy'$$

[Ecuación 2]. Simplificación del campo eléctrico en el plano (x,y,z)

El número de Fresnel se define como:

$$F = \frac{d^2}{L\lambda}$$

[Ecuación 3]. Número de Fresnel

Donde d es una distancia característica del objeto que recorta el frente de ondas, L la distancia entre planos y λ la longitud de onda de la luz incidente.

Una revisión de la fenomenología más conocida a la que da lugar este fenómeno se puede consultar en el Anexo 3.

3. Descripción de cálculos

El objetivo principal de los cálculos a realizar es la obtención de la distribución de probabilidad de clases de diámetro a partir de las energías medidas. La relación entre las energías y distribución de probabilidad de clases de diámetros se puede escribir como:

$$[E_i] = [M_{ij}][D_j]$$

[Ecuación 4]. Relación entre la distribución de probabilidad de clases de diámetro con las energías

$[E_i]$: energías recogidas por el fotodetector

$[D_j]$: vector de distribución de probabilidad de clases de diámetro

$[M_{ij}]$: matriz de coeficientes que depende de:

- Longitud de onda del láser
- La distancia focal de la lente
- Las dimensiones del detector

Así, son necesarios una serie de cálculos previos:

- Cálculo de las clases de diámetro
- Cálculo de las matrices de coeficientes

3.1. Cálculo de las clases de diámetro

Las distribuciones de diámetros se obtienen buscando obtener el primer máximo de la función de intensidad luminosa, que se presentará en la ecuación 6, para cada radio externo de los anillos. Dado que el máximo de la función se da para 1,3566, se obtienen los diámetros de acuerdo con la fórmula:

$$d = \frac{1,3566 \cdot \lambda}{\pi} \sqrt{\left(\frac{F}{\rho_{max}}\right)^2 + 1}$$

[Ecuación 5]. Fórmula utilizada para el cálculo de los diámetros

λ = Longitud de onda del haz láser = 0,6328 mm

F = focal = {63, 100, 300, 600, 800, 1000}mm

ρ_{max} = radio externo del semianillo [mm]

Semianillo	r_{int} [mm]	r_{ext} [mm]	Área [mm ²]
1	0,149	0,218	0,040
2	0,254	0,318	0,058
3	0,353	0,417	0,077
4	0,452	0,518	0,101
5	0,554	0,625	0,131
6	0,66	0,737	0,169
7	0,772	0,856	0,215
8	0,892	0,986	0,277
9	1,021	1,128	0,361
10	1,163	1,285	0,469
11	1,321	1,461	0,612
12	1,496	1,656	0,792
13	1,692	1,88	1,055
14	1,915	2,131	1,373
15	2,167	2,416	1,793
16	2,451	2,738	2,339
17	2,774	3,101	3,018
18	3,137	3,513	3,928
19	3,549	3,978	5,072
20	4,013	4,501	6,526
21	4,536	5,085	8,297
22	5,121	5,738	10,524
23	5,773	6,469	13,384
24	6,505	7,282	16,827
25	7,318	8,184	21,088
26	8,219	9,185	26,409
27	9,22	10,287	32,695
28	10,323	11,501	40,383
29	11,537	12,837	49,773
30	12,873	14,3	60,909
31	14,336	15,9	74,282

Tabla 2. Radios internos y externos de los semianillos detectores

El primer máximo de la distribución de energías corresponde se da para 1,3566, que aparece en la ecuación anterior y se trata de un resultado obtenido de manera analítica a partir de la forma de la función de Bessel de orden 1.

Tal y como se ha desarrollado el procedimiento, se tienen 31 clases de diámetros que se han construido a partir de los 31 anillos útiles (todos menos el central). Así, se llega a un sistema determinado.

Clases Diam[mm]					
Focal[mm]					
63	100	300	600	800	1000
1,175567	1,835244	5,451065	10,891822	14,520425	18,149370
1,301541	2,038217	6,065469	12,121675	16,160431	20,199496
1,445089	2,268854	6,762407	13,516506	18,020392	22,524555
1,608606	2,530960	7,553321	15,099204	20,130825	25,162693
1,794858	2,828936	8,451430	16,896210	22,526988	28,157986
2,007626	3,168798	9,474821	18,943710	25,257126	31,570741
2,250489	3,556234	10,640576	21,275870	28,366799	35,457905
2,527478	3,997656	11,967950	23,931203	31,907358	39,883669
2,843882	4,501468	13,482191	26,960212	35,946139	44,932204
3,204539	5,075364	15,206393	30,409090	40,544734	50,680502
3,615472	5,728914	17,169290	34,335306	45,779772	57,224346
4,085330	6,475864	19,412153	38,821410	51,761317	64,701321
4,622097	7,328895	21,973045	43,943532	58,590879	73,238311
5,233093	8,299637	24,886867	49,771475	66,361527	82,951655
5,925790	9,399959	28,189243	56,376492	75,168269	93,960112
6,712041	10,648687	31,936673	63,871585	85,161771	106,452016
7,606853	12,069652	36,200675	72,399796	96,532760	120,665775
8,622027	13,681596	41,037481	82,073593	109,431191	136,788834
9,780100	15,520299	46,554456	93,107704	124,143371	155,179077
11,092707	17,604237	52,807032	105,613000	140,817126	176,021288
12,592990	19,986022	59,953065	119,905191	159,873406	199,841652
14,331886	22,746521	68,235167	136,469511	181,959187	227,448891
16,362843	25,970573	77,907868	155,815015	207,753212	259,691434
18,787288	29,819179	89,454183	178,907736	238,543526	298,179337
21,736419	34,500599	103,498895	206,997247	275,996223	344,995218
25,452695	40,399689	121,196585	242,392705	323,190183	403,987676
30,390170	48,237179	144,709454	289,418517	385,891280	482,364056
37,259462	59,141032	177,421394	354,842468	473,123229	591,404000
47,710106	75,729564	227,187354	454,374457	605,832561	757,290673
66,552432	105,638225	316,913699	633,827216	845,102918	1056,378627
97,253232	154,369829	463,108818	926,217510	1234,956656	1543,695806

Tabla 3. Clases de diámetros calculados para cada lente

3.2. Cálculo de las matrices de coeficientes

Para la obtención de los términos de la matriz de coeficientes se evalúa la integral de Fraunhofer [Ver ecuación 2], en el caso de una apertura circular de diámetro D , caso equivalente a la difracción por una esfera (obstáculo circular al frente de onda). Dicha integral conduce a la siguiente expresión para la intensidad:

$$I(\theta) = I_0 D^2 \left(\frac{2J_1(\alpha)}{\alpha} \right)^2$$

[Ecuación 6]. Intensidad obtenida para el caso de una apertura circular

Con: $\alpha = kD\theta$, siendo k el número de ondas y θ el ángulo con respecto a la luz incidente.

Las funciones de Bessel $J_n(x)$, se pueden definir por la expresión:

$$J_n(x) = \sum_{k=0}^{\infty} \frac{(-1)^k}{k! \Gamma(n+k+1)} \left(\frac{x}{2} \right)^{n+2k}$$

[Ecuación 7]. Funciones de Bessel

Con: $\Gamma(n+k+1) = \int_0^{\infty} e^{-x} \cdot x^{n+k} dx$, la función gamma

Así, se obtiene el patrón de intensidad en el plano en función del ángulo con la dirección de propagación de la luz incidente.

A continuación se muestra un esquema de los elementos principales de un difractómetro:

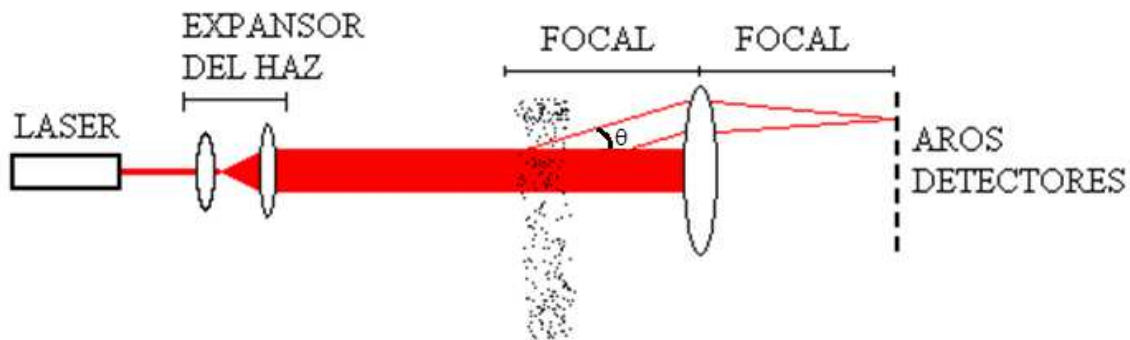


Figura 2. Esquema de difractómetro

En el difractómetro se dispone de aros detectores en forma de coronas semicirculares concéntricas. En la siguiente figura se muestra la forma de uno de los aros:

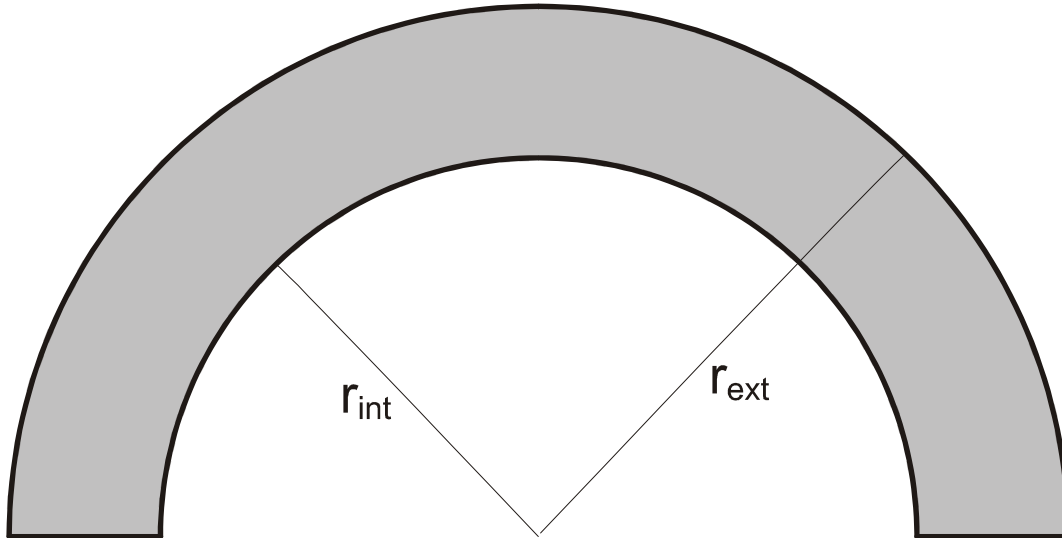


Figura 3. Aros detectores del difractómetro

Puesto que los datos que el difractómetro envía al ordenador son las energías totales que llegan a los anillos es necesario calcular qué energía llega a cada anillo para las distintas clases de diámetro. Así, la energía difundida por una partícula de diámetro D en una semicorona limitada por r_{int} , radio interior, y r_{ext} , radio exterior, se puede calcular como:

$$En(r_{int}, r_{ext}) = I_0 D^2 \int_{r_{int}}^{r_{ext}} \left(\frac{2J_1\left(D \frac{r}{f}\right)}{D \frac{r}{f}} \right)^2 \pi r dr$$

[Ecuación 8]. Energía difundida por una partícula de diámetro D en una semicorona

Donde se ha aproximado $\theta \sim \tan \theta = \frac{r}{f}$

Aprovechando las propiedades de las funciones de Bessel se puede demostrar que:

$$\frac{J_1^2(x)}{x} = -\frac{1}{2} \frac{d}{dx} [J_0^2(x) + J_1^2(x)]$$

[Ecuación 9]. Integral a través de la cual se calculan los términos de la matriz de coeficientes

Por lo que la integral anterior se puede resolver analíticamente. Así, es posible calcular los términos de la matriz de coeficientes. Se tienen 31 anillos y se quiere obtener la distribución de probabilidad en volumen para 31 clases de diámetros. Por tanto el problema pasa a ser discreto.

Para el cálculo de las matrices de coeficientes se ha hecho uso de la siguiente ecuación, en ella se expresa cada término de la matriz de coeficientes. Cada término es proporcional a la energía que llega al detector i por unidad de volumen de las gotas de diámetro j .

$$M_{ij} = -\frac{1}{D_j} \cdot \left[J_1^2\left(\frac{\pi \cdot D_j \cdot \sin \theta_{ext_i}}{0,6328}\right) + J_0^2\left(\frac{\pi \cdot D_j \cdot \sin \theta_{ext_i}}{0,6328}\right) - J_1^2\left(\frac{\pi \cdot D_j \cdot \sin \theta_{int_i}}{0,6328}\right) - J_0^2\left(\frac{\pi \cdot D_j \cdot \sin \theta_{int_i}}{0,6328}\right) \right]$$

[Ecuación 10]. Ecuación utilizada para el cálculo de los términos de la matriz de coeficientes

Siendo: $\text{sen}\theta_{\text{inti}} \sim \frac{r_{\text{inti}}}{f}$; $\text{sen}\theta_{\text{exti}} \sim \frac{r_{\text{exti}}}{f}$

Para obtener distribuciones de tamaño en volumen se divide por el diámetro de la clase. La expresión [Ecuación 6] es la intensidad que difunde una partícula o gota de diámetro D , así la intensidad varía con D^2 y para obtener los elementos de la matriz de coeficientes de manera que la distribución de diámetros sea volumétrica hay que dividir por D^3 , de ahí que para tener distribuciones en volumen, finalmente, haya que dividir por D_j .

La matriz M_{ij} de dimensiones 31x31 se calcula mediante la [Ecuación 10], a través de una hoja de cálculo de Excel.

Se calculan 6 matrices diferentes ya que se poseen 6 lentes de distintas distancias focales: 63, 100, 300, 600, 800, 1000 (todas ellas en milímetros). Dichas matrices se pueden consultar en el Anexo1 Detalles del código del software MalvernStudio, en el apartado de clases (concretamente en la clase "datos").

3.3. Cálculo de la distribución de probabilidad de clases de tamaño

Como se ha comentado, el sistema de ecuaciones que relaciona la distribución de energías adquiridas por el fotodetector con los diámetros de partículas que hay dentro del volumen intersectado por el haz láser se puede expresar matricialmente como en la ecuación 4.

Por lo tanto, la matriz $[M_{ij}]$ es una constante para cada experimento, que depende únicamente de lo anteriormente nombrado, es decir, es un parámetro calculado que no depende de la información recibida por el detector. En el Malvern HS-2600 se puede cambiar la lente mientras que el resto de los parámetros son fijos. Añadir que esta matriz está mal condicionada por lo que no se puede obtener la distribución de tamaños invirtiéndola.

El vector E_i corresponde a las energías leídas por el software de adquisición de datos que controla el equipo.

Y el vector de distribución de probabilidad de clases de tamaño de partículas $[D_j]$ se calcula mediante un proceso iterativo basado en el protocolo original del Malvern. Antes de comenzar con el proceso iterativo, se ha de normalizar el vector de energías leídas y una vez realizado dicho cálculo se procede de la siguiente manera:

0/Cálculo del vector $D_{j_{\text{inicial}}}$

Se repite el siguiente algoritmo hasta que el último error calculado sea menor que el error anterior, es decir, cuando se deje de reducir el error:

1/Normalización del vector D_j

2/Guardado del vector D_j en una variable auxiliar

3/Cálculo de la nueva $D_j \rightarrow D_j = D_j^2$

4/Cálculo de vector de energías (que posteriormente es normalizado) a través de la
 $D_{jsupuesta} \rightarrow [E_{i\text{calculada}}] = [M_{ij}][D_j]$

5/Cálculo del error $\rightarrow error = \sum_{i=0}^{30} (E_i - E_{i\text{calculada}})^2$

Tras este primer paso se realiza otro proceso iterativo, con la misma condición de parada, para minimizar el error de acuerdo con las energías leídas. El algoritmo es similar al anterior y se esquematiza a continuación:

1/Normalización del vector D_j

2/Guardado del vector D_j en una variable auxiliar

3/Cálculo de la nueva $D_j \rightarrow D_j = D_j \left(\left(\frac{E_j}{E_{j\text{calculada}}} \right)^2 + \left(\frac{E_{j+1}}{E_{j+1\text{calculada}}} \right)^2 \right)$

(Salvo para $j=30$, en este caso: $D_{30} = 2D_{30} \left(\frac{E_{30}}{E_{30\text{calculada}}} \right)^2$)

4/Cálculo de vector de energías (que posteriormente es normalizado) a través de la
 $D_{jsupuesta} \rightarrow [E_{i\text{calculada}}] = [M_{ij}][D_j]$

5/Cálculo del error $\rightarrow error = \sum_{i=0}^{30} (E_i - E_{i\text{calculada}})^2$

Cuando se termina todo el proceso iterativo, la variable a guardar como resultado final es la guardada en la variable auxiliar, la cual corresponde al vector de distribución de probabilidad de clases de diámetros.

Con la normalización se consiguen distribuciones en las que la suma de todas las frecuencias es 1.

3.4. Cálculo de los estadísticos y otros parámetros de interés

Los primeros estadísticos calculados son D_{10} , D_{30} , D_{32} , D_{43} , y todos ellos se calculan mediante la siguiente ecuación:

$$D_{nm} = \frac{\sum_{x=0}^{31} (\text{frecuencia de la clase } diam[x]) d_x^n}{\sum_{x=0}^{31} (\text{frecuencia de la clase } diam[x]) d_x^m}$$

[Ecuación 11]. Ecuación de los estadísticos D_{nm}

Los percentiles son el diámetro para el cual un porcentaje determinado de la distribución está por debajo de dicho diámetro. Los percentiles calculados son $D[10]$, $D[50]$ y $D[90]$.

Para el cálculo de los percentiles se procede de la siguiente manera: Siendo d_i y d_{i+1} el rango en el que está comprendido el diámetro en cual se cumple la condición de un percentil determinado y sus respectivas frecuencias f_i y f_{i+1} , se construye una recta:

$$\begin{aligned}f_{i+1} &= Ad_i + B \\f_i &= Ad_i + B\end{aligned}$$

Por lo tanto:

$$\begin{aligned}A &= \frac{f_{i+1} - f_i}{d_{i+1} - d_i} \\B &= f_{i+1} - Ad_{i+1}\end{aligned}$$

Luego el percentil correspondiente se obtiene de la siguiente ecuación:

$$\frac{X}{100} = A \cdot D[X] + B$$

Por ejemplo, para el caso del percentil D[10] se tendría:

$$0.1 = A \cdot D[10] + B$$

Añadir que esta ecuación es válida para aquellas distribuciones que hayan sido normalizadas y que los datos utilizados para los diámetros d_{i+1} y d_i son los límites de las clases de diámetro.

Otro parámetro que se tiene en cuenta es el denominado span, que se calcula de la siguiente manera:

$$Span = \frac{(D[90] - D[10])}{D[50]}$$

[Ecuación 12]. Ecuación del Span

Dicho parámetro ofrece una idea de la anchura de la distribución.

Y por último, la transmisión:

$$Transmisión = \frac{EnergíaMedida_{anillo0}}{EnergíaBackground_{anillo0}}$$

[Ecuación 13]. Ecuación de la transmisión

La transmisión evalúa el porcentaje de energía luminosa que no es dispersada por las partículas frente a la energía total del haz láser. En algunos difractómetros comerciales se da una estimación de la concentración de gotas en función de esta variable. No obstante la relación transmisión-concentración está afectada por errores relacionados con dispersiones múltiples y, siendo precisos, depende también de la distribución por lo que las concentraciones así obtenidas presentan importantes errores.

4. Implementación del sistema de adquisición

Este capítulo se compone de los siguientes apartados:

- Adaptación del hardware
- Software: código de adquisición y procesado de datos

En este capítulo se exponen las principales tareas necesarias para la elaboración del presente proyecto, ya que con la adaptación del hardware y la programación del software se obtienen los elementos necesarios para realizar medidas a través del Malvern HS-2600. Además, dicho software ofrece múltiples opciones de procesado de datos que serán explicadas en secciones posteriores.

4.1. Adaptación del hardware

En primer lugar, se explica el funcionamiento interno del Malvern ya que en base a ello se hacen las conexiones pertinentes, además de la disposición en memoria de la tarjeta de adquisición de datos. Por lo tanto, la presente sección se divide en las siguientes partes:

- 4.1.1. Funcionamiento interno del Malvern
- 4.1.2. Conexiones y cableado
- 4.1.3. Tarjeta de adquisición de datos
 - Especificaciones de la tarjeta de adquisición de datos NI USB-6501
 - Mapa de bits

4.1.1. Funcionamiento interno del Malvern HS-2600

Para comunicarse a través de un ordenador con el Malvern es necesario transmitir señales eléctricas a través de un medio físico. Esto se consigue gracias a las denominadas líneas de comunicación, y en este caso son de dos tipos: las que transmiten señales del Malvern al ordenador y las que lo hacen del ordenador al Malvern.

El Malvern manda la información al ordenador mediante 11 líneas de comunicación, siendo 10 de ellas las encargadas de transmitir el valor de la energía medida por uno de los semianillos detectores. Es decir, cada vez que se realiza una medida se obtiene la energía detectada por cada uno de los semianillos, por lo tanto, el dato final obtenido es un vector de 32 componentes ya que el sistema de detección está provisto de 32 semianillos. Añadir, que el hecho de que la información obtenida por cada uno de los semianillos sea enviada a través de 10 líneas, quiere decir que cada componente del vector de energías es un dato de 10 bits. La otra línea que queda, es la encargada de informar cuándo dichos datos están disponibles para ser leídos.

Por otro lado, el ordenador envía la información necesaria al Malvern a través de 8 líneas, siendo 5 de ellas las utilizadas para direccionar hacia la memoria en la cual el Malvern almacena los datos obtenidos. Hay 32 direcciones de memoria ya que al haber 32 semianillos detectores se obtienen 32 lecturas. Las 3 líneas restantes están relacionadas con el disparo que ha de realizarse para provocar la lectura y conversión de la señal captada por cada uno de los semianillos. Una de esas 3 líneas sirve para indicar el modo de disparo, otra para desinhibir/inhibir dicho disparo y la última para el disparo propiamente dicho.

A continuación, se muestra la siguiente tabla resumen de lo anteriormente nombrado:

Malvern-->Ordenador [11 líneas]	
Líneas requeridas	Función
10	Lectura semianillo
1	Datos disponibles
Ordenador-->Malvern [8 líneas]	
Líneas requeridas	Función
5	Direccionar memoria
1	Modo disparo
1	Desinhibición/Inhibición disparo
1	Disparo

Tabla 4. Líneas de comunicación

Por lo tanto, el procedimiento a seguir para lograr una correcta comunicación con el Malvern a través de un ordenador y viceversa es el siguiente:

- 1) Preparación del disparo:
 - a) Selección del modo de disparo
 - b) Desinhibición del disparo
 - c) Disparo
 - d) Inhibición del disparo
- 2) Determinar si los datos están disponibles para ser leídos
- 3) Direccionamiento de memoria
- 4) Lectura de los datos

Todo ello se explica con mayor detalle en la sección 4.2. Código de adquisición y procesado de datos.

4.1.2. Conexiones y cableado

Una vez conocido el funcionamiento interno del Malvern, se establece el cableado y las conexiones pertinentes. Dicho cableado y conexiones guarda total relación con las líneas de comunicación, anteriormente mencionadas, necesarias para la correcta comunicación entre Malvern y el ordenador y viceversa.

En primer lugar, se expone una figura en la que se observa la disposición del cableado con sus respectivas conexiones, así como los tipos de conectores y dispositivos utilizados.

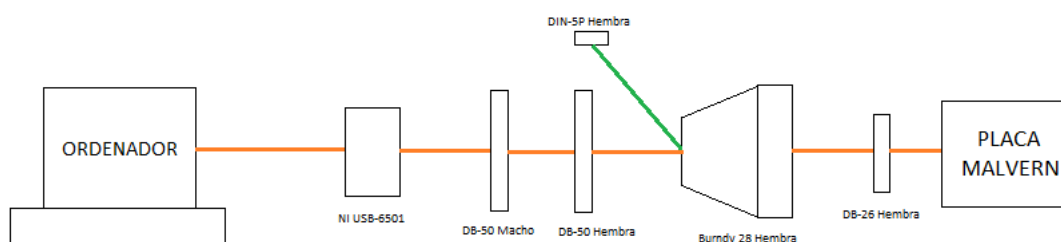


Figura 4. Esquema de la disposición de los cables y conexiones

La línea naranja corresponde a 20 líneas de comunicación, de las cuales, 11 son las líneas que van del Malvern al ordenador, 8 las que van del ordenador al Malvern y la última es la referencia de tierra (GND). Por otro lado, la línea verde corresponde a las 4 líneas de comunicación que conectan el Malvern con la fuente de alimentación.

A continuación, se muestra una tabla en la que se resume lo anteriormente mencionado.

	Línea naranja
Malvern-->Ordenador	11
Ordenador-->Malvern	8
GND digital	1
	Línea verde
Malvern-->Alimentación	4

Tabla 5. Leyenda del esquema de la disposición de los cables y conexiones

Se ha de tener en cuenta que en la figura 4 únicamente se muestran los conectores y dispositivos más relevantes a la hora de realizar el cableado, además, aunque se haya dibujado una única línea no quiere decir que sólo haya una línea de comunicación, sino que la línea representada contiene todas las líneas previamente comentadas (este hecho se ha explicado previamente de forma detallada).

El ordenador a través del cual se controla el Malvern está conectado a la tarjeta de adquisición NI USB-6501 y dicha tarjeta se conecta a 20 cables que terminan en un DB-50 macho. Estos cables corresponden a las 20 líneas de comunicación que requiere la instalación y que a través del DB-50 macho se conectan al DB-50 hembra.

Antes de mostrar la tabla de correspondencias entre el DB-50 hembra y el Burndy 28 hembra con su respectiva función, añadir que el Burndy 28 posee la siguiente vista frontal:

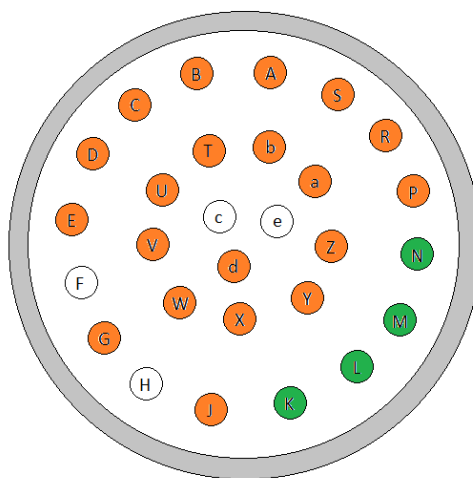


Figura 5. Vista frontal del conector Burndy 28 hembra

Como se puede observar en la figura 5., las partes que están coloreadas en naranja corresponden a las conexiones a las cuales se conectan las 20 líneas de comunicación, tal y como se ha visto en anteriores figuras, y por otro lado, las partes verdes corresponden a las líneas que conectan el Malvern con la fuente de alimentación.

Una vez vista la parte frontal del Burndy 28 hembra, se muestra la tabla de correspondencias de dicho conector con respecto al DB-50 hembra, además de las funciones de cada una de las líneas.

Función	DB-50 Hembra	Burndy 28 Hembra
Multiplexado	1	C
Multiplexado	2	B
Multiplexado	3	A
Multiplexado	4	S
Multiplexado	5	R
Inhibición disparo	9	J
Modo disparo	18	E
Disparo	19	P
GND	25	D
Datos disponibles	32	G
Dato	34	d
Dato	35	a
Dato	36	b
Dato	37	T
Dato	38	U
Dato	39	V
Dato	40	W
Dato	41	X
Dato	42	Y
Dato	43	Z

Tabla 6. Correspondencias entre el DB-50 hembra y el Burndy 28 hembra

Por último, se muestra la tabla de correspondencias entre el Burndy 28 y el DIN-5P hembra:

Función	DIN-5P Hembra	Burndy 28 Hembra
Masa	1	N
5V	2	K
15V	3	L
15V	4	M

Tabla 7. Correspondencia entre el DIN-5P hembra y el Burndy 28 hembra

4.1.3. Tarjeta de adquisición de datos

La tarjeta de adquisición de datos es el dispositivo encargado de recibir los datos del Malvern y transmitirlos al ordenador. La nueva tarjeta utilizada es la NI USB-6501 de National Instruments, la cual ofrece 24 líneas DIO (digital input/output) distribuidas en tres bytes o puertos, y un contador 32-bit; todo ello a través de una conexión USB 2.0. La siguiente figura muestra las dimensiones de dicha tarjeta:

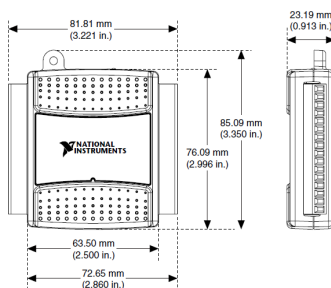


Figura 6. Dimensiones de la NI USB-6501

La NI USB-6501 posee dos bloques de terminales desmontables para señales digitales, los cuales se muestran en la siguiente tabla con sus correspondientes asignaciones.

Modulo	Terminal	Señal	Módulo	Terminal	Señal
	1	GND		17	P0.0
	2	+5V		18	P0.1
	3	P1.7		19	P0.2
	4	P1.6		20	P0.3
	5	P1.5		21	P0.4
	6	P1.4		22	P0.5
	7	GND		23	P0.6
	8	GND		24	P0.7
	9	P2.7/PFI 0		25	GND
	10	P2.6		26	GND
	11	P2.5		27	P1.0
	12	P2.4		28	P1.1
	13	P2.3		29	P1.2
	14	P2.2		30	P1.3
	15	P2.1		31	+5V
	16	P2.0		32	GND

Tabla 8. Asignaciones de los terminales digitales

[Terminales en morado: Puerto 0/Terminales en azul: Puerto 1/Terminales en rojo: Puerto 2]

A continuación, se muestra otra tabla en la que se describen las señales que ofrece cada uno de los conectores I/O (input/output).

Nombre de la señal	Sentido	Descripción
P0.<0..7>	Output o Input	Señales digitales I/O--Se puede configurar cada señal individualmente como input o como output
P1.<0..7>		
P2.<0..7>		
P2.7/PFI 0	Output o Input	Esta conexión es configurable como una línea digital I/O o como un contador
5V	Output	La fuente de voltaje proporcionada por el USB
GND	-	Tierra--La referencia para las señales digitales

Tabla 9. Descripciones de las señales

Como se puede observar, la NI USB-6501 posee 24 líneas digitales, P0.<0..7>, P1.<0..7>, P2.<0..7>, en tres puertos digitales input/output.

Añadir que a lo largo del presente proyecto se habla de input/output (entrada/salida) cuando se quiere hacer referencia al sentido en el que son enviados los datos, ya que en este ámbito es habitual el uso de terminología inglesa.

Una vez conocidas las especificaciones de la tarjeta de adquisición utilizada, se muestra, mediante un mapa de bits, la disposición en memoria establecida de dicha tarjeta de las líneas/funciones del Burndy 28.

	Líneas							
	0	1	2	3	4	5	6	7
P0	DO	D1	D2	D3	D4	D5	D6	D7
P1	DD						D8	D9
P2	M0	M1	M2	M3	M4	MD	ID	DE

Tabla 10. Mapa de bits de la tarjeta de adquisición de datos

Lo primero a tener en cuenta, es que tanto el puerto 0 como el puerto 1 se configuran como puertos de lectura y que el puerto 2 se usa para escritura, además de la leyenda que se muestra a continuación.

<D0..D9>	Dato
<M0..M9>	Multiplexado
DD	Datos disponibles
ID	Inhibición/Desinhibición disparo
MD	Modo disparo
DE	Disparo externo

Tabla 11. Leyenda del mapa de bits

A continuación, se expone una breve explicación sobre el mapa de bits establecido, ya que en la sección 4.2. Código de adquisición y procesado de datos se explicará con todo detalle cada una de las líneas de cada uno de los puertos.

Una secuencia de señales en las líneas 5, 6 y 7 del puerto 2 inicia la secuencia de lectura de datos del difractor.

Cuando se comienza la lectura de datos, los valores de las celdas resaltadas en azul no se modifican. Por otro lado, para el direccionamiento de memoria o multiplexado se hace uso de las 5 celdas amarillas, ya que para contemplar los números comprendidos entre el 0 y el 31 (con los que se numeran los semianillos que posee el Malvern) se necesitan 5 bits. Y por último, las celdas rojas corresponden al dato de 10 bits obtenido de la lectura del semianillo correspondiente al direccionamiento realizado.

Una vez establecidas las correspondencias entre las líneas de comunicación y su disposición en memoria de la tarjeta de adquisición de datos, ya se puede comenzar a programar el software que controla el Malvern HS-2600.

4.2. Software: código de adquisición y procesado de datos

En esta sección se expone todo lo relacionado con el desarrollo del software de adquisición y procesado de datos, empezando por la lectura de datos del Malvern, después con el cálculo de la distribución de tamaño y otros parámetros de interés, en tercer lugar, las presentaciones gráficas, y por último, el guardado y lectura de archivos. Además de lo anteriormente mencionado, se realiza una descripción de las ventanas o formularios con sus respectivos posibles tránsitos entre las mismas.

En los siguientes apartados se comentan con detalle las distintas funcionalidades del código. Las ventanas que se han generado se corresponden con las tareas que se necesitan en las distintas fases de medida, análisis... En el formulario inicial, que se presenta al usuario cuando se ejecuta el programa, se presentan las opciones “alinear”, “background” (iluminación de fondo), “medir”, “leer” y “tratamiento de datos”. Estas ventanas completan las necesidades operativas del usuario. Concretamente, antes de proceder a la lectura y procesado de datos son necesarios los siguientes pasos previos:

- La ventana “alinear” guía al usuario para conseguir que la mayor parte de la energía del haz láser llegue al anillo central, como preparación del equipo para la medida.
- La ventana “background” permite al usuario registrar la luz de fondo para una posterior corrección de las medidas brutas.

4.2.1. Lectura de datos del Malvern

Hay tres opciones del software con las que se realiza la comunicación con el Malvern:

- Alinear
- Background
- Medir

Como ya se ha comentado en anteriores secciones, para llevar a cabo la comunicación con el Malvern se han de seguir una serie de pautas, que en su conjunto vamos a denominar como algoritmo de comunicación.

Para realizar dicho algoritmo de comunicación se ha creado una clase denominada "AlgoritmoComunicacion", la cual reduce las líneas de código escritas, además de ofrecer una lectura de código más clara y sencilla. Además, en dicha clase se hace uso de otra clase llamada "basica" que principalmente se encarga de establecer los canales de lectura y escritura. Ambas clases están expuestas en detalle en el Anexo 1 Detalles del código del software. De la clase "AlgoritmoComunicacion" merece la pena destacar los siguientes aspectos:

- DD(Datos disponibles): cuando los datos están disponibles se produce un flanco de bajada, de uno a cero.
- DE(Disparo externo): para llevar a cabo el disparo se ha de realizar un flanco de subida, de cero a uno, por eso el valor final que ha de haber en ese bit es un uno.
- MD(Modo de disparo): se escribe un cero cuando se quiere un disparo interno y un uno cuando se quiere uno externo.
- ID (Inhibición/Desinhibición de disparo): para la inhibición del disparo se escribe un cero y para la desinhibición se escribe un uno.

A continuación, se muestra la secuencia que se ha de llevar a cabo antes de comenzar la lectura del difractor:

1. Selección del modo externo de disparo: escritura de un uno ($00100000_2 = 32_{10}$) en la línea 5 del puerto 2
2. Desinhibición del disparo: escritura de un uno ($01100000_2 = 96_{10}$) en la línea 6 del puerto 2.
3. Disparo: escritura de un uno ($11100000_2 = 224_{10}$) en la línea 7 del puerto 2. Para realizar el disparo se ha de producir un flanco de subida, y como se puede observar anteriormente había un cero.
4. Inhibición del disparo: escritura de un cero ($10100000_2 = 160_{10}$) en la línea 6 del puerto 2.

Además, para construir el dato final obtenido, a través de la clase "AlgoritmoComunicacion", energias[i] se han de combinar los datos del puerto 0 con los del puerto 1.

El dato completo y ordenado es D9D8D7D6D5D4D3D2D1D0 y tal y como está dispuesto dicho dato en el mapa de bits, se han de realizar los siguientes cálculos:

$$datoP0 = D7 * 2^7 + D6 * 2^6 + D5 * 2^5 + D4 * 2^4 + D3 * 2^3 + D2 * 2^2 + D1 * 2^1 + D0 * 2^0$$

Al $datoP1_0$, lectura bruta del puerto 1, hay que aplicarle una máscara ya que solo interesan dos de los bits del byte:

$$datoP1 = datoP1_0 \& \{11000000\} = D9 * 2^7 + D8 * 2^6 = 2^6 * (D9 * 2 + D8)$$
$$[11000000_2 = 192_{10}]$$

Pero el *datoP1* correcto, de acuerdo con el peso de las cifras es el siguiente:

$$datoP1_{correctado} = D9 * 2^9 + D8 * 2^8 = 2^8 * (D9 * 2 + D8)$$

Es decir: $\frac{2^8(D9*2+D8)}{2^6(D9*2+D8)} = 2^2 = 4$, hay que multiplicar por 4 para obtener el número correcto del dato.

Por lo tanto, la ecuación a tener en cuenta para la obtención del dato correcto de la lectura de energía de un determinado semianillo *i*, es la siguiente:

$$energias[i] = datoP0 + [datoP1_0 \& 192] * 4$$

La anterior ecuación corresponde a las energías brutas obtenidas en cada uno de los semianillos detectores. Sin embargo, el dato guardado en el campo "MatrizEnergias" es el obtenido tras restar el "background" y multiplicar por la variable "calibdiados":

medidaRealizada.MatrizEnergias[muestras, anillo] = (AlgoritmoComunicacion.energias[anillo] - medidaRealizada.Bckg[anillo]) * datos.calibdiados[anillo];

Cada componente del vector "calibdiados" corresponde al factor de corrección de un semianillo fotodetector. Gracias a estos factores de corrección se consiguen obtener datos más precisos, ya que se tienen en cuenta la respuesta de los fotodetectores calibrada por el fabricante. Este vector se puede consultar en la clase "datos" dentro de apartado de clases que se encuentra en el Anexo 1.

Por otro lado, cada vez que se realiza una medida, ésta es guardada en un listado hasta que se cierra el software, además, existe la posibilidad de que dicha medida sea guardada en un archivo de texto (se explica en detalle en la subsección 4.2.4. Guardado y lectura de archivos). Todo ello implica la necesidad de la creación de una tercera clase: la clase "medida", expuesta en detalle en el Anexo 1 Detalles del código del software.

A continuación, se muestra la descripción de los principales campos que posee la clase medida para tener una visión más clara de lo que aporta dicha clase. Se ha de tener en cuenta que los índices utilizados no son los mismos que los usados en el código de programación.

Nombre: nombre de la medida realizada.(tipo string)

FechaHora: fecha y hora en el momento en el que se realiza la medida.(tipo string)

Longitud: número de muestras de una medida(int).{1,2,3, ..., *N*} Siendo *N* = Longitud

Lente: focal de la lente utilizada para la realización de la medida {63, 100, 300, 600, 800, 1000}

AnillosDesech: número de energías medidas por los anillos más próximos al central que son establecidas a cero para la realización del cálculo de la distribución de probabilidad de clases de tamaño (tipo int).

i: anillos {0,1,2, ...,31}

Se utilizará el índice k para las clases de tamaño. Hay 31 clases de tamaño $\rightarrow k$:clases de tamaño $\{0,2,3, \dots, 30\}$

j : número de muestras $\{0,2,3, \dots, N - 1\}$

Bckg: double []

(...)

\rightarrow *anillos*[i]

Cada componente del vector corresponde a la energía medida por el anillo[i]

MatrizEnergias: double [,]

$\left(\begin{array}{ccc} & \dots & \\ \vdots & \ddots & \vdots \\ & \dots & \end{array} \right) \downarrow$ *Número de muestras*[j]

\rightarrow *anillos*[i]

Cada componente de la MatrizEnergias corresponde a la energía medida por el anillo[i] de la muestra [j]: E_{ij}

PromEnerg: double []

(...)

\rightarrow *anillos*[i]

Cada componente del vector se calcula mediante la siguiente expresión:

$$E_i = \frac{1}{N} \sum_{j=1}^N E_{ij} = \text{Energía promedio obtenida en el anillo}[i]$$

MatrizDiametros: double [,]

$\left(\begin{array}{ccc} & \dots & \\ \vdots & \ddots & \vdots \\ & \dots & \end{array} \right) \downarrow$ *Número de muestras*[j]

\rightarrow *clases de tamaño*[k]

Cada componente de la MatrizDiametros corresponde a la frecuencia de aparición de la clase de tamaño[k] de la muestra [j]: D_{kj}

PromDiamMet: double []

(...)

\rightarrow *clases de tamaño*[j]

$D_k =$ frecuencia de la clase de tamaño[k]. Calculado a partir del vector PromEnerg.

Para cada muestra de la MatrizDiametros se calculan unos parámetros estadísticos y otros parámetros de interés.

MatrizParam: double[,]

$\begin{pmatrix} \vdots & \ddots & \vdots \\ \vdots & \ddots & \vdots \\ \vdots & \ddots & \vdots \end{pmatrix} \downarrow \text{Número de muestras}[j]$

→ tipos de parámetros { D_{10} D_{30} D_{32} D_{43} $D[10]$ $D[50]$ $D[90]$ Span Transmisión }

Leyenda: **Diámetros significativos(Estadísticos)** **Percentiles**

PromParam: double[]

(...)

→ tipos de parámetros { D_{10} D_{30} D_{32} D_{43} $D[10]$ $D[50]$ $D[90]$ Span Transmisión }

Además, esta clase posee un campo en el que se contempla si se ha realizado el cálculo de la distribución de probabilidad de clases de tamaño:

CtrlCalcDiam: tipo booleana

En el Anexo 1 Detalles del código del software, se muestran los diagramas de las clases en los cuales se exponen los distintos métodos y campos que contiene cada una de ellas, de esta manera se obtiene una visualización más clara de las clases.

En los diagramas expuestos de dicho anexo se puede observar que además de los datos obtenidos a través de una medición, también aparecen los parámetros calculados ya que la clase medida ofrece los campos necesarios para poder guardar todos los parámetros asociados a una medida, desde las energías medidas hasta los datos obtenidos a través de cálculos. Para la obtención de la distribución de probabilidad de clases de tamaño y otros parámetros de interés son necesarios una serie de cálculos que se exponen en la siguiente subsección 4.2.2.

4.2.2. Cálculo de la distribución de probabilidad de clases tamaño y otros parámetros

En primer lugar, el código encargado de los cálculos consta de un bucle que se repite para cada muestra de la medida, es decir se realizan una serie de instrucciones hasta que una variable, inicializada en cero, es mayor o igual que "medidaSeleccionada.Longitud". A continuación se expone un esquema de lo anteriormente explicado:

```
muestras=0
```

```
while (muestras < medidaSeleccionada.Longitud)
```

```
{
```

- Cálculo de la "MatrizDiametros" a través del método iterativo basado en el protocolo original del Malvern.
- Cálculo de los estadísticos (D_{mn})
- Cálculo de los percentiles
- Cálculo del span y de la transmisión.

```
muestras++ (Se incrementa la variable en uno).
```

```
}
```

Una vez realizados los cálculos para cada muestra de la medida realizada, se procede al cálculo de las variables promedio: vector promedio de los parámetros de interés (estadísticos, percentiles, span y transmisión) y el vector de probabilidad de tipos de diámetro promedio. El primer vector se calcula realizando una media aritmética de cada uno de los parámetros, en cambio, para el segundo se calcula mediante el método iterativo basado en el protocolo original del Malvern, pero partiendo de las energías promedio de dicha medida. Todo el código asociado a los cálculos anteriormente mencionados puede consultarse en el Anexo 1, en el apartado de cálculos.

Además, para todos estos cálculos son necesarios una serie de datos:

- Matriz de coeficientes
- Clases de diámetro
- Límites de las clases de diámetro

Todos ellos están contemplados en la clase datos, la cual puede observarse en el Anexo 1, en el apartado de clases.

4.2.3. Representaciones gráficas

El uso de representaciones gráficas se realiza en las siguientes opciones del software:

- Alinear
- Background
- Representación de datos del formulario tratamiento de datos.

Además, las representaciones gráficas que se realizan son de dos tipos:

- Energías vs. anillos
- Frecuencia de clases de diámetro vs. clases de diámetro

La siguiente tabla muestra las opciones del software según el tipo de representación requerida:

Energías vs. Anillos	Frecuencia vs. Clases de diámetro
Alinear Background Tratamiento de datos	Tratamiento de datos

Tabla 12. Opciones del software según el tipo de representación

Las opciones ofrecidas por la representación de datos que aparecen en el formulario de tratamiento de datos se muestran en la siguiente figura:

Representación de datos

Representar

☒ Energías leídas

☐ Distribución de diámetros

Figura 7. Opciones de la representación de datos

Por eso el tratamiento de datos aparece tanto en la columna de energías vs. anillos como en la de frecuencia vs. clases de diámetro.

Las líneas de código utilizadas en cada una de las opciones del software que proporcionan representaciones gráficas se exponen en el apartado Representaciones gráficas del Anexo 1

4.2.4. Guardado y lectura de archivos

El software está provisto de la opción de guardado para que el usuario pueda tener acceso a las medidas y cálculos realizados con anterioridad a través de un archivo texto. Además, al haber una opción de lectura de archivos, se pueden realizar nuevos cálculos, medidas o simplemente realizar representaciones gráficas con resultados obtenidos en sesiones anteriores que hayan sido guardados. El guardado y lectura de archivos ofrece una gran versatilidad de uso, ya que se pueden retomar sesiones de medida previamente guardadas para representarlas gráficamente, obtener nuevos resultados y guardarlos.

Las líneas de código que crean y escriben el archivo de texto para el guardado de medidas se muestran en el apartado de guardado y lectura del Anexo 1 Detalles del código del software. Además, en dicho anexo también se expone un ejemplo de guardado de una medición en la que se han calculado todos los parámetros que ofrece el software.

Los dos métodos utilizados para la escritura de un archivo son los siguientes:

`File.WriteAllLines(string path, string[] contents)-->` Crea un archivo nuevo, escribe en él la matriz de cadenas especificada y, a continuación, lo cierra.

`File.AppendAllText(string path, string contents)`--> Abre un archivo, le anexa la cadena especificada y, a continuación, cierra el archivo. Si el archivo no existe, este método crea un archivo, escribe la cadena especificada en él y, a continuación, lo cierra.

Destacar que para la escritura de vectores o matrices, se realiza elemento a elemento ya que no existe una función prevista para esta operación, de ahí la utilización de los bucles de tipo for (las instrucciones de bucle repiten un bloque de código especificado hasta que se cumpla una condición determinada).

En cuanto a la lectura de archivos de texto, el procedimiento seguido es el siguiente:

- Lectura de todas las líneas del archivo seleccionado mediante la siguiente línea de código:

lectura = `File.ReadAllText(filePath)`--> Este método abre un archivo de texto, lee todas sus líneas y, a continuación, cierra el archivo.

- División en partes de la lectura realizada a través de unos determinados delimitadores, de tal forma, que se obtiene un vector en el que cada componente es una palabra, a este vector se le ha denominado parts. Las líneas de código que realizan lo anteriormente nombrado son las siguientes:

```
char[] delimiters = new char[] { ' ', '\n', '\t', '\r', '\v' };
```

```
parts = lectura.Split(delimiters, StringSplitOptions.RemoveEmptyEntries);
```

- Guardado en RAM de los parámetros de interés leídos. Estos parámetros se guardan en los campos de una variable de tipo medida y a su vez, dicha variable medida se guarda en una lista de medidas.

El código asociado a la lectura de archivos se recoge en el Anexo 1 Detalles del código del software.

Ahora que ya se han comentado todas las opciones que ofrece el software, se muestra la descripción y diseño de los formularios o ventanas que posee el software, además de los posibles tránsitos entre los mismos.

4.2.5. Descripción y diseño de los formularios

En primer lugar, se muestra la ventana de inicio del software, de tal forma, que se visualizan las principales opciones que ofrece dicho software.

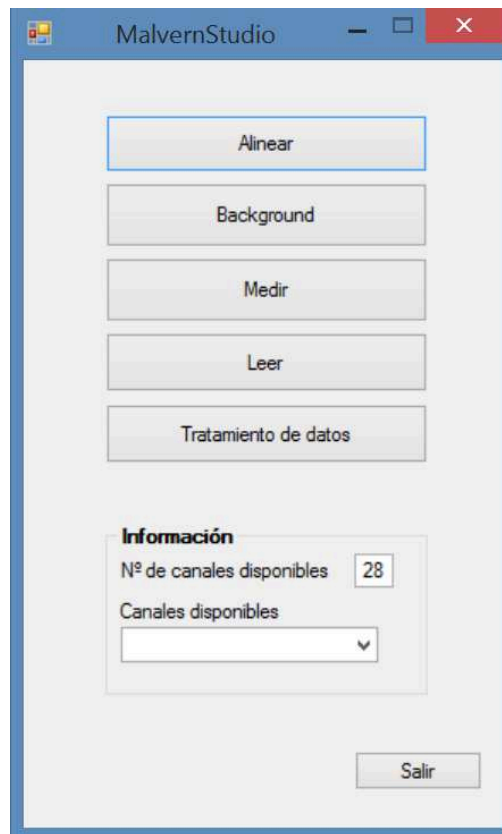


Figura 8. Ventana inicial del software

Como se puede observar en la figura 8, estas son las opciones que ofrece:

- **Alinear:** antes de comenzar a medir es preciso ajustar manualmente el Malvern mediante unas ruletas para alinear correctamente el haz láser, de tal manera que su imagen, en el plano de los anillos, incida en el anillo central.
- **Background:** otra de las acciones previas que hay que realizar antes de comenzar a medir. El background se toma sin partículas en la zona de medida. Es, por tanto, una medida del ruido óptico de fondo y electrónico que posteriormente se resta de las medidas, ya que se ha de tener en cuenta la medida del cero para poder tarar el dispositivo.
- **Medir:** esta opción nos permite obtener los datos de las energías leídas por los 32 semianillos del Malvern
- **Leer:** mediante esta opción se pueden leer archivos de medidas anteriormente guardadas, para poder visualizar los datos en gráficas o realizar nuevos cálculos.

- Tratamiento de datos: permite realizar cálculos a partir de las medidas de energía, tales como la distribución de probabilidad de clases de diámetro, estadísticos y otros parámetros de interés. Además de visualizar mediante gráficas todos los datos previamente mencionados y el guardado de dichos datos, asociados a una determinada medida.

Antes de continuar con la descripción de los formularios restantes, se expone un esquema en el que se muestra la jerarquía existente entre los distintos formularios que componen el software, además de todos los posibles tránsitos.

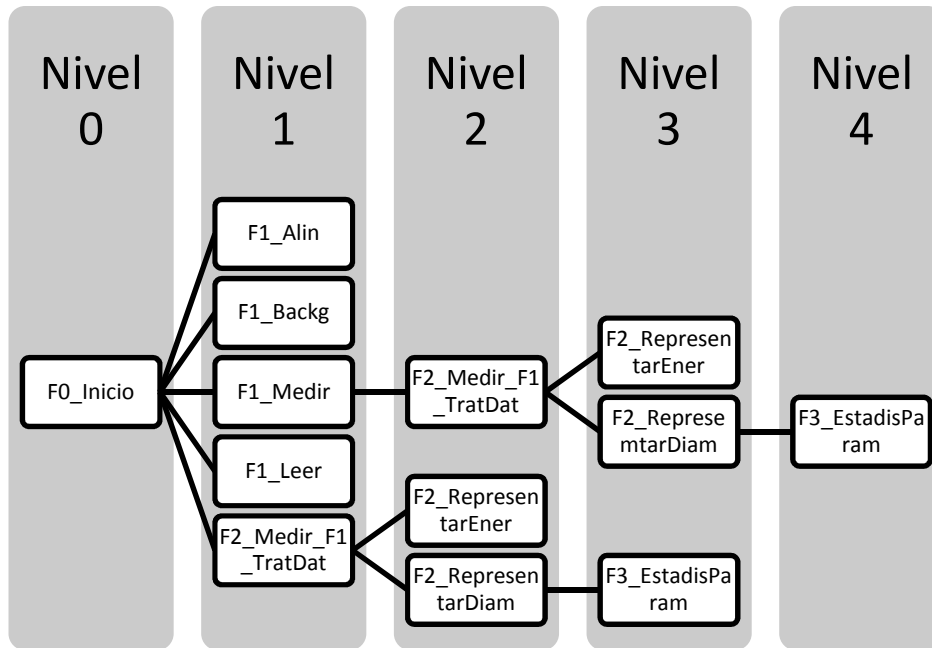


Figura 9. Jerarquía de formularios

Como se puede observar en la anterior figura, al formulario F2_Medir_F1_TratDat se puede acceder de dos manera distintas, a través de F0_Inicio y a través de F1_Medir, de ahí el nombre establecido. Además, la nomenclatura utilizada en la figura 9 es la misma que la utilizada en el código de programación.

A continuación, se muestra la descripción de cada uno de los formularios a los que se puede acceder a través de F0_Inicio (la ventana inicial del software).

Alinear: F1_Alin

Mediante este formulario se realiza el alineamiento del dispositivo. Lo que ofrece esta opción, es una medición en continuo para poder ajustar el alineamiento del haz gracias a unas ruletas disponibles en el difractor. En el gráfico de energía vs. anillo se observa la ajuste, básicamente hay que buscar que el porcentaje de energía incidente en el anillo central (anillo 0) sea lo mayor posible. La medición en continuo no es más que la realización de una medida detrás otra, mostrándola gráficamente, hasta que el usuario decide parar, porque ya considera que el alineamiento del dispositivo es el adecuado.

Esta es una imagen del formulario cuando se está realizando el alineamiento:

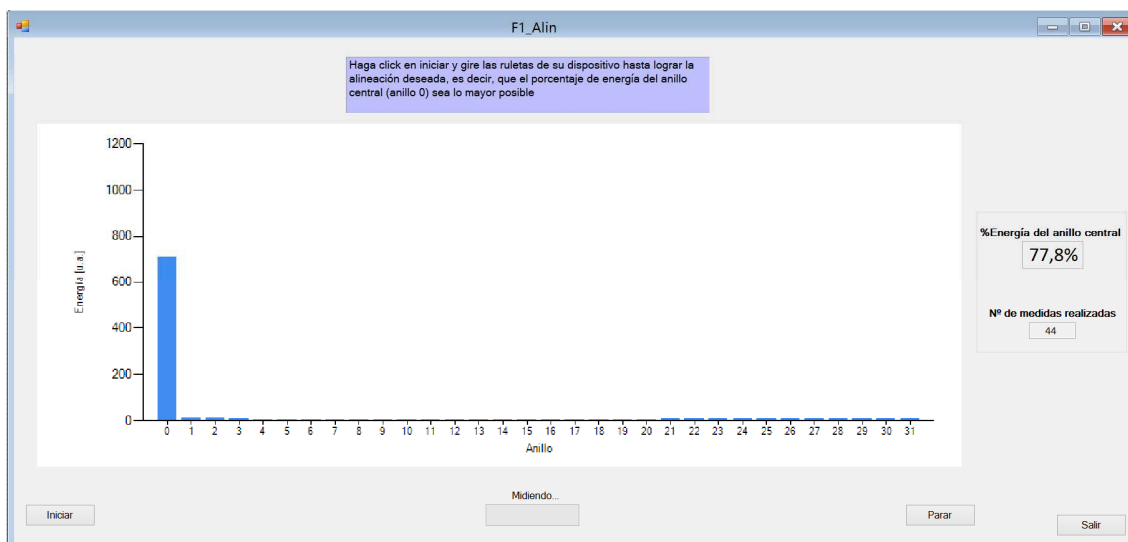


Figura 10. Formulario F1_Alin durante la alineación

Background: F1_Backg

A través del siguiente formulario se pueden realizar medidas del background. Para ello cuando se realiza una medida del background, esta consiste en una medida de 100 muestras mediante la cuales se calcula el promedio y finalmente dicho promedio es mostrado gráficamente.

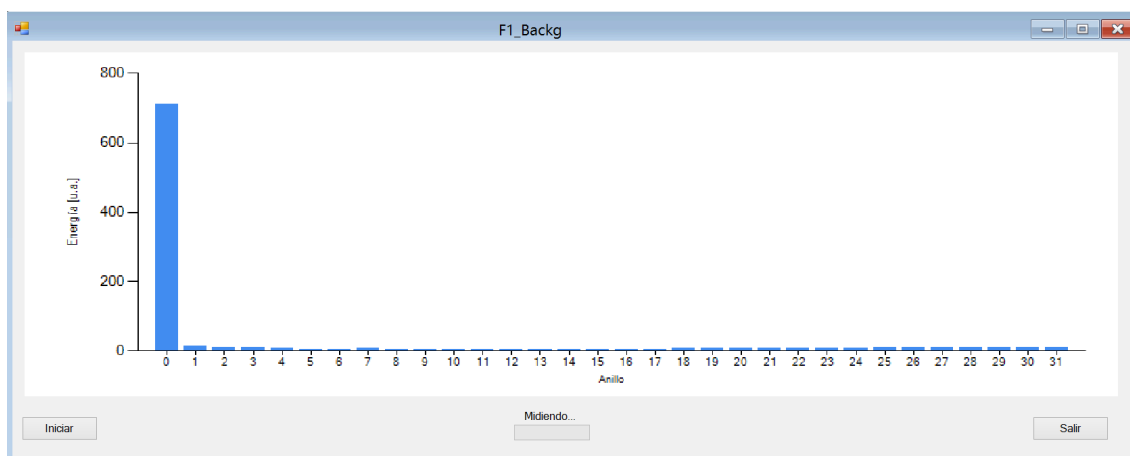


Figura 11. Formulario F1_Backg tras la medida del Background

Medir: F1: medir

El formulario medir ofrece al usuario las siguientes posibilidades:

- Establecer el nombre de la medida
- Fijar el número de muestras a realizar para una determinada medida
- Determinar el tipo de lente utilizada en la medición

Además, se pueden visualizar los nombres de todas las medidas realizadas o leídas en una sesión a través de un cuadro de texto, por lo que son medidas disponibles para un posterior tratamiento. A dicho tratamiento de datos se puede acceder directamente desde este formulario.

La siguiente figura muestra el diseño del formulario F1_Medir, con la cual se ve más claramente todo lo anteriormente mencionado.

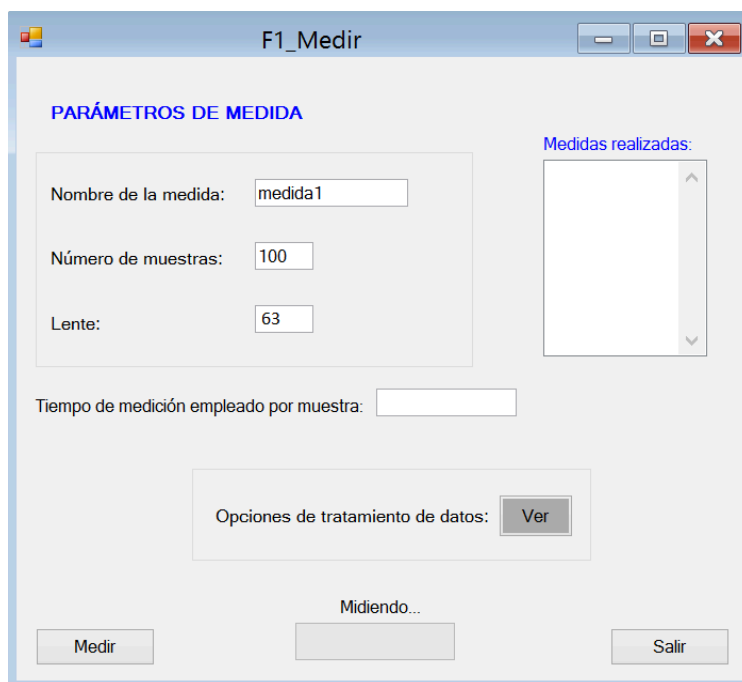


Figura 12. Formulario F1_Medir tras realizar una serie de medidas

Lectura: F1_Leer

Este formulario ofrece la posibilidad de leer archivos de texto de extensión (*.med) y guardar lo leído en una variable de tipo medida (esta última operación se realiza de forma interna, es decir, el usuario no interfiere en este hecho). Los archivos con este tipo de extensión son medidas realizadas y guardadas en anteriores sesiones.

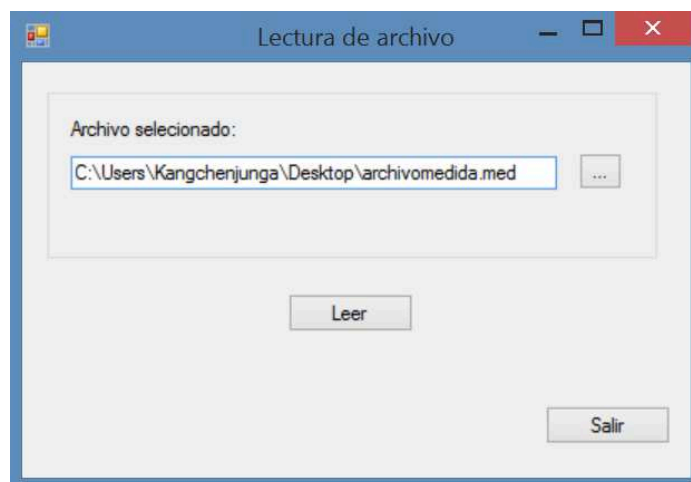


Figura 13. Formulario F1_Leer tras la lectura de un archivo

F2_Medir_F1_TratDat

El tratamiento de datos ofrece al usuario la posibilidad de seleccionar cualquier medida que haya sido realizada en la sesión actual a través de la opción medir o mediante la opción leer. Para el manejo del listado de medidas disponibles se usa la siguiente variable:

List<medida> listaMedidas

Se trata de una variable de tipo List, la cual representa una lista de objetos a los que se puede tener acceso por el índice. El tipo de objetos que almacena se define entre < >, en nuestro caso son de tipo medida.

Una vez seleccionada la medida deseada se ofrecen las siguientes opciones para la misma:

- Guardar la medida en un archivo de texto de extensión ".med".
- Calcular la distribución de diámetros eligiendo el número de anillos a desechar (puede ser cero).
- Representaciones gráficas:
 - Energías vs anillos
 - Probabilidad de clases de diámetro vs clases de diámetro

The screenshot shows a software window titled 'Tratamiento de datos'. Inside, there is a form titled 'OPCIONES'. The form has several sections: 'Seleccionar medida' with a dropdown menu showing 'Transp_DEF'; 'Guardar medida seleccionada' with a 'Guardar' button; 'Opciones de cálculo' with a 'Calcular' button and a sub-section titled 'Cálculo de la distribución de probabilidad de clases de diámetro' containing 'Lente utilizada' (set to 100) and 'Anillos desechados' (set to 1); 'Representación de datos' with a 'Representar' button and two checkboxes, 'Energías leídas' (checked) and 'Distribución de diámetros' (unchecked). A 'Salir' button is at the bottom right.

Figura 14. Formulario F2_MedirF1_TratDat

F2_RepresentarEner

Cuando este formulario es cargado, se visualiza por defecto la visualización gráfica de energías frente a anillos del promedio de energías. Sin embargo, si el usuario desea ver una muestra en concreto, lo puede hacer mediante la opción de " Selección de muestra" que aparece en la parte inferior central del formulario.

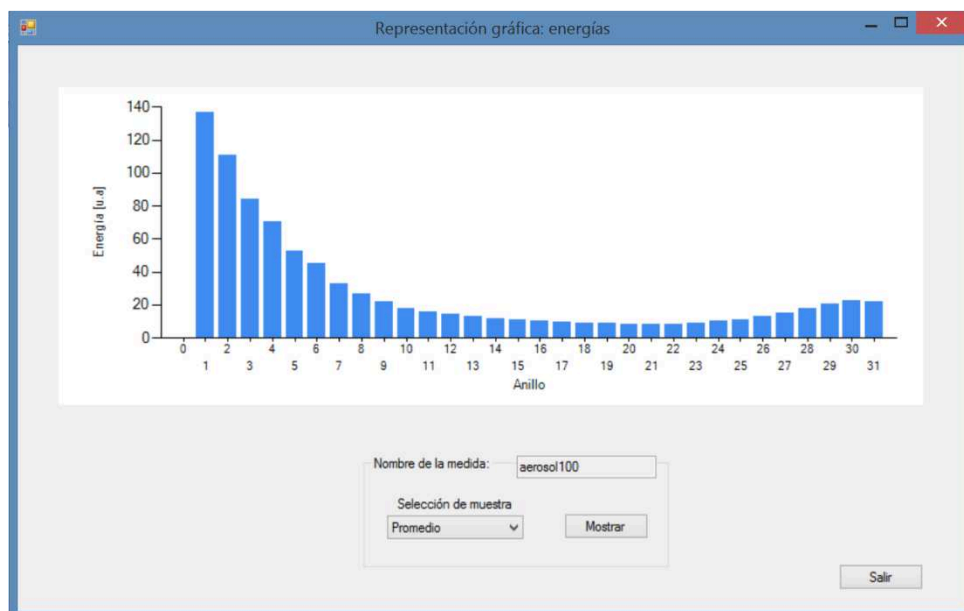


Figura 15. Formulario F2_RepresentarEner

F2_RepresentarDiam

Este formulario es idéntico al anterior salvo que se trata de una representación de frecuencias frente a clases de diámetro y además, a través de dicho formulario, se tiene acceso al formulario que muestra los estadísticos y otros parámetros de interés.

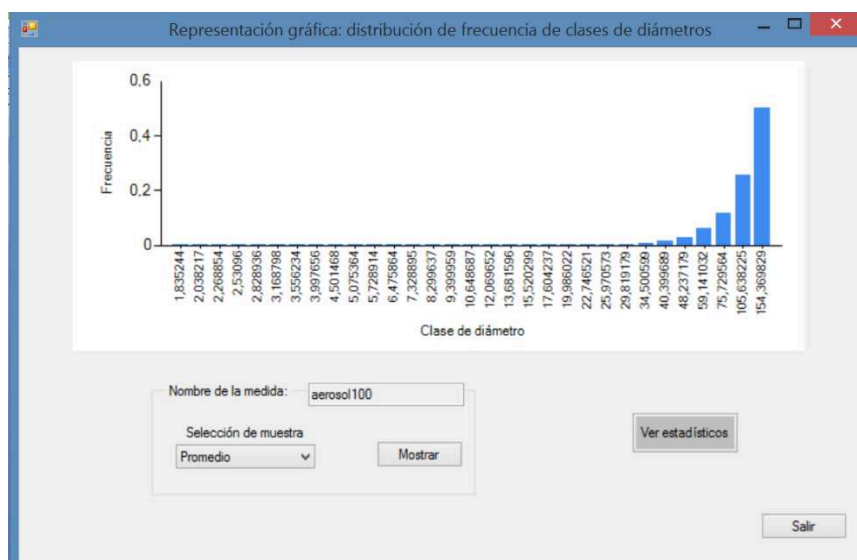


Figura 16. Formulario F2_RepresentarDiam

F3_EstadisYParam

Por último, este formulario presenta los siguientes parámetros:

- Diámetros significativos (estadísticos):

D_{10}

D_{30}

D_{32}

D_{43}

- Percentiles:

D[10]

D[50]

D[90]

- Span

- Transmisión

Se pueden mostrar dichos parámetros para cada una de las muestras realizadas, además de la promedia que es la que se muestra por defecto al cargar el formulario.

The screenshot shows a software window titled "Estadísticos y otros parámetros de interés". It contains two main sections: "Estadísticos" and "Percentiles".

Estadísticos		Percentiles		Otros parámetros	
D10	109,18	D[10]	56,59	Span	1,07
D30	1858976,67	D[50]	108,48	Transmisión	0,00
D32	133,22	D[90]	164,74		
D43	140,11				

Below the data tables, there is a section for "Nombre de la medida:" with the value "aerosol100". A "Selección de muestra:" dropdown menu is set to "Promedio". A "Mostrar" button is next to it. A "Salir" button is located at the bottom right.

Figura 17. Formulario F3_EstadisYParam.

5. Validación del software

La validación del software consta de las siguientes comprobaciones:

- Lectura correcta de los datos proporcionados por el difractómetro Malvern
- Cálculo de la distribución de probabilidad de clases de tamaño

Inicialmente, se propuso para dicha validación una medida de las energías obtenidas cuando se colocaba una instalación (atravesada por el haz láser) capaz de provocar cavitación en el agua mediante ultrasonidos y así generar burbujas, dado que se prevé utilizar este equipo para estudios de cavitación. De esta manera, se obtenía la energía difractada por las burbujas generadas por cavitación para, posteriormente, calcular la distribución de probabilidad de clases de diámetros. Pero dicha instalación posee una serie de problemas que provocan que los datos obtenidos no sean los adecuados para la validación del cálculo de la distribución de probabilidad de clases de tamaño.

Se realizaron lecturas con el difractómetro con un recipiente de forma de paralelepípedo rectangular lleno de agua que tiene en su base un oscilador acústico y que genera burbujas de cavitación. A pesar del correcto alineamiento de las paredes transparentes se tiene una significativa atenuación de la luz del haz láser y un aumento del ruido óptico (a pesar de la cuidadosa limpieza de las paredes se observaba dispersión de la luz en ellas). Dado que la cantidad de burbujas generadas era muy pequeña, junto con la atenuación comentada, no se tenía suficiente señal como para obtener distribuciones fiables de diámetro. Básicamente el problema se resume en que la señal no es mucho mayor que el ruido óptico y no se pudieron obtener distribuciones confiables.

La elaboración de una instalación capaz de reproducir la cavitación y generar burbujas de un diámetro y cantidad dentro del rango de medida del difractómetro será abordado en trabajos posteriores y no concierne al principal objetivo del presente proyecto.

Por lo tanto, para validar con la máxima precisión el funcionamiento completo del software, se ha creado una plantilla con partículas de diámetro de 130 micras que se ha impreso en una transparencia y colocada en el volumen de intersección del haz láser.

A continuación se muestra la plantilla de partículas utilizada:

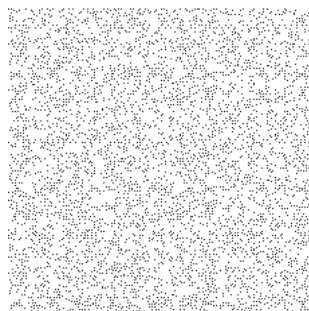


Figura 18. Plantilla de partículas de 130 micras de diámetro (patrón en transparencia)

Las ventajas de realizar una medida con la plantilla anteriormente mostrada, son las siguientes:

- A priori se sabe que se ha de obtener una distribución de probabilidad de clases de tamaño monomodal para verificar que el software realiza los cálculos correctamente.
- Se evitan problemas de conglomerados de partículas, ya que en la plantilla dichas partículas están perfectamente separadas las unas de las otras.
- Se consigue realizar una medida en continuo e invariante en el tiempo.

Para llevar a cabo esta validación, en primer lugar se muestra el resultado obtenido de las energías obtenidas:

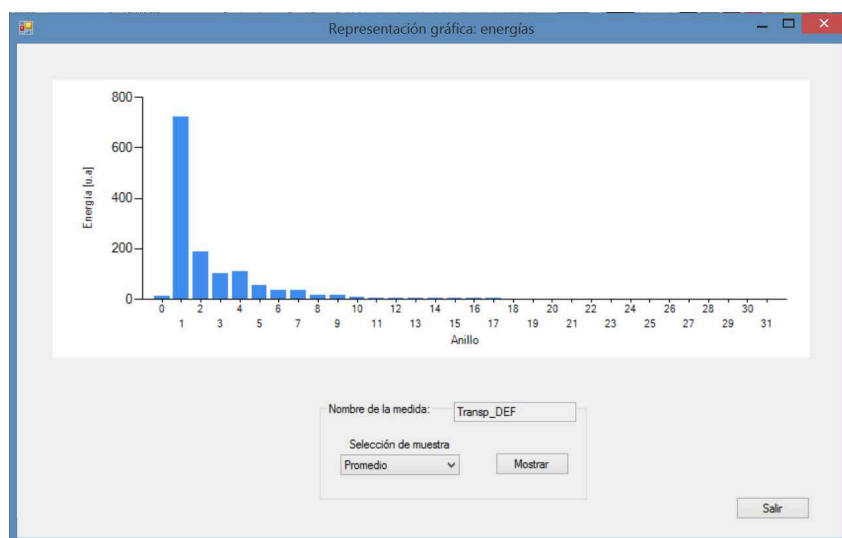


Figura 19. Representación gráfica de las energías leídas en el experimento de patrón en transparencia

En segundo lugar, se muestran los resultados obtenidos tras la realización del cálculo, empezando por la representación gráfica de la distribución de frecuencias de clases de tamaño:

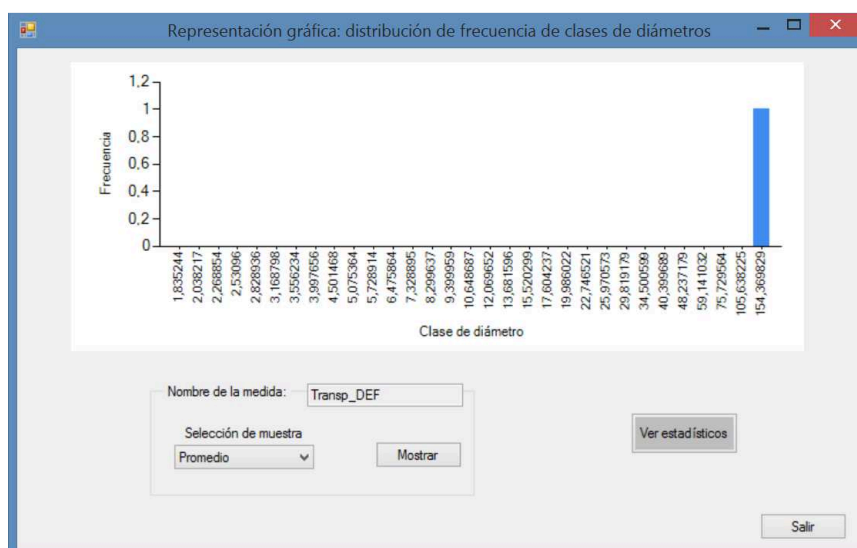


Figura 20 Distribución de frecuencia de clases de diámetros obtenida en el experimento de patrón en transparencia

También se ha contrastado esta medida con la obtenida con otro difractómetro (Spraytec) disponible en el LIFTEC (laboratorio de investigación en fluidodinámica y tecnologías de la combustión, centro mixto universidad de Zaragoza CSIC). La distribución obtenida se presenta en la siguiente figura y se observa que la mayor parte de la distribución aparece para las mismas clases de diámetro:

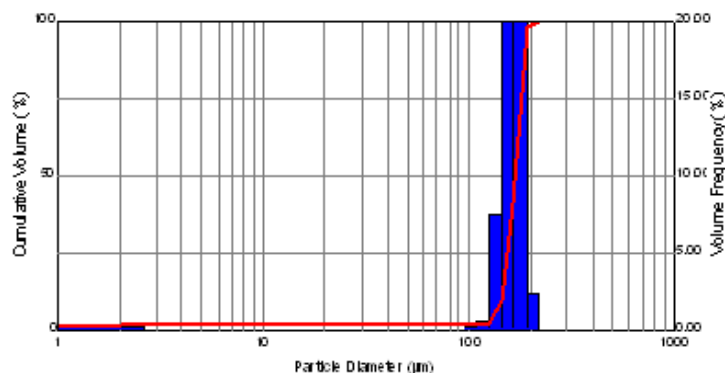


Figura 21. Distribución de frecuencias de clases de diámetros obtenida mediante el SprayTec del patrón en transparencia

La siguiente figura muestra los estadísticos y parámetros de interés obtenidos en esta medida:



Figura 22 Estadísticos y otros parámetros de interés obtenidos en el experimento de patrón en transparencia

Para la validación del software se ha realizado, además, una medida de un aerosol comercial para limpiar cristales. A continuación, se muestran los resultados obtenidos:

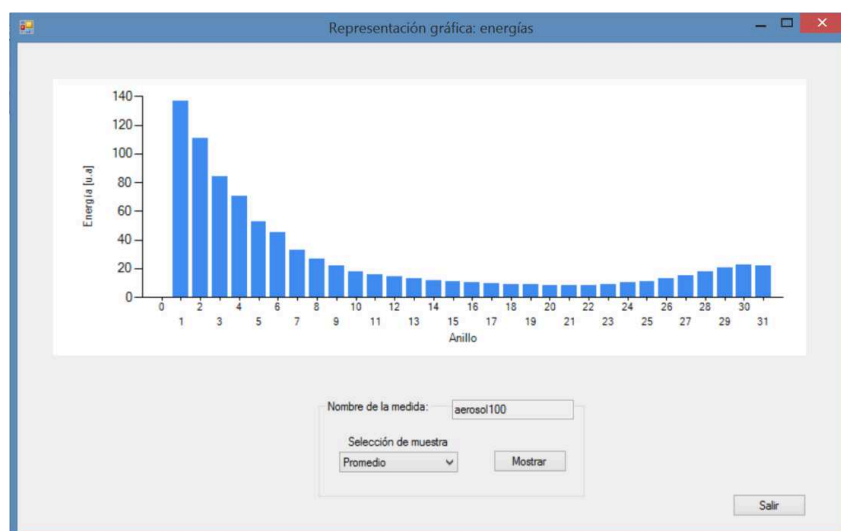


Figura 23. Representación gráfica de las energías leídas en el experimento del aerosol comercial



Figura 24. Distribución de frecuencia de clases de diámetros obtenida en el experimento del aerosol comercial

Contrastado, también, con la distribución de frecuencias de clases de diámetros obtenida mediante el difractómetro SprayTec:

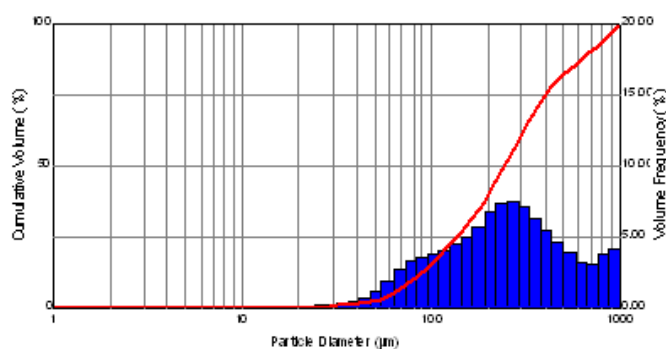


Figura 25. Distribución de frecuencias de clases de diámetros obtenida mediante el SprayTec del aerosol comercial



Se observa que el rango de clases de diámetros que contempla este difractor es mayor al contemplado en el Malvern (hasta 154 micras con la lente de 100). De ahí, que para realizar una comparativa correcta, únicamente se tenga en cuenta la tendencia obtenida mediante el Spraytec hasta las 200 micras.

De esta manera, se comprueba que el software programado funciona correctamente ya que:

- Para el patrón en transparencia se obtiene una distribución de probabilidad de diámetros monomodal correspondiente a la clase de tamaño que más se acerca a la real: 154 *micras* ~ 130 *micras*, validándose así los cálculos realizados.
- Se han comprobado que se obtienen los mismos resultados (en el rango abarcado por el Malvern) que los obtenidos mediante el difractor Spraytec.

A modo de ejemplo, en el apartado 1.3. del Anexo 1 se muestra el fichero obtenido del patrón en transparencia tras el cálculo y el guardado.

6. Conclusiones

En el presente proyecto se han cumplido todas las expectativas relacionadas con la implementación del nuevo sistema de adquisición de datos del Malvern HS-2600 y el desarrollo de los cálculos para obtener distribuciones de tamaño de gotas, partículas o burbujas.

Se ha escrito el código que realiza todas las funciones necesarias y se han calculado las clases de diámetro y matrices para cada una de las lentes disponibles para el difractor. También, se ha realizado el cableado de la tarjeta de adquisición y su adaptación al equipo.

En un principio se pretendía validar su aplicación para la medida de gotas producidas por cavitación. Con el dispositivo actual no ha sido posible obtener medidas de burbujas pero se ha validado el funcionamiento del sistema con un patrón de partículas impreso sobre una transparencia y con un aerosol comercial. En el futuro se mejorará la instalación experimental para conseguir un mejor nivel de señal y abordar la medida del diámetro de burbujas producidas por cavitación. En la verificación del sistema se ha comprobado que la adaptación del cableado, la lectura de las energías en los anillos y los cálculos son correctos por lo que la parte esencial del proyecto ha sido implementada correctamente.

Se dispone, por tanto, de un nuevo sistema de adquisición que funciona en los sistemas operativos actuales y que ofrece una gran versatilidad dado que el ordenador se comunica a través de un puerto USB, disponible en cualquier ordenador de uso común en nuestros días. Nos libera de la necesidad de mantener un ordenador desactualizado tanto en software como en hardware para poder operar con el difractor. De hecho, las medidas de validación se han adquirido con un ordenador portátil y es como se piensa trabajar en el futuro puesto que además se libera espacio en el laboratorio para el ordenador dedicado a la adquisición.

7. Bibliografía

1. Óptica. Justiniano Casas Pelaez ISBN-10: 8460500624
2. Bessel Functions: project for the Penn State - Göttingen Summer School on Number Theory. Martin Kreh
3. <http://msdn.microsoft.com/>
4. Malvern HS-2600 User Manual

Proyectos fin de carrera:

5. Título: Instalación para determinación de tamaño de partícula de un spray por difracción láser mediante aplicación de la transformación tomográfica.
Autores: Jorge Luis Lagranja Berges y Francisco Javier Vecino Soler
Director: César Dopazo García
Año: 1986



6. Título: Optimización de un equipo de difracción láser Malvern 2600 para la determinación dinámica de tamaños de partículas. Análisis del sistema. Mejoras al hardware y al software.

Autor: José Ignacio García Palacín

Director: Luis Aisa

Año: 1988

7. Título: Mejora de la resolución e interface de usuario del sistema de medida dinámica de gotas Malvern 2600.

Autor: Agustín Esbec Albalad

Director: José Ignacio García Palacín

Año: 1997

8. Anexos

Anexo1: Detalles del código del software MalvernStudio

1.1. Cálculos

```
//Vectores de energías
double[] Eleida_ad;//Teniendo en cuenta los anillos desechados y con
el anillo central eliminado
double[] Eleida_ad_norm;//Igual que la variable anterior solo que
ahora normalizada
double suma;

double[] clasesDiam;
double[] limclasesDiam;
double[,] matrizCoef;

int muestras, tiposDiam, tiposParam;
int f, c;
int pasos, pasos2;

double sumnum, sumden, perc, perc_ant, A, B, sumParam;

double error, error_ant;
double[] Dajust, Dajust_n;
double[] D_interm;
double[] Eajust, Eajust_n;

double[] coefsDiam;
coefsDiam = new double[31] { 235, 265.3, 299.6, 338.3, 382, 431.3,
487, 550, 621, 701, 791.5, 893.7, 1009, 1139.4, 1286.4, 1452.7, 1640.3, 1852,
2091, 2361, 2666.3, 3010.5, 3399.25, 338.2, 4333.8, 4893.43, 5525.3, 6238.8,
7500, 8500, 9500 };

textBoxProgress.Show();
textBoxProgress.Text = "Calculando...espere";

if (anillDesch == false)
{
    MessageBox.Show("Establece todas la opciones de cálculo antes de
comenzar con el cálculo");
}
else
{
    MessageBox.Show("Has establecido todas las opciones de cálculo");
    medidaSeleccionada.CtrlCalcDiam = true;
}

////////Cálculos////////
//Se calcula:
//distribuciones de diámetro (de todas las muestras y la promedio)
//estadísticos y parámetros de interés [9 en total] (de todas las
muestras y la promedio)

if (medidaSeleccionada.CtrlCalcDiam == true)
{
```



```
medidaSeleccionada.AnillosDesech =
Convert.ToInt16(textBoxAnillosDes.Text);

clasesDiam = new double[31];
matrizCoef = new double[31, 31];
limclasesDiam = new double[32];

medidaSeleccionada.nuevainstanciaMatricesDiam();
medidaSeleccionada.nuevainstanciaParam();

switch (medidaSeleccionada.Lente)
{
    case 63:
        clasesDiam = datos.clasesdiam63;
        limclasesDiam = datos.limclasesdiam63;
        matrizCoef = datos.matriz63;
        break;
    case 100:
        clasesDiam = datos.clasesdiam100;
        limclasesDiam = datos.limclasesdiam100;
        matrizCoef = datos.matriz100;
        break;
    case 300:
        clasesDiam = datos.clasesdiam300;
        limclasesDiam = datos.limclasesdiam300;
        matrizCoef = datos.matriz300;
        break;
    case 600:
        clasesDiam = datos.clasesdiam600;
        limclasesDiam = datos.limclasesdiam600;
        matrizCoef = datos.matriz600;
        break;
    case 800:
        clasesDiam = datos.clasesdiam800;
        limclasesDiam = datos.limclasesdiam800;
        matrizCoef = datos.matriz800;
        break;
    case 1000:
        clasesDiam = datos.clasesdiam1000;
        limclasesDiam = datos.limclasesdiam1000;
        matrizCoef = datos.matriz1000;
        break;
}

muestras = 0;
while (muestras < medidaSeleccionada.Longitud)
{
    //Obtención de E_leida_ad_norm\\
    Eleida_ad = new double[31];
    //Quitar anillo central
    for (f = 1; f < 32; f++)
    {
        Eleida_ad[f - 1] =
medidaSeleccionada.MatrizEnergias[muestras, f];
    }
    //Desechar anillos-->Poner a cero los elementos más próximos
al central
    for (f = 0; f < medidaSeleccionada.AnillosDesech; f++)
    {
        Eleida_ad[f] = 0;
    }
}
```

```
//Normalización de las energías leídas
Eleida_ad_norm = new double[31];
suma = 0;
for (f = 0; f < 31; f++)
{
    suma = suma + Eleida_ad[f];
}
for (f = 0; f < 31; f++)
{
    Eleida_ad_norm[f] = Eleida_ad[f] / suma;
}

//Cálculo de la Dajust_n inicial\\
Dajust = new double[31];
for (tiposDiam = 0; tiposDiam < 30; tiposDiam++)
{
    Dajust[30 - tiposDiam] = Eleida_ad_norm[tiposDiam] /
coefsDiam[tiposDiam];
}
Dajust[0] = 3 * Dajust[1] - 1.5 * Dajust[2];

//Guardado de la Dajust en la Dajust_n
Dajust_n = new double[31];
for (tiposDiam = 0; tiposDiam < 31; tiposDiam++)
{
    Dajust_n[tiposDiam] = Dajust[tiposDiam];
}

error = 9;
error_ant = 10;
pasos = 0; pasos2 = 0;
D_interm = new double[31];
Eajust_n = new double[31];

//Método iterativo\\
while (error < error_ant)
{
    //Normalización de la Dajust_n (se vuelve a normalizar en
el método iterativo)
    suma = 0;
    for (tiposDiam = 0; tiposDiam < 31; tiposDiam++)
    {
        suma += Dajust_n[tiposDiam];
    }
    for (tiposDiam = 0; tiposDiam < 31; tiposDiam++)
    {
        Dajust_n[tiposDiam] = Dajust_n[tiposDiam] / suma;
    }
    //Guardado de la Dajust_n en la Dinterm
    for (tiposDiam = 0; tiposDiam < 31; tiposDiam++)
    {
        D_interm[tiposDiam] = Dajust_n[tiposDiam];
    }
    //Se guarda en la Dajust la segunda corrección
    for (tiposDiam = 0; tiposDiam < 31; tiposDiam++)
    {
        Dajust_n[tiposDiam] = Dajust_n[tiposDiam] *
Dajust_n[tiposDiam];
    }
    //Cálculo de la E_ajust
    Eajust = new double[31];
    for (f = 0; f < 31; f++)
```

```

        {
            Eajust[f] = 0;
            for (c = 0; c < 31; c++)
            {
                Eajust[f] += matrizCoef[f, c] * Dajust_n[c];
            }
        }
        //Cálculo de la E_ajust_n
        for (f = 0; f < 31; f++) Eajust_n[f] = 0;
        suma = 0;
        for (f = 0; f < 31; f++)
        {
            suma += Eajust[f];
        }
        for (f = 0; f < 31; f++)
        {
            Eajust_n[f] = Eajust[f] / suma;
        }

        //Cálculo del error
        error_ant = error;
        error = 0;
        for (f = 0; f < 31; f++)
        {
            error += Math.Pow((Eleida_ad_norm[f] - Eajust_n[f]),
2);
        }
        pasos++;

    } //Fin del while (error < error_ant)

    //SEGUNDO BUCLE
    while (error < error_ant)
    {
        //Normalización de la Dajust_n (se vuelve a normalizar en
el método iterativo)
        suma = 0;
        for (tiposDiam = 0; tiposDiam < 31; tiposDiam++)
        {
            suma += Dajust_n[tiposDiam];
        }
        for (tiposDiam = 0; tiposDiam < 31; tiposDiam++)
        {
            Dajust_n[tiposDiam] = Dajust_n[tiposDiam] / suma;
        }
        //Guardado de la Dajust_n en la Dinterm
        for (tiposDiam = 0; tiposDiam < 31; tiposDiam++)
        {
            D_interm[tiposDiam] = Dajust_n[tiposDiam];
        }
        //Se guarda en la Dajust el cuadrado de la misma
        for (tiposDiam = 30; tiposDiam > 0; tiposDiam--)
        {
            Dajust_n[tiposDiam] = Dajust_n[tiposDiam] *
(Math.Pow(Eleida_ad_norm[tiposDiam] / Eajust_n[tiposDiam], 2.0) +
Math.Pow(Eleida_ad_norm[tiposDiam - 1] / Eajust_n[tiposDiam - 1], 2.0));
        }
        Dajust_n[0] = 2.0 * Dajust_n[0] *
Math.Pow(Eleida_ad_norm[0] / Eajust_n[0], 2.0);
        //Cálculo de la E_ajust
        Eajust = new double[31];
        for (f = 0; f < 31; f++)
    
```

```

        {
            Eajust[f] = 0;
            for (c = 0; c < 31; c++)
            {
                Eajust[f] += matrizCoef[f, c] * Dajust_n[c];
            }
        }
        //Cálculo de la E_ajust_n
        for (f = 0; f < 31; f++) Eajust_n[f] = 0;
        suma = 0;
        for (f = 0; f < 31; f++)
        {
            suma += Eajust[f];
        }
        for (f = 0; f < 31; f++)
        {
            Eajust_n[f] = Eajust[f] / suma;
        }

        //Cálculo del error
        error_ant = error;
        error = 0;
        for (f = 0; f < 31; f++)
        {
            error += Math.Pow((Eleida_ad_norm[f] - Eajust_n[f]),
2);
        }
        pasos2++;
    }
    //FIN DEL SEGUNDO BUCLE

    //Guardado de la D_interm obtenida en MATRIZDIAMETROS\\
    for (tiposDiam = 0; tiposDiam < 31; tiposDiam++)
    {
        medidaSeleccionada.MatrizDiametros[muestras, tiposDiam] =
D_interm[tiposDiam];
    }

    //Dnm's: ESTADÍSTICOS\\

    //Dnm10
    sumnum = 0;
    sumden = 0;
    for (tiposDiam = 0; tiposDiam < 31; tiposDiam++)
    {
        sumnum += medidaSeleccionada.MatrizDiametros[muestras,
tiposDiam] * Math.Pow(clasesDiam[tiposDiam], 1);
        sumden += medidaSeleccionada.MatrizDiametros[muestras,
tiposDiam] * Math.Pow(clasesDiam[tiposDiam], 0);
    }
    medidaSeleccionada.MatrizParam[muestras, 0] = sumnum /
sumden;

    //Dnm30
    sumnum = 0;
    sumden = 0;
    for (tiposDiam = 0; tiposDiam < 31; tiposDiam++)
    {
        sumnum += medidaSeleccionada.MatrizDiametros[muestras,
tiposDiam] * Math.Pow(clasesDiam[tiposDiam], 3);
    }

```

```
        sumden += medidaSeleccionada.MatrizDiametros[muestras,
tiposDiam] * Math.Pow(clasesDiam[tiposDiam], 0);
    }
    medidaSeleccionada.MatrizParam[muestras, 1] = sumnum /
sumden;

    //Dnm32
    sumnum = 0;
    sumden = 0;
    for (tiposDiam = 0; tiposDiam < 31; tiposDiam++)
    {
        sumnum += medidaSeleccionada.MatrizDiametros[muestras,
tiposDiam] * Math.Pow(clasesDiam[tiposDiam], 3);
        sumden += medidaSeleccionada.MatrizDiametros[muestras,
tiposDiam] * Math.Pow(clasesDiam[tiposDiam], 2);
    }
    medidaSeleccionada.MatrizParam[muestras, 2] = sumnum /
sumden;

    //Dnm43
    sumnum = 0;
    sumden = 0;
    for (tiposDiam = 0; tiposDiam < 31; tiposDiam++)
    {
        sumnum += medidaSeleccionada.MatrizDiametros[muestras,
tiposDiam] * Math.Pow(clasesDiam[tiposDiam], 4);
        sumden += medidaSeleccionada.MatrizDiametros[muestras,
tiposDiam] * Math.Pow(clasesDiam[tiposDiam], 3);
    }
    medidaSeleccionada.MatrizParam[muestras, 3] = sumnum /
sumden;

    //PERCENTILES\\
    //D[10]
    perc = 0;
    perc_ant = 0;
    tiposDiam = 0;
    while ((perc < 0.1) && (tiposDiam < 31))
    {
        perc_ant = perc;
        perc += medidaSeleccionada.MatrizDiametros[muestras,
tiposDiam];
        tiposDiam++;
    }

    A = (perc - perc_ant) / (limclasesDiam[tiposDiam] -
limclasesDiam[tiposDiam - 1]);
    B = perc - A * limclasesDiam[tiposDiam];

    medidaSeleccionada.MatrizParam[muestras, 4] = (0.1 - B) / A;

    //D[50]
    perc = 0;
    perc_ant = 0;
    tiposDiam = 0;
    while ((perc < 0.5) && (tiposDiam < 31))
    {
        perc_ant = perc;
        perc += medidaSeleccionada.MatrizDiametros[muestras,
tiposDiam];
        tiposDiam++;
    }
}
```

```
A = (perc - perc_ant) / (limclasesDiam[tiposDiam] -  
limclasesDiam[tiposDiam - 1]);  
B = perc - A * limclasesDiam[tiposDiam];  
  
medidaSeleccionada.MatrizParam[muestras, 5] = (0.5 - B) / A;  
  
//D[90]  
perc = 0;  
perc_ant = 0;  
tiposDiam = 0;  
while ((perc < 0.9) && (tiposDiam < 31))  
{  
    perc_ant = perc;  
    perc += medidaSeleccionada.MatrizDiametros[muestras,  
tiposDiam];  
    tiposDiam++;  
}  
  
A = (perc - perc_ant) / (limclasesDiam[tiposDiam] -  
limclasesDiam[tiposDiam - 1]);  
B = perc - A * limclasesDiam[tiposDiam];  
  
medidaSeleccionada.MatrizParam[muestras, 6] = (0.9 - B) / A;  
  
//Span  
  
medidaSeleccionada.MatrizParam[muestras, 7] =  
(medidaSeleccionada.MatrizParam[muestras, 6] -  
medidaSeleccionada.MatrizParam[muestras, 4]) /  
medidaSeleccionada.MatrizParam[muestras, 5];  
  
//Transmisión  
medidaSeleccionada.MatrizParam[muestras, 8] =  
medidaSeleccionada.MatrizEnergias[muestras, 0] / medidaSeleccionada.Bckg[0];  
  
muestras++;  
  
} //Fin del while (muestras < medidaSeleccionada.Longitud)  
  
//Cálculo del VECTOR PROMEDIO de los parámetros ESTADÍSTICOS\\  
  
for (tiposParam = 0; tiposParam < 9; tiposParam++)  
{  
    sumParam = 0;  
    for (muestras = 0; muestras < medidaSeleccionada.Longitud;  
muestras++)  
    {  
        sumParam += medidaSeleccionada.MatrizParam[muestras,  
tiposParam];  
    }  
    medidaSeleccionada.PromParam[tiposParam] = sumParam /  
medidaSeleccionada.Longitud;  
}  
  
//CÁLCULO DEL VECTOR DIÁMETROS PROMEDIO\\  
  
//Obtención de E_leida_ad\\  
Eleida_ad = new double[31];
```

```
//Quitar anillo central
for (f = 1; f < 32; f++)
{
    Eleida_ad[f - 1] = medidaSeleccionada.PromEnerg[f];
}

//Desechar anillos-->Poner a cero los elementos más próximos al
central
for (f = 0; f < medidaSeleccionada.AnillosDesech; f++)
{
    Eleida_ad[f] = 0;
}
//Normalización de las energías leídas
Eleida_ad_norm = new double[31];
suma = 0;
for (f = 0; f < 31; f++)
{
    suma = suma + Eleida_ad[f];
}
for (f = 0; f < 31; f++)
{
    Eleida_ad_norm[f] = Eleida_ad[f] / suma;
}

//Cálculo de la Dajust_n inicial\\
Dajust = new double[31];
for (tiposDiam = 0; tiposDiam < 30; tiposDiam++)
{
    Dajust[30 - tiposDiam] = Eleida_ad_norm[tiposDiam] /
coefsDiam[tiposDiam];
}
Dajust[0] = 3 * Dajust[1] - 1.5 * Dajust[2];

//Guardado de la Dajust en la Dajust_n
Dajust_n = new double[31];
for (tiposDiam = 0; tiposDiam < 31; tiposDiam++)
{
    Dajust_n[tiposDiam] = Dajust[tiposDiam];
}

error = 9;
error_ant = 10;
pasos = 0; pasos2 = 0;
D_interm = new double[31];
Eajust_n = new double[31];

//Método iterativo
while (error < error_ant)
{
    //Normalización de la Dajust_n
    suma = 0;
    for (tiposDiam = 0; tiposDiam < 31; tiposDiam++)
    {
        suma += Dajust_n[tiposDiam];
    }
    for (tiposDiam = 0; tiposDiam < 31; tiposDiam++)
    {
        Dajust_n[tiposDiam] = Dajust_n[tiposDiam] / suma;
    }
    //Guardado de la Dajust_n en la Dinterm
    for (tiposDiam = 0; tiposDiam < 31; tiposDiam++)
```

```
{
    D_interm[tiposDiam] = Dajust_n[tiposDiam];
}
//Se guarda en la Dajust el cuadrado de la misma
for (tiposDiam = 0; tiposDiam < 31; tiposDiam++)
{
    Dajust_n[tiposDiam] = Dajust_n[tiposDiam] *
Dajust_n[tiposDiam];
}

//Cálculo de la E_ajust
Eajust = new double[31];

for (f = 0; f < 31; f++)
{
    Eajust[f] = 0;
    for (c = 0; c < 31; c++)
    {
        Eajust[f] += matrizCoef[f, c] * Dajust_n[c];
    }
}
//Cálculo de la E_ajust_n
for (f = 0; f < 31; f++) Eajust_n[f] = 0;
suma = 0;
for (f = 0; f < 31; f++)
{
    suma += Eajust[f];
}
for (f = 0; f < 31; f++)
{
    Eajust_n[f] = Eajust[f] / suma;
}

//Cálculo del error
error_ant = error;
error = 0;
for (f = 0; f < 31; f++)
{
    error += Math.Pow((Eleida_ad_norm[f] - Eajust_n[f]), 2);
}
pasos++;
} //Fin del while (error < error_ant)

//SEGUNDO BUCLE
while (error < error_ant)
{
    //Normalización de la Dajust_n (se vuelve a normalizar en el
método iterativo)
    suma = 0;
    for (tiposDiam = 0; tiposDiam < 31; tiposDiam++)
    {
        suma += Dajust_n[tiposDiam];
    }
    for (tiposDiam = 0; tiposDiam < 31; tiposDiam++)
    {
        Dajust_n[tiposDiam] = Dajust_n[tiposDiam] / suma;
    }
    //Guardado de la Dajust_n en la Dinterm
    for (tiposDiam = 0; tiposDiam < 31; tiposDiam++)
    {
        D_interm[tiposDiam] = Dajust_n[tiposDiam];
    }
}
```



```
//Se guarda en la Dajust el cuadrado de la misma
for (tiposDiam = 30; tiposDiam > 0; tiposDiam--)
{
    Dajust_n[tiposDiam] = Dajust_n[tiposDiam] *
(Math.Pow(Eleida_ad_norm[tiposDiam] / Eajust_n[tiposDiam], 2.0) +
Math.Pow(Eleida_ad_norm[tiposDiam - 1] / Eajust_n[tiposDiam - 1], 2.0));
}
Dajust_n[0] = 2.0 * Dajust_n[0] * Math.Pow(Eleida_ad_norm[0]
/ Eajust_n[0], 2.0);
//Cálculo de la E_ajust
Eajust = new double[31];
for (f = 0; f < 31; f++)
{
    Eajust[f] = 0;
    for (c = 0; c < 31; c++)
    {
        Eajust[f] += matrizCoef[f, c] * Dajust_n[c];
    }
}
//Cálculo de la E_ajust_n
for (f = 0; f < 31; f++) Eajust_n[f] = 0;
suma = 0;
for (f = 0; f < 31; f++)
{
    suma += Eajust[f];
}
for (f = 0; f < 31; f++)
{
    Eajust_n[f] = Eajust[f] / suma;
}

//Cálculo del error
error_ant = error;
error = 0;
for (f = 0; f < 31; f++)
{
    error += Math.Pow((Eleida_ad_norm[f] - Eajust_n[f]), 2);
}
pasos2++;

}
//FIN DEL SEGUNDO BUCLE

//Guardado de la D_interm obtenida en MatrizDiametros
for (tiposDiam = 0; tiposDiam < 31; tiposDiam++)
{
    medidaSeleccionada.PromDiam[tiposDiam] = D_interm[tiposDiam];
}

} //Fin if (medidaSeleccionada.CtrlCalcDiam == true)

MessageBox.Show("Cálculos finalizados");
textBoxProgress.Hide();
```

1.2. Representaciones gráficas

A continuación se presenta el código asociado a las gráficas que aparecen en los distintos formularios.

Alinear

```
for (i = 0; i < 32; i++)
{
    this.ChartAlin.Series["SeriesAlin"].Points.AddXY(i,
AlgoritmoComunicacion.energias[i]);
}
```

Background

```
for (anillo = 0; anillo < 32; anillo++)
{
    this.chartBckg.Series["SeriesBckg"].Points.AddXY(anillo,
BckgTravel[anillo]);
}
```

Tratamiento de datos-Representación de energías leídas

```
if (muestraindexSelec < medidaSeleccionada.Longitud)
{
    for (anillo = 0; anillo < 32; anillo++)
    {
        this.chartRepEner.Series["SeriesRepEner"].Points.AddXY(anillo,
medidaSeleccionada.MatrizEnergias[muestraindexSelec, anillo]);
    }
    else//Solo entra cuando (indexSelec < medidaSeleccionada.Longitud)
que es cuando se selecciona el promedio
    {
        for (anillo=0;anillo<32;anillo++)
        {
            this.chartRepEner.Series["SeriesRepEner"].Points.AddXY(anillo,medidaSeleccionada.
PromEner[anillo]);
        }
    }
}
```

En este caso se tiene en cuenta que según la muestra seleccionada se toman unos datos u otros. Este hecho se tiene en cuenta mediante la variable muestraindexSelec.

Tratamiento de datos-Representación de distribución de diámetros

```
if (muestraindexSelec < medidaSeleccionada.Longitud)
{
    for (i = 0; i < 31; i++)
    {
        this.chartRepDiam.Series["SeriesRepDiam"].Points.AddXY(tiposDiam[i],
        medidaSeleccionada.MatrizDiametros[muestraindexSelec, i]);
    }
}
if (muestraindexSelec == (medidaSeleccionada.Longitud))
{
    for (i = 0; i < 31; i++)
    {
        this.chartRepDiam.Series["SeriesRepDiam"].Points.AddXY(tiposDiam[i],
        medidaSeleccionada.PromDiam[i]);
    }
}
```

Donde previamente se han asignado los tiposDiam según la siguiente secuencia de código:

```
switch (medidaSeleccionada.Lente)
{
    case 63:
        tiposDiam = datos.clasesdiam63;
        break;
    case 100:
        tiposDiam = datos.clasesdiam100;
        break;
    case 300:
        tiposDiam = datos.clasesdiam300;
        break;
    case 600:
        tiposDiam = datos.clasesdiam600;
        break;
    case 800:
        tiposDiam = datos.clasesdiam800;
        break;
    case 1000:
        tiposDiam = datos.clasesdiam1000;
        break;
}
```

Esto se debe a que según la lente utilizada, las clases de diámetros a tener en cuenta son unas u otras, es decir, la numeración del eje X cambia según la lente.

Finalmente, se muestra una descripción de los principales métodos usados en la representación de gráficas que ofrece el software.

`this.Chart.Series["Serie0"].Points.AddXY(x,y)`-->agrega el punto (X,Y) especificado a la "Serie0" de la gráfica "Chart".

`this.ChartAlin.Series["Serie0"].Points.Clear()`-->quita todos los elementos de la "Serie0" de la gráfica "Chart".

1.3. Guardado y lectura de archivos

Ejemplo de guardado en un archivo

#####

Información de la Medida de energías

#####

#Nombre de la medida

Nombre transp130micras

#Fecha y hora

FechaHora 13/08/2014 14:55:47

#Número de muestras

Longitud 1

#Lente utilizada

Lente 100

#Background

Bckg

19,41 150,18 110,68 108,35 98,02 98,58 99,63 106,46 103,56 113,91 115,7 125,32 129,62
142,8 158,92 171,44 179,6 188,46 190,59 193,01 191,52 187,52 181,7 180,17 172,84 166,58
164,1 159,44 154,12 155,38 154,84 153,49

#Energías

MatrizEnergias

7,150356 442,8036648 120,5121312 69,107579 76,4313344 37,7736774 24,5626992
26,300735 11,704976 11,5448825 7,560925 0,7000664 3,607745 0,37674 0,0484008
0,0140112 0,00736 0,0050976 0 0 0 0 0 0 0 0 0 0 0

PromEnerg

7,150356 442,8036648 120,5121312 69,107579 76,4313344 37,7736774 24,5626992
26,300735 11,704976 11,5448825 7,560925 0,7000664 3,607745 0,37674 0,0484008
0,0140112 0,00736 0,0050976 0 0 0 0 0 0 0 0 0 0 0

Control del cálculo de la distribución de probabilidad

#0 = Sí/Cálculo realizado(False)

#1 = No/Cálculo no realizado(True)

CtrlCalcDiam 1

Datos calculados

#Anillos desechados y probabilidad de clases de diámetro

AnillosDesech 0

MatrizDiametros

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3,92745082390525E-298 1,01967615696825E-237
1,50538227236881E-171          9,52528788098987E-214          3,11650868781228E-144
4,22564151653234E-129          2,41728356270771E-125          1,84891571635775E-99
5,52853914647458E-98 1,19917847216484E-82 1,10884383792015E-59 4,18594612123196E-
59 2,86821062508805E-40 1
```

PromDiam

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3,92745082390525E-298 1,01967615696825E-237
1,50538227236881E-171          9,52528788098987E-214          3,11650868781228E-144
4,22564151653234E-129          2,41728356270771E-125          1,84891571635775E-99
5,52853914647458E-98 1,19917847216484E-82 1,10884383792015E-59 4,18594612123196E-
59 2,86821062508805E-40 1
```

#Resultados estadísticos y otros parámetros de interés

#Parámetros calculados para cada muestra (disposición de una fila):

{D10, D30, D32, D43, D[10], D[50], D[90], Span, Transmisión}

MatrizParam

```
154,369829 3678639,83362683 154,369829 154,369829 131,15133591 154,36982875
177,58832159 0,300816461714187 0,368385162287481
```

PromParam

```
154,369829 3678639,83362683 154,369829 154,369829 131,15133591 154,36982875
177,58832159 0,300816461714187 0,368385162287481
```

Guardado de un archivo

```
textBoxGuardar.Show();
textBoxGuardar.Text = "Guardando datos...espere";
SaveFileDialog dlgsave = new SaveFileDialog();
dlgsave.FileName = medidaSeleccionada.Nombre;
dlgsave.Filter = "Archivos MalvernStudio (.med)|*.med";

if (dlgsave.ShowDialog() == DialogResult.OK)
{
    ruta = dlgsave.FileName;

    string[] plantillaI
={ "#####",
    "# Información de la Medida de energías",

    "#####", "\r",
    "#Nombre de la medida",
    "Nombre" + " " + medidaSeleccionada.Nombre, "\r",
    "#Fecha y hora",
    "FechaHora" + " " +
medidaSeleccionada.FechaHora, "\r",
    "#Número de muestras",
    "Longitud" + " " +
medidaSeleccionada.Longitud, "\r",
    "#Lente utilizada",
    "Lente" + " " + medidaSeleccionada.Lente, "\r",
    "#Background",
    "Bckg"};

    File.WriteAllLines(ruta, plantillaI);

    //Vector Background
    for (c = 0; c < 32; c++)
    {
        File.AppendAllText(ruta, medidaSeleccionada.Bckg[c] + " ");
    }
    File.AppendAllText(ruta,
Environment.NewLine + Environment.NewLine);

File.AppendAllText(ruta, "#Energías" + Environment.NewLine + "MatrizEnergias" + Environm
ent.NewLine);

    //Matriz energías
    for (f=0; f<medidaSeleccionada.Longitud; f++)
    {
        for (c=0; c<32; c++)
        {
            File.AppendAllText(ruta,
medidaSeleccionada.MatrizEnergias[f,c] + " ");
        }
        File.AppendAllText(ruta, Environment.NewLine);
    }

    File.AppendAllText(ruta, Environment.NewLine +
"PromEnerg" + Environment.NewLine);
    //Matriz promedio de energías
    for (c = 0; c < 32; c++)
    {
        File.AppendAllText(ruta, medidaSeleccionada.PromEnerg[c] + "
");
    }
}
```

```
File.AppendAllText(ruta,
Environment.NewLine+Environment.NewLine);

//Hay que hacer conversión de tipo para que lo escriba como uno o
cero en vez de con true o false
int ctrlcalc;
ctrlcalc = Convert.ToInt16(medidaSeleccionada.CtrlCalcDiam);

//Control de cálculos
string[] plantillaIII = { "#### Control del cálculo de la
distribución de probabilidad", "\r",
"#0 = Sí/Cálculo realizado(False)",
"#1 = No/Cálculo no realizado(True)",
"CtrlCalcDiam" + " " + ctrlcalc, "\r",
};
File.AppendAllLines(ruta, plantillaIII);

//Matrices Diámetros
if (medidaSeleccionada.CtrlCalcDiam == true)
{
File.AppendAllText(ruta, "#### Datos calculados" +
Environment.NewLine + Environment.NewLine + "#Anillos desechados y probabilidad
de clases de diámetro");

File.AppendAllText(ruta, Environment.NewLine +
Environment.NewLine + "AnillosDesech" + " " + medidaSeleccionada.AnillosDesech +
Environment.NewLine +
"MatrizDiametros" + Environment.NewLine);
//MatrizDiam
for (f = 0; f < medidaSeleccionada.Longitud; f++)
{
for (c = 0; c < 31; c++)
{
File.AppendAllText(ruta,
medidaSeleccionada.MatrizDiametros[f, c] + " ");
}
File.AppendAllText(ruta, Environment.NewLine);
}
File.AppendAllText(ruta, "PromDiam" + Environment.NewLine);
//PromDiam
for (c = 0; c < 31; c++)
{
File.AppendAllText(ruta, medidaSeleccionada.PromDiam[c] +
" ");
}

//Estadísticos y otros parámetros de interés

File.AppendAllText(ruta, Environment.NewLine +
Environment.NewLine + "#Resultados estadísticos y otros parámetros de interés" +
Environment.NewLine +
"#Parámetros calculados para cada
muestra (disposición de una fila):" + Environment.NewLine +
"{D10, D30, D32, D43, D[10], D[50],
D[90], Span, Transmisión}");
```

```
//MatrizParam
File.AppendAllText(ruta, Environment.NewLine +
Environment.NewLine + "MatrizParam" + Environment.NewLine);
for (f = 0; f < medidaSeleccionada.Longitud; f++)
{
    for (c = 0; c < 9; c++)
    {
        File.AppendAllText(ruta,
medidaSeleccionada.MatrizParam[f, c] + " ");
    }
    File.AppendAllText(ruta, Environment.NewLine);
}
File.AppendAllText(ruta, "PromParam" + Environment.NewLine);
//PromParam
for (c = 0; c < 9; c++)
{
    File.AppendAllText(ruta, medidaSeleccionada.PromParam[c]
+ " ");
}

} //Fin del if (medidaSeleccionada.CtrlCalcDiam == true)

MessageBox.Show("Datos guardados");
} // Fin del if (dlgsave.ShowDialog() == DialogResult.OK)

textBoxGuardar.Hide();
```


Lectura de un archivo

medida medidaRealizada = new medida();//En realidad es medida leída pero mantengo la misma nomenclatura que cuando se realiza la medida ya que el tratamiento es el mismo

```
try
{
    textBoxLeyendo.Show();
    textBoxLeyendo.Text = "Leyendo...espere";
    MessageBox.Show("Comienza la lectura");

    lectura = File.ReadAllText(FilePath);//Abre un archivo de texto,
    lee todas sus líneas y a continuación, cierra el archivo

    char[] delimiters = new char[] { ' ', '\n', '\t', '\r', '\v' };
    //Aquí se parten las líneas en partes separadas por los
    delimitadores y no se incluyen los elementos
    //vacíos (no confundir con el espacio), en nuestro caso de trata
    de los delimitadores: tab's, tab's verticales, espacios, newlines y retornos de
    carro

    parts = lectura.Split(delimiters,
StringSplitOptions.RemoveEmptyEntries);//Cada Split es un string y parts es un
array de strings
    nparts = parts.Length;

    rutavacia = false;
}
catch (Exception)
{
    textBoxLeyendo.Hide();
    MessageBox.Show("No se ha seleccionado ningún archivo para leer.
Haz click en el botón de examinar para elegir un archivo de lectura", "AVISO",
MessageBoxButtons.OK, MessageBoxIcon.Information);
    rutavacia = true;
}

if (rutavacia == false)
{
    //COMIENZO DEL BUCLE GRANDE FOR\\
    for (i = 0; i < nparts; i++)
    {
        if ((parts[i] == "Nombre"))
        {
            z = i + 1;
            medidaRealizada.Nombre = parts[z];
        }

        if (parts[i] == "FechaHora")
        {
            z = i + 1;
            medidaRealizada.FechaHora = Convert.ToString(parts[z] + '
' + parts[z + 1]);
        }

        if (parts[i] == "Longitud")
        {
            z = i + 1;
            medidaRealizada.Longitud = Convert.ToInt32(parts[z]);
        }
    }
}
```

```
if (parts[i] == "Lente")
{
    z = i + 1;
    medidaRealizada.Lente = Convert.ToInt32(parts[z]);
}

if (parts[i] == "Bckg")
{
    medidaRealizada.nuevainstanciaMatricesEnergias();
    z = i + 1;
    for (anillo = 0; anillo < 32; anillo++)
    {
        medidaRealizada.Bckg[anillo] =
double.Parse(parts[z]);
        z++;
    }
}
if (parts[i] == "MatrizEnergias")
{
    z = i + 1;

    for (muestras = 0; muestras < medidaRealizada.Longitud;
muestras++)
    {
        for (anillo = 0; anillo < 32; anillo++)
        {
            medidaRealizada.MatrizEnergias[muestras, anillo]
= double.Parse(parts[z]);
            z++;
        }
    }
}

if (parts[i] == "PromEnerg")
{
    //Instancia de la matriz promedio de energías
    inicializado en el anterior if
    z = i + 1;

    for (anillo = 0; anillo < 31; anillo++)
    {
        medidaRealizada.PromEnerg[anillo] =
double.Parse(parts[z]);
        z++;
    }
}

//Control de Cálculos
if (parts[i] == "CtrlCalcDiam")
{
    z = i + 1;
    medidaRealizada.CtrlCalcDiam =
Convert.ToBoolean(Convert.ToInt16(parts[z]));
}

//Matrices de probabilidad de clases de diámetros, resultados
estadísticos y otros parámetros de interés
if (medidaRealizada.CtrlCalcDiam == true)
{
    if (parts[i] == "AnillosDesech")
    {

```

```
        z = i + 1;
        medidaRealizada.AnillosDesech =
Convert.ToInt16(parts[z]);
    }
    if (parts[i] == "MatrizDiametros")
    {
        medidaRealizada.nuevainstanciaMatricesDiam();
        z = i + 1;

        for (muestras = 0; muestras <
medidaRealizada.Longitud; muestras++)
        {
            for (tiposDiam = 0; tiposDiam < 31; tiposDiam++)
            {
                medidaRealizada.MatrizDiametros[muestras,
tiposDiam] = double.Parse(parts[z]);
                z++;
            }
        }
    }
    if (parts[i] == "PromDiam")
    {
        //Instancia de la matriz promedio de energías
        inicializado en el anterior if
        z = i + 1;

        for (tiposDiam = 0; tiposDiam < 32; tiposDiam++)
        {
            medidaRealizada.PromDiam[tiposDiam] =
double.Parse(parts[z]);
            z++;
        }
    }
    if (parts[i] == "MatrizParam")
    {
        medidaRealizada.nuevainstanciaParam();
        z = i + 1;

        for (muestras = 0; muestras <
medidaRealizada.Longitud; muestras++)
        {
            for (tiposParam = 0; tiposParam < 9;
tiposParam++)
            {
                medidaRealizada.MatrizParam[muestras,
tiposParam] = double.Parse(parts[z]);
                z++;
            }
        }
    }
    if (parts[i] == "PromParam")
    {
        z = i + 1;
        for (tiposParam = 0; tiposParam < 9; tiposParam++)
        {
            medidaRealizada.PromParam[tiposParam] =
double.Parse(parts[z]);
            z++;
        }
    }
} //Fin del if (medidaRealizada.CtrlCalcDiam == true)
```

```
//Fin del FOR (i = 0; i < nparts; i++). Este for es el encargado  
de leer el fichero entero y de guardar todo lo leído en  
    *el campo correspondiente de la variable de tipo medida */  
  
/////GUARDADO DE LOS DATOS LEÍDOS DE LA MEDIDA CORRESPONDIENTE  
EN LA LISTAMEDIDAS\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
```

```
listaMedidas.Add(medidaRealizada);  
listaMedidasNombre.Add(medidaRealizada.Nombre);  
  
MessageBox.Show("Lectura finalizada");  
textBoxLeyendo.Hide();  
  
} //Fin del if (rutavacia == false)
```

1.4. Classes

Diagramas de las clases

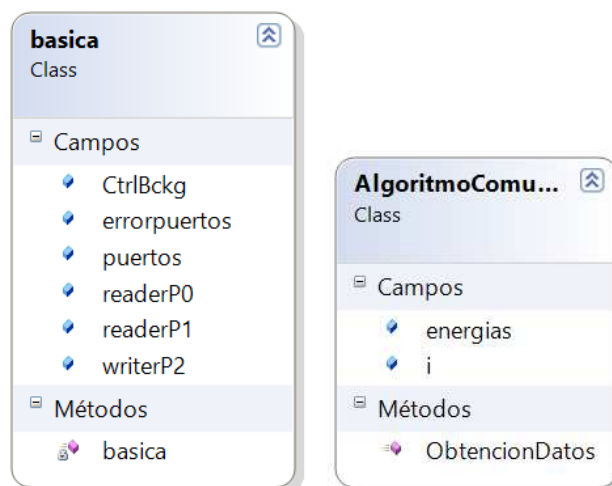


Figura 26. Diagramas de la clase "basica" y de la clase "AlgoritmoComunicacion"

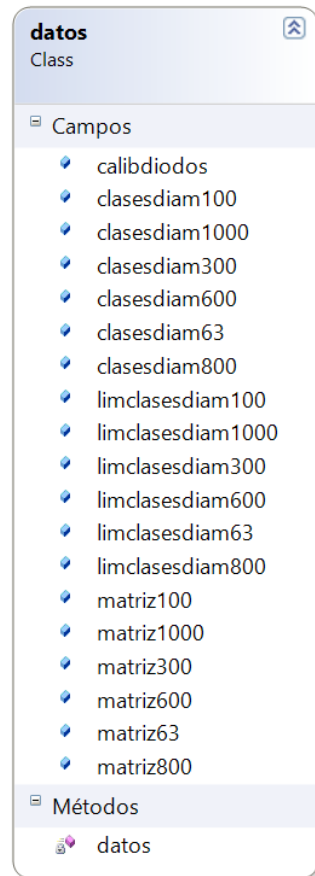


Figura 27. Diagrama de la clase "datos"

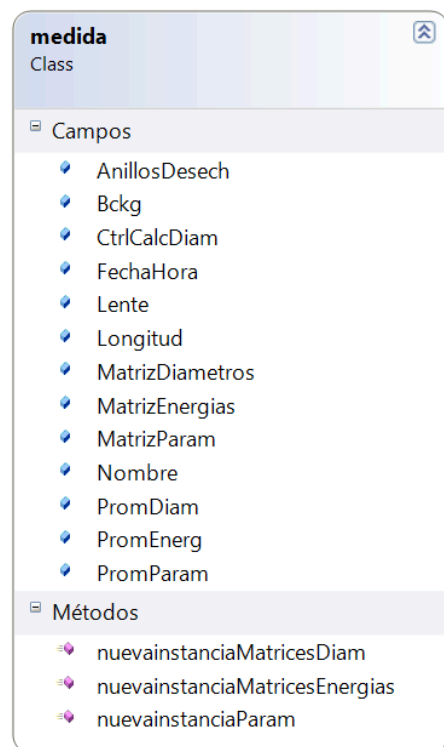


Figura 28. Diagrama de la clase "medida"

CLASE AlgoritmoComunicación

```
class AlgoritmoComunicacion
{
    public static int i;
    public static int[] energias;

    //Hay que definir una función (sin que sea constructor) ya que se ejecuta
    más de una vez
    public static void ObtencionDatos()
    {
        //A la hora de escribir o leer, únicamente hay que hacer referencia a:
        writerP2, readerP0 o readerP1

        //COMIENZO DEL ALGORITMO DE COMUNICACIÓN CON EL
        MALVERN\\

        //Creación de las variables modo disparo (md) desinhibición
        disparo(id) y disparo (de)
        byte md, id, de;
        int datdisp;

        //1.SELECCIÓN MODO DISPARO (modo externo)

        //Escribir un uno en la línea 5 del puerto2 (00100000)-->32=2^5
        md = 32;
        basica.writerP2.WriteSingleSamplePort(true, md);

        //2. DESINHIBICIÓN DEL DISPARO\\

        //Escribir un uno en la línea 6 del puerto2 (01100000)-->96=2^6+2^5

        id = 96;
        basica.writerP2.WriteSingleSamplePort(true, id);

        //3.DISPARO\\

        //Hay que escribir en la línea 7 del puerto 2 [P2] primero un 0 y
        después un 1 (11100000)-->224=2^7+2^6+2^5 (ya se ha escrito con anterioridad el
        //cero previo)

        de = 224;
        basica.writerP2.WriteSingleSamplePort(true, de);

        //4.INHIBICIÓN DEL DISPARO\\

        //Escribir un 0 cero en la línea 6 del puerto2 (10100000)--
        >160=2^7+2^5
        id = 160;
        basica.writerP2.WriteSingleSamplePort(true, id);

        //4. ¿DATOS DISPONIBLES?\\

        //Hay que leer en la línea 0 del puerto 1 un cero->es decir, ha de
        haber un flanco de bajada
        //Repetir la lectura de la línea hasta que datdisp sea par
```

```
datdisp = basica.readerP1.ReadSingleSamplePortByte();
while ((datdisp % 2) == 1)
{
    datdisp = basica.readerP1.ReadSingleSamplePortByte();
}

//5.COMIENZO DEL DIRECCIONAMIENTO Y LECTURA DE LAS ENERGÍAS DE LOS 32
ANILLOS\\

energias = new int[32];
byte direccionar;

for (i = 0; i < 32; i++)
{
    //COMIENZA EL DIRECCIONAMIENTO\\
    //Hay que escribir el número correspondiente al anillo que se
    quiere leer (del 0 al 31) en el P2
    //Para escribir siempre hay que hacerlo a traves de writerP2

    direccionar = Convert.ToByte(160 + i); //Se escribe (160+i) ya
    que hay que escribir un (101****)=2^7+2^5+i
    basica.writerP2.WriteSingleSamplePort(true, direccionar);

    //COMIENZA EL DELAY\\
    Application.DoEvents();
    Thread.Sleep(1); //En milisegundos

    //COMIENZA LA LECTURA\\
    //Lectura del dato de 10 bits( P0 y P1 ) del anillo i

    //Hay que leer a través del P1 para obtener dos de los bits
    que componen el dato completo
    //Máscara {11000000}--> convertido a entero: 2^7+2^6=192
    int datoP1;
    datoP1 = basica.readerP1.ReadSingleSamplePortByte();
    datoP1 = (datoP1 & 192) * 4;

    //Hay que leer a través del puerto 0 para leer los 8 bits
    restantes que componen el dato
    int datoP0;
    datoP0 = basica.readerP0.ReadSingleSamplePortByte();

    //GUARDADO del dato completo en energias[i]\\
    energias[i] = (datoP0 + datoP1);
}
}
```

CLASE basica

```
public class medida
{
    //Datos de interés
    public string Nombre;
    public string FechaHora; //En medir convertimos la variable de tipo
    DateTime en un string
    public int Longitud; //Número de muestras de una medida
    public int Lente;

    ///Background
    public double[] Bckg;

    ///Energías medidas
    public double[,] MatrizEnergias; //[long,anillos=32]
    public double[] PromEnerg; //[1,anillos=32]

    //Control del cálculo
    public bool CtrlCalcDiam;

    //Anillos desechados
    public int AnillosDesech;

    ///Probabilidad de tipos de diámetro
    //[long,frecuencia de los distintos tipos de diámetros] A partir de la
    MatrizEnergias
    public double[,] MatrizDiametros;

    //Promedio
    //[1,probabilidad de los distintos tipos de diámetros] A partir de la
    matriz de energías promedio. Se calcula con su correspondiente método de cálculo
    public double[] PromDiam;

    //Estadísticos y otros parámetros de interés.
    //Se calcula una matriz en la que cada elemento corresponde a un tipo de
    parámetro[i] de la muestra [j]
    public double[,] MatrizParam;
    public double[] PromParam;

    //Instancias
    public void nuevainstanciaMatricesEnergias()
    {
        Bckg = new double[32];
        MatrizEnergias = new double[Longitud, 32];
        PromEnerg = new double[32]; //(Numero de columnas=1, por eso se pone
    como un array unidimensional)
    }

    public void nuevainstanciaMatricesDiam()
    {
        MatrizDiametros = new double[Longitud, 31];
        PromDiam = new double[31];
    }

    public void nuevainstanciaParam()
    {
        MatrizParam = new double[Longitud, 9];
        PromParam = new double[9];
    }
}
```


CLASE datos

```
class datos
{
    public static double[] clasesdiam63, clasesdiam100, clasesdiam300,
    clasesdiam600, clasesdiam800, clasesdiam1000;
    public static double[] limclasesdiam63, limclasesdiam100,
    limclasesdiam300, limclasesdiam600, limclasesdiam800, limclasesdiam1000;
    public static double[,] matriz63, matriz100, matriz300, matriz600,
    matriz800, matriz1000;
    public static double[] calibdiodos;

    static datos()
    {
        //Vector de corrección: calibración de diodos
        calibdiodos = new double[32] {0.58, 0.612, 0.642, 0.674, 0.709,
        0.667, 0.732, 0.745, 0.782, 0.815, 0.835, 0.869, 0.895, 0.897, 0.938, 0.834,
        0.92, 0.944, 0.951, 0.952, 0.972, 0.994, 1, 0.984, 0.991, 0.977, 0.984, 0.982,
        0.993, 0.998, 0.998, 1 };

        //Vectores clases diámetros
        clasesdiam63 = new double[31] { 1.175567, 1.301541, 1.445089,
        1.608606, 1.794858, 2.007626, 2.250489, 2.527478, 2.843882, 3.204539, 3.615472,
        4.085330, 4.622097, 5.233093, 5.925790, 6.712041, 7.606853, 8.622027, 9.780100,
        11.092707, 12.592990, 14.331886, 16.362843, 18.787288, 21.736419, 25.452695,
        30.390170, 37.259462, 47.710106, 66.552432, 97.253232 };
        clasesdiam100 = new double[31] { 1.835244, 2.038217, 2.268854,
        2.530960, 2.828936, 3.168798, 3.556234, 3.997656, 4.501468, 5.075364, 5.728914,
        6.475864, 7.328895, 8.299637, 9.399959, 10.648687, 12.069652, 13.681596,
        15.520299, 17.604237, 19.986022, 22.746521, 25.970573, 29.819179, 34.500599,
        40.399689, 48.237179, 59.141032, 75.729564, 105.638225, 154.369829 };
        clasesdiam300 = new double[31] { 5.451065, 6.065469, 6.762407,
        7.553321, 8.451430, 9.474821, 10.640576, 11.967950, 13.482191, 15.206393,
        17.169290, 19.412153, 21.973045, 24.886867, 28.189243, 31.936673, 36.200675,
        41.037481, 46.554456, 52.807032, 59.953065, 68.235167, 77.907868, 89.454183,
        103.498895, 121.196585, 144.709454, 177.421394, 227.187354, 316.913699,
        463.108818 };
        clasesdiam600 = new double[31] { 10.891822, 12.121675, 13.516506,
        15.099204, 16.896210, 18.943710, 21.275870, 23.931203, 26.960212, 30.409090,
        34.335306, 38.821410, 43.943532, 49.771475, 56.376492, 63.871585, 72.399796,
        82.073593, 93.107704, 105.613000, 119.905191, 136.469511, 155.815015, 178.907736,
        206.997247, 242.392705, 289.418517, 354.842468, 454.374457, 633.827216,
        926.217510, };
        clasesdiam800 = new double[31] { 14.520425, 16.160431, 18.020392,
        20.130825, 22.526988, 25.257126, 28.366799, 31.907358, 35.946139, 40.544734,
        45.779772, 51.761317, 58.590879, 66.361527, 75.168269, 85.161771, 96.532760,
        109.431191, 124.143371, 140.817126, 159.873406, 181.959187, 207.753212,
        238.543526, 275.996223, 323.190183, 385.891280, 473.123229, 605.832561,
        845.102918, 1234.956656 };
        clasesdiam1000 = new double[31] { 18.149370, 20.199496, 22.524555,
        25.162693, 28.157986, 31.570741, 35.457905, 39.883669, 44.932204, 50.680502,
        57.224346, 64.701321, 73.238311, 82.951655, 93.960112, 106.452016, 120.665775,
        136.788834, 155.179077, 176.021288, 199.841652, 227.448891, 259.691434,
        298.179337, 344.995218, 403.987676, 482.364056, 591.404000, 757.290673,
        1056.378627, 1543.695806 };
    }
}
```

```
//Vectores límites clases diámetros
limclasesdiam63 = new double[32] { 1.1166591, 1.2344743, 1.3686078,
1.5215707, 1.6956414, 1.8940745, 2.1211783, 2.3797989, 2.6751575, 3.0126059,
3.3964724, 3.8344715, 4.3361893, 4.9080046, 5.5581808, 6.2933991, 7.1306833,
8.0830231, 9.1610319, 10.3991684, 11.7862458, 13.3997333, 15.2640388, 17.4616480,
20.1129271, 23.3599112, 27.5454780, 33.2348620, 41.2840617, 54.1361507,
78.9687141, 115.5377501 };
limclasesdiam100 = new double[32] { 1.7401743, 1.9303143, 2.1461202,
2.3915874, 2.6703335, 2.9875385, 3.3500582, 3.7624104, 4.2329007, 4.7700358,
5.3806930, 6.0771348, 6.8745929, 7.7831980, 8.8160766, 9.9838419, 11.3135316,
12.8257730, 14.5374186, 16.5031793, 18.7052941, 21.2667505, 24.2262915,
27.7148547, 31.9235041, 37.0776941, 43.7216834, 52.7526738, 65.5293909,
85.9297378, 125.3467127, 183.3929448 };
limclasesdiam300 = new double[32] { 5.1629944, 5.7391351, 6.3918026,
7.1330122, 7.9736303, 8.9292290, 10.0204122, 11.2607391, 12.6751603, 14.2892208,
16.1235655, 18.2150147, 20.6092916, 23.3367992, 26.4369348, 29.9415520,
33.9317937, 38.4695558, 43.6054069, 49.5035050, 56.1105595, 63.7955699,
72.6747649, 83.1409717, 95.7673937, 111.2303972, 131.1627732, 158.2561342,
196.5866535, 257.7880547, 376.0393439, 550.1782916 };
limclasesdiam600 = new double[32] { 10.3151364, 11.4685083,
12.7748408, 14.2581713, 15.9402358, 17.8521851, 20.0352348, 22.5165044,
25.3459019, 28.5745221, 32.2436575, 36.4269549, 41.2158658, 46.6711987,
52.8717512, 59.8812337, 67.8619370, 76.9376559, 87.2095295, 99.0058787,
112.2201210, 127.5902620, 145.3487592, 166.2812698, 191.5342027, 222.4602908,
262.3251194, 316.5119146, 393.1730222, 515.5758921, 752.0785389, 1100.3564813 };
limclasesdiam800 = new double[32] { 13.7514037, 15.2894453,
17.0314161, 19.0093676, 21.2522815, 23.8016935, 26.7125594, 30.0210387,
33.7936767, 38.0986006, 42.9908679, 48.5686753, 54.9539593, 62.2277982,
70.4952563, 79.8412812, 90.4822618, 102.5832581, 116.2791230, 132.0076183,
149.6266339, 170.1201786, 193.7981958, 221.7082288, 255.3788232, 296.6136232,
349.7667428, 422.0158173, 524.2306409, 687.4344806, 1002.7713563, 1467.1419553 };
limclasesdiam1000 = new double[32] { 17.1880329, 19.1107078,
21.2882837, 23.7608257, 26.5645613, 29.7514111, 33.3900703, 37.5257387,
42.2415987, 47.6228098, 53.7381941, 60.7104982, 68.6921434, 77.7844778,
88.1188321, 99.8013910, 113.1026415, 128.2289089, 145.3487592, 165.0093956,
187.0331801, 212.6501245, 242.2476581, 277.1352102, 319.2234633, 370.7669724,
437.2083805, 527.5197319, 655.2882691, 859.2930763, 1253.4641786, 1833.9274327 };

//Matrices de coeficientes
matriz63 = new double[31, 31]
{ { 2.30167E-05, 2.54763E-05, 2.82777E-05, 3.14678E-05, 3.5101E-05,
3.92467E-05, 4.39743E-05, 4.93638E-05, 5.56992E-05, 6.25203E-05, 7.04988E-05,
7.95997E-05, 9.1053E-05, 0.000101764, 0.000115106, 0.000130224, 0.000147316,
0.0001667, 0.00018859, 0.000213395, 0.000241317, 0.000273563, 0.000310619,
0.000353902, 0.000405782, 0.000468773, 0.000548075, 0.000651531, 0.000788182,
0.000970001, 0.001127382 }, { 3.32697E-05, 3.68241E-05, 4.08722E-05, 4.54814E-05,
5.07303E-05, 5.67185E-05, 6.35459E-05, 7.13271E-05, 8.04711E-05, 9.03117E-05,
0.000101816, 0.000114929, 0.000131416, 0.000146816, 0.000165972, 0.000187635,
0.000212067, 0.000239684, 0.000270744, 0.000305751, 0.000344883, 0.000389665,
0.0004405, 0.000498896, 0.000567263, 0.000647435, 0.000742963, 0.000856019,
0.000977403, 0.00105439, 0.0011762728 }, { 4.47783E-05, 4.95607E-05, 5.50066E-05,
6.12067E-05, 6.8266E-05, 7.63179E-05, 8.54955E-05, 9.59516E-05, 0.000108233,
0.000121442, 0.000136873, 0.000154445, 0.000176508, 0.000197083, 0.000222623,
0.000251429, 0.000283803, 0.000320233, 0.000360963, 0.00040652, 0.000456939,
0.000513875, 0.00057735, 0.000648462, 0.000728724, 0.000817663, 0.000913826,
0.001006791, 0.001056516, 0.000931834, 0.000279905 }, { 5.81585E-05, 6.43672E-05,
7.14364E-05, 7.94831E-05, 8.86427E-05, 9.90873E-05, 0.000110988, 0.00012454,
0.000140447, 0.000157544, 0.000177496, 0.000200187, 0.000228627, 0.000255091,
0.000287854, 0.000324674, 0.000365862, 0.000411929, 0.000463027, 0.000519588,
0.000581328, 0.000649762, 0.000724105, 0.000804349, 0.000889894, 0.000975966,
0.00105245, 0.001090399, 0.001015445, 0.000658252, 2.07384E-05 }, { 7.60224E-05,
8.41336E-05, 9.33672E-05, 0.000103875, 0.000115833, 0.000129464, 0.000144986,
```

0.000162653, 0.000183373, 0.00020562, 0.000231549, 0.000260988, 0.000297801,
0.000331959, 0.000374101, 0.000421236, 0.000473638, 0.000531777, 0.00059558,
0.000665209, 0.000739777, 0.000820273, 0.000904452, 0.000990212, 0.001073173,
0.001141774, 0.001173383, 0.001117014, 0.000875565, 0.000340388, 4.70495E-05 }, {
9.76552E-05, 0.000108067, 0.000119917, 0.000133399, 0.000148736, 0.000166209,
0.000186096, 0.000208712, 0.000235211, 0.000263626, 0.000296691, 0.000334153,
0.000380861, 0.000424047, 0.000477088, 0.000536055, 0.000601093, 0.000672494,
0.000749759, 0.000832489, 0.00091879, 0.001008502, 0.001097076, 0.001179065,
0.001244428, 0.001272747, 0.001227266, 0.001040594, 0.000632683, 8.91725E-05,
0.000133312 }, { 0.000124092, 0.000137311, 0.000152351, 0.000169457, 0.000188906,
0.000211053, 0.00023624, 0.000264856, 0.000298344, 0.000334195, 0.000375829,
0.000422874, 0.000481316, 0.000535105, 0.000600795, 0.000673259, 0.000752365,
0.000838021, 0.000928989, 0.001023886, 0.001119257, 0.00121293, 0.001297029,
0.001361371, 0.00138855, 0.001348628, 0.001193819, 0.000859624, 0.000347379,
3.75878E-06, 8.19885E-05 }, { 0.000160096, 0.000177132, 0.000196509, 0.000218535,
0.000243566, 0.000272046, 0.000304406, 0.000341127, 0.000384032, 0.000429873,
0.00048297, 0.000542767, 0.000616703, 0.000684355, 0.000766372, 0.000855934,
0.000952384, 0.001054903, 0.001161002, 0.001267634, 0.001368914, 0.001459414,
0.001526548, 0.001553813, 0.001515144, 0.001372909, 0.001078703, 0.000606817,
0.000109927, 7.90964E-05, 4.48694E-06 }, { 0.000208384, 0.000230528, 0.000255704,
0.000284306, 0.000316785, 0.000353707, 0.00039561, 0.000443086, 0.000498449,
0.00055745, 0.000625569, 0.000701956, 0.000795842, 0.000881109, 0.000983504,
0.001093843, 0.001210536, 0.001331469, 0.001452125, 0.001566792, 0.001666,
0.00173945, 0.001768537, 0.001729554, 0.001588881, 0.001309423, 0.000865689,
0.000322469, 5.28381E-06, 0.000183593, 5.22178E-05 }, { 0.00027041, 0.000299096,
0.00033169, 0.000368693, 0.000410676, 0.000458344, 0.000512363, 0.00057345,
0.000644504, 0.000719984, 0.000806767, 0.000903548, 0.001021583, 0.001127739,
0.001253626, 0.001386878, 0.001524326, 0.001661702, 0.001791368, 0.001903632,
0.00198418, 0.002016198, 0.001974814, 0.001830766, 0.001549725, 0.001114405,
0.000560566, 8.64893E-05, 7.0064E-05, 0.000157311, 5.88419E-05 }, { 0.000352237,
0.000389522, 0.000431856, 0.000479873, 0.000534288, 0.00059598, 0.000665757,
0.00074447, 0.000835728, 0.000932267, 0.001042666, 0.001164899, 0.001312463,
0.001443453, 0.001596161, 0.001753837, 0.001910723, 0.002059095, 0.002186684,
0.002278164, 0.002313247, 0.002267549, 0.002112183, 0.001819017, 0.00136924,
0.00079735, 0.000241044, 6.92268E-06, 0.000218468, 3.17749E-05, 6.03306E-06 }, {
0.000455436, 0.000503511, 0.000558048, 0.000619834, 0.000689749, 0.000768866,
0.000858134, 0.000958518, 0.001074418, 0.001196369, 0.00133486, 0.001486764,
0.001667696, 0.001825513, 0.00200523, 0.002184345, 0.002353153, 0.00249884,
0.002602918, 0.00264312, 0.002593452, 0.002423573, 0.002108508, 0.001636797,
0.001032884, 0.000418099, 3.26203E-05, 0.000120344, 0.000254207, 3.06468E-05,
5.23137E-05 }, { 0.000605314, 0.000668983, 0.000741123, 0.000822733, 0.000914901,
0.001018945, 0.001135968, 0.001267022, 0.001417508, 0.001574733, 0.001751629,
0.001943211, 0.002167224, 0.002357848, 0.002567628, 0.002765618, 0.002935842,
0.003057801, 0.003104621, 0.003046049, 0.002852847, 0.002494786, 0.001967733,
0.001304459, 0.000609592, 0.000107255, 4.59705E-05, 0.000309375, 0.000108235,
0.000121713, 8.77628E-06 }, { 0.00078588, 0.000868164, 0.000961256, 0.001066363,
0.001184778, 0.001318025, 0.001467278, 0.001633533, 0.001823071, 0.002019243,
0.002237221, 0.002469249, 0.00273363, 0.002950685, 0.003177359, 0.003372555,
0.003511943, 0.003565692, 0.003498976, 0.003277266, 0.002878496, 0.002290058,
0.001558496, 0.000794347, 0.000196875, 1.5099E-05, 0.000260175, 0.000318745,
1.16451E-05, 4.12532E-05, 3.96051E-05 }, { 0.001023053, 0.001129539, 0.001249776,
0.001385193, 0.001537266, 0.00170768, 0.001897535, 0.002107522, 0.002344636,
0.002586959, 0.002851652, 0.003126651, 0.003428427, 0.003662853, 0.003886889,
0.004047047, 0.004108503, 0.004031621, 0.003777613, 0.003318115, 0.002658891,
0.001834669, 0.000978142, 0.000292291, 1.22787E-05, 0.000206406, 0.000432134,
0.000107143, 0.000136491, 4.30317E-05, 1.40561E-05 }, { 0.001329896, 0.001467274,
0.001621998, 0.001795693, 0.00198994, 0.002206436, 0.002445923, 0.002708324,
0.003000838, 0.003294648, 0.003607998, 0.003922305, 0.004247877, 0.00447817,
0.004661992, 0.004732675, 0.004644792, 0.004353072, 0.003827375, 0.003068374,
0.002143413, 0.001171407, 0.000385388, 2.27249E-05, 0.000169551, 0.00047916,
0.000301558, 2.5702E-05, 0.000124075, 6.53755E-05, 1.54589E-05 }, { 0.001707002,
0.001881613, 0.002077626, 0.002296744, 0.002540449, 0.002810132, 0.003105645,

0.003425353, 0.003775528, 0.004118814, 0.004472422, 0.004808466, 0.005124121,
0.005308279, 0.005388349, 0.005288408, 0.004958194, 0.004360949, 0.003501901,
0.00244815, 0.001365848, 0.000472682, 3.66511E-05, 0.000146447, 0.000503327,
0.000464091, 4.05974E-05, 0.00019592, 1.81691E-05, 3.03868E-05, 2.0707E-05 }, {
0.002207538, 0.002430503, 0.002679725, 0.002956784, 0.003262716, 0.003598051,
0.003960855, 0.00434661, 0.004758819, 0.005148896, 0.005529808, 0.005860272,
0.006114389, 0.006189893, 0.006075673, 0.005697389, 0.005015552, 0.004030828,
0.002826696, 0.001580721, 0.000570433, 5.23968E-05, 0.000139489, 0.000534263,
0.000598514, 0.000143488, 0.000111727, 0.00013096, 0.000119207, 3.71123E-05,
1.77875E-05 }, { 0.002827611, 0.003108537, 0.003420783, 0.003765375, 0.004142233,
0.004550042, 0.00498361, 0.005433499, 0.005897241, 0.00631282, 0.006683514,
0.006950666, 0.007052763, 0.00692409, 0.006494855, 0.005719675, 0.004603708,
0.003233017, 0.001817919, 0.000660009, 6.83733E-05, 0.000140479, 0.000575586,
0.000715509, 0.000264463, 4.65239E-05, 0.000266639, 3.54603E-05, 2.17053E-05,
4.6685E-05, 1.41437E-05 }, { 0.003600524, 0.003950679, 0.004337001, 0.004759224,
0.005215054, 0.005699741, 0.00620259, 0.00670622, 0.007197424, 0.007598857,
0.00789674, 0.008012097, 0.007835313, 0.007381854, 0.006507077, 0.005243905,
0.003695362, 0.002086755, 0.00076905, 8.2778E-05, 0.000144398, 0.000627069,
0.000828464, 0.000386777, 2.63651E-05, 0.000296054, 5.92199E-05, 0.00014797,
8.62584E-05, 2.40222E-05, 1.2672E-05 }, { 0.004517199, 0.004944485, 0.005411318,
0.005914945, 0.006449173, 0.007003479, 0.007558549, 0.008085138, 0.008552974,
0.008869897, 0.008996721, 0.008836246, 0.00822567, 0.007326276, 0.005918435,
0.004184678, 0.002382669, 0.000890695, 0.000101983, 0.000151679, 0.000678544,
0.0009391, 0.000499178, 3.34164E-05, 0.000277743, 0.000181953, 0.000121314,
2.64053E-05, 3.08581E-05, 2.10765E-05, 1.29013E-05 }, { 0.005635278, 0.006149416,
0.00670385, 0.007291515, 0.007899794, 0.008508998, 0.009086888, 0.009587315,
0.009955167, 0.01008715, 0.009911397, 0.009323412, 0.008117897, 0.006684319,
0.0047524, 0.002729574, 0.001044229, 0.000128118, 0.000153396, 0.00073936,
0.001058307, 0.000614699, 5.47578E-05, 0.000249443, 0.000301136, 5.01903E-05,
0.000115701, 0.000103458, 3.68481E-05, 2.20738E-05, 7.51787E-06 }, { 0.007017685,
0.007628319, 0.008275237, 0.008944279, 0.009612707, 0.010246897, 0.01079608,
0.011191369, 0.011344507, 0.01115171, 0.010506327, 0.009324189, 0.007397422,
0.005432437, 0.003158742, 0.001238364, 0.000167068, 0.000151875, 0.000798218,
0.001192278, 0.000746155, 8.62909E-05, 0.000227201, 0.000400399, 3.45266E-05,
0.000213528, 7.40257E-05, 4.46932E-05, 4.71441E-05, 2.16102E-05, 9.90753E-06 }, {
0.008593685, 0.009295873, 0.010021638, 0.010746019, 0.011431545, 0.01202523,
0.012452413, 0.012617739, 0.012395788, 0.011707363, 0.010423766, 0.008535439,
0.00591645, 0.003631188, 0.0014667, 0.000217241, 0.000142445, 0.000845375,
0.00132617, 0.000878018, 0.000127603, 0.000205702, 0.000477253, 6.39465E-05,
0.000232121, 3.57655E-05, 7.35959E-05, 4.14021E-05, 3.73874E-05, 1.98342E-05,
7.96098E-06 }, { 0.010421466, 0.011204121, 0.011984821, 0.012722952, 0.013360688,
0.013819824, 0.013997637, 0.013772807, 0.012987512, 0.011630235, 0.009584971,
0.006996593, 0.003936364, 0.0017554, 0.000291564, 0.000126357, 0.000876375,
0.001465763, 0.001041192, 0.000179138, 0.000180114, 0.000544168, 0.000119655,
0.00020665, 8.55774E-05, 0.000153739, 9.12513E-05, 5.80925E-05, 2.93633E-05,
1.37785E-05, 6.50896E-06 }, { 0.012529009, 0.013367034, 0.014158895, 0.014842411,
0.015333405, 0.015523755, 0.015282888, 0.014473651, 0.012927932, 0.010768779,
0.007955414, 0.004878104, 0.0019042, 0.000399306, 0.000104228, 0.000891415,
0.001609221, 0.001236208, 0.000259013, 0.00015565, 0.000603387, 0.00019642,
0.000168589, 0.000176826, 0.000146053, 4.39555E-05, 3.80011E-05, 4.67544E-05,
2.40781E-05, 1.448E-05, 5.32807E-06 }, { 0.014747601, 0.015584306, 0.016306189,
0.016824316, 0.017024791, 0.016770867, 0.015915825, 0.014335509, 0.011902527,
0.008958984, 0.005620209, 0.002567291, 0.00043422, 7.82296E-05, 0.000870846,
0.00173219, 0.001455433, 0.000370939, 0.000119772, 0.000646811, 0.000295504,
0.000125348, 0.000267322, 9.61534E-05, 6.39882E-05, 8.47351E-05, 4.87122E-05,
3.6077E-05, 1.99609E-05, 1.16521E-05, 5.70118E-06 }, { 0.017119609, 0.017877298,
0.018421141, 0.018631637, 0.018366064, 0.017469275, 0.015805794, 0.013315889,
0.009993257, 0.00648019, 0.00310689, 0.000764299, 7.83386E-05, 0.000817268,
0.001834655, 0.001704449, 0.000534498, 8.27779E-05, 0.000667541, 0.000413394,
8.33919E-05, 0.000346145, 5.71335E-05, 0.000143898, 0.000130849, 8.31327E-05,
5.41166E-05, 3.03668E-05, 2.13825E-05, 1.25278E-05, 4.02045E-06 }, { 0.019555451,
0.020122141, 0.020341667, 0.020064874, 0.019130537, 0.017391361, 0.01477363,

```
0.011359083, 0.007363339, 0.003757966, 0.001053355, 5.39987E-05, 0.000845022,
0.001902454, 0.001978416, 0.000758975, 5.64962E-05, 0.000654355, 0.000564307,
5.68806E-05, 0.000401429, 5.14906E-05, 0.000218033, 0.000102427, 4.89132E-05,
4.58078E-05, 4.53526E-05, 2.52415E-05, 1.91408E-05, 9.74212E-06, 3.86432E-06 }, {
0.021813557, 0.022039543, 0.021754653, 0.020790959, 0.018995142, 0.016276499,
0.012698129, 0.008572655, 0.004418203, 0.001430149, 8.52759E-05, 0.000597754,
0.002007726, 0.002250738, 0.001055985, 6.23002E-05, 0.000590311, 0.000731587,
5.88353E-05, 0.00041738, 9.53352E-05, 0.000250832, 5.81132E-05, 6.5133E-05,
6.37082E-05, 4.09461E-05, 3.65862E-05, 2.12787E-05, 1.32774E-05, 8.30522E-06,
3.6261E-06 }, { 0.02377795, 0.023487713, 0.022503231, 0.020662422, 0.017865696,
0.014149558, 0.009801138, 0.005430393, 0.0018482, 0.000172234, 0.000446607,
0.001837999, 0.00247135, 0.001431214, 0.000132138, 0.000474991, 0.000895317,
0.000120812, 0.000377156, 0.000193214, 0.000224534, 5.80423E-05, 0.000137758,
0.000114153, 7.72023E-05, 4.2862E-05, 3.01054E-05, 2.37885E-05, 1.38154E-05,
6.88296E-06, 2.91353E-06 } };
```

```
matriz100 = new double[31, 31]
{ { 1.42606E-05, 1.58337E-05, 1.76203E-05, 1.96501E-05, 2.19573E-05,
2.45858E-05, 2.75793E-05, 3.09884E-05, 3.49925E-05, 3.93007E-05, 4.43371E-05,
5.00797E-05, 5.7304E-05, 6.40582E-05, 7.24701E-05, 8.20001E-05, 9.27735E-05,
0.00010499, 0.000118785, 0.000134415, 0.000152009, 0.000172328, 0.000195675,
0.000222946, 0.000255633, 0.00029532, 0.000345282, 0.000410462, 0.000496555,
0.000611104, 0.000710256 }, { 2.06136E-05, 2.28869E-05, 2.54686E-05, 2.84014E-05,
3.17347E-05, 3.55315E-05, 3.98547E-05, 4.47769E-05, 5.05561E-05, 5.67715E-05,
6.40337E-05, 7.23084E-05, 8.2708E-05, 9.24195E-05, 0.000104497, 0.000118153,
0.000133553, 0.000150959, 0.000170533, 0.000192593, 0.000217251, 0.000245469,
0.000277499, 0.000314293, 0.000357368, 0.000407881, 0.000468068, 0.000539298,
0.000615774, 0.000664278, 0.000740524 }, { 2.77448E-05, 3.08037E-05, 3.42771E-05,
3.82222E-05, 4.27053E-05, 4.78107E-05, 5.36224E-05, 6.02368E-05, 6.79993E-05,
7.63427E-05, 8.60841E-05, 9.71725E-05, 0.00011109, 0.000124065, 0.000140169,
0.000158328, 0.000178735, 0.000201696, 0.000227364, 0.000256073, 0.000287845,
0.000323722, 0.000363719, 0.000408525, 0.000459098, 0.000515136, 0.000575725,
0.000634298, 0.000665628, 0.000707072, 0.00076337 }, { 3.60364E-05, 4.00078E-05,
4.45166E-05, 4.9637E-05, 5.54542E-05, 6.20771E-05, 6.96134E-05, 7.81865E-05,
8.82414E-05, 9.90405E-05, 0.000111636, 0.000125956, 0.000143897, 0.000160587,
0.000181246, 0.000204458, 0.000230421, 0.000259457, 0.000291662, 0.000327307,
0.000366214, 0.000409339, 0.000456185, 0.000506749, 0.000560651, 0.000614885,
0.000663077, 0.000686988, 0.000639762, 0.000414709, 1.30616E-05 }, { 4.71073E-05,
5.22958E-05, 5.81856E-05, 6.48725E-05, 7.24675E-05, 8.11108E-05, 9.09417E-05,
0.000102118, 0.000115216, 0.000129269, 0.000145639, 0.000164218, 0.000187442,
0.000208986, 0.00023556, 0.000265277, 0.000298311, 0.000334958, 0.000375172,
0.000419056, 0.00046605, 0.000516778, 0.000569825, 0.000623867, 0.000676144,
0.000719373, 0.00073929, 0.000703771, 0.000551636, 0.000214438, 2.96523E-05 }, {
6.05153E-05, 6.71761E-05, 7.47354E-05, 8.33153E-05, 9.30567E-05, 0.000104138,
0.000116733, 0.000131042, 0.000147794, 0.000165745, 0.000186622, 0.000210266,
0.000239735, 0.000266974, 0.000300423, 0.000337603, 0.000378605, 0.000423616,
0.000472318, 0.000524462, 0.000578854, 0.000635393, 0.000691214, 0.000742883,
0.000784074, 0.00080192, 0.00077326, 0.00065563, 0.000398601, 5.61627E-05,
8.39976E-05 }, { 7.69024E-05, 8.53597E-05, 9.49552E-05, 0.000105842, 0.000118197,
0.000132243, 0.000148197, 0.000166304, 0.000187476, 0.000210126, 0.000236416,
0.000266111, 0.000302986, 0.000336917, 0.000378346, 0.00042404, 0.000473916,
0.000527917, 0.000585261, 0.000645079, 0.000705193, 0.000764233, 0.000817238,
0.00085779, 0.000874918, 0.000849759, 0.0007522, 0.000541602, 0.00021883,
2.37012E-06, 5.16373E-05 }, { 9.92234E-05, 0.000110124, 0.000122487, 0.000136508,
0.00015241, 0.000170475, 0.000190975, 0.000214213, 0.000241341, 0.000270306,
0.000303839, 0.000341586, 0.000388243, 0.000430923, 0.000482655, 0.000539137,
0.000599955, 0.000664594, 0.000731484, 0.000798706, 0.000862551, 0.000919597,
0.000961914, 0.000979099, 0.000954726, 0.000865081, 0.000679664, 0.000382292,
6.92185E-05, 4.98716E-05, 2.82356E-06 }, { 0.000129164, 0.000143335, 0.0001594,
0.000177609, 0.000198246, 0.000221669, 0.000248218, 0.000278266, 0.000313276,
0.000350562, 0.000393588, 0.000441815, 0.000501069, 0.000554869, 0.000619463,
0.000689058, 0.00076265, 0.00083891, 0.000914987, 0.000987284, 0.00104983,
```


0.001096135, 0.001114471, 0.001089898, 0.001001225, 0.000825081, 0.000545416,
0.000203102, 3.32371E-06, 0.000115706, 3.29316E-05 }, { 0.000167632, 0.000185992,
0.000206795, 0.000230357, 0.000257037, 0.000287283, 0.000321514, 0.000360184,
0.000405124, 0.000452832, 0.000507656, 0.000568769, 0.000643277, 0.000710269,
0.000789697, 0.000873758, 0.000960454, 0.001047096, 0.001128868, 0.001199661,
0.00125045, 0.001270638, 0.001244546, 0.001153731, 0.000976562, 0.000702159,
0.000353103, 5.44208E-05, 4.42187E-05, 9.90721E-05, 3.7042E-05 }, { 0.000218394,
0.000242263, 0.000269288, 0.00029987, 0.000334458, 0.000373611, 0.000417838,
0.000467676, 0.000525409, 0.000586441, 0.000656199, 0.000733401, 0.000826569,
0.000909252, 0.001005623, 0.001105113, 0.00120409, 0.001297687, 0.001378165,
0.001435863, 0.001457988, 0.001429169, 0.001331198, 0.001146349, 0.000862783,
0.000502291, 0.000151733, 4.37961E-06, 0.000137759, 1.99572E-05, 3.81348E-06 }, {
0.000282436, 0.000313223, 0.000348048, 0.00038741, 0.000431864, 0.000482089,
0.000538685, 0.000602263, 0.000675605, 0.000752724, 0.000840257, 0.000936226,
0.001050494, 0.001150136, 0.001263583, 0.00137663, 0.001483156, 0.001575082,
0.001640747, 0.001666108, 0.001634776, 0.001527626, 0.001328919, 0.001031459,
0.000650705, 0.000263222, 2.04707E-05, 7.60055E-05, 0.000160145, 1.94009E-05,
3.29752E-05 }, { 0.000375482, 0.000416268, 0.000462351, 0.00051436, 0.000572985,
0.000639059, 0.000713279, 0.000796311, 0.000891573, 0.000991035, 0.001102881,
0.00122396, 0.001365484, 0.001485882, 0.001618351, 0.001743352, 0.001850808,
0.001927784, 0.001957332, 0.001920371, 0.001798467, 0.001572573, 0.001240121,
0.000821838, 0.000383792, 6.73694E-05, 2.91287E-05, 0.000195151, 6.80048E-05,
7.67321E-05, 5.51383E-06 }, { 0.000487652, 0.000540389, 0.000599882, 0.000666897,
0.000742251, 0.000826909, 0.000921614, 0.001026997, 0.001147035, 0.001271191,
0.001409074, 0.00155578, 0.001722882, 0.001860033, 0.00200323, 0.002126519,
0.002214543, 0.00224848, 0.002206359, 0.002066407, 0.001814731, 0.001443426,
0.000981938, 0.0005001, 0.000123679, 9.60657E-06, 0.000164369, 0.000200729,
7.39924E-06, 2.58333E-05, 2.49466E-05 }, { 0.000635095, 0.000703383, 0.000780271,
0.000866662, 0.000963492, 0.001071827, 0.001192366, 0.001325547, 0.001475803,
0.001629256, 0.001796785, 0.001970759, 0.002161603, 0.00230981, 0.002451416,
0.002552625, 0.002591452, 0.002542884, 0.002382455, 0.002092303, 0.001676143,
0.001156003, 0.000615759, 0.000183583, 7.72656E-06, 0.000130674, 0.000272521,
6.71276E-05, 8.63089E-05, 2.73412E-05, 8.93883E-06 }, { 0.000826032, 0.000914197,
0.001013213, 0.001124108, 0.001247884, 0.001385618, 0.001537782, 0.001704328,
0.001889828, 0.002076022, 0.002274489, 0.002473472, 0.002679507, 0.0028252,
0.002941463, 0.002986156, 0.002930595, 0.002746216, 0.002414047, 0.001934622,
0.001350623, 0.00073732, 0.000241943, 1.41624E-05, 0.00010765, 0.000302605,
0.000189482, 1.64605E-05, 7.78378E-05, 4.09655E-05, 9.67199E-06 }, { 0.001061005,
0.001173174, 0.001298734, 0.001438764, 0.001594204, 0.001765939, 0.001953879,
0.00215699, 0.002379264, 0.002597013, 0.002821182, 0.003034116, 0.003234048,
0.003350649, 0.00340133, 0.003338083, 0.003129185, 0.002751492, 0.002208469,
0.001542744, 0.00085954, 0.000296531, 2.27506E-05, 9.33349E-05, 0.000318384,
0.000292038, 2.51735E-05, 0.000123987, 1.16353E-05, 1.94504E-05, 1.29765E-05 }, {
0.001373344, 0.001516752, 0.001676592, 0.001853867, 0.002049236, 0.002263039,
0.002494051, 0.002739416, 0.003001374, 0.003249091, 0.003490844, 0.003700473,
0.003861597, 0.003909447, 0.003837073, 0.003597485, 0.003165838, 0.002542782,
0.001781448, 0.000994478, 0.000357507, 3.24117E-05, 8.93177E-05, 0.000338621,
0.000377058, 8.92725E-05, 7.15122E-05, 8.17189E-05, 7.52011E-05, 2.30602E-05,
1.11845E-05 }, { 0.002106594, 0.002323425, 0.002563882, 0.002828821, 0.003118292,
0.003431454, 0.003764573, 0.004110764, 0.004468707, 0.004791236, 0.005081918,
0.005296632, 0.005390769, 0.005309608, 0.005004238, 0.004436492, 0.003606187,
0.002572546, 0.001488181, 0.000577797, 8.12435E-05, 9.1073E-05, 0.000409768,
0.000549255, 0.000236075, 3.32014E-05, 0.000200496, 2.72076E-05, 2.64958E-05,
2.99174E-05, 9.06741E-06 }, { 0.002245718, 0.002471736, 0.002720389, 0.002991503,
0.00328362, 0.003593724, 0.003915014, 0.004236444, 0.004549645, 0.0048054,
0.004995048, 0.005068436, 0.004956012, 0.004667844, 0.004112332, 0.003310887,
0.002329524, 0.001311811, 0.000480525, 5.06872E-05, 9.36882E-05, 0.000399539,
0.000523106, 0.000241047, 1.7025E-05, 0.000188091, 3.61877E-05, 9.29676E-05,
5.42574E-05, 1.54661E-05, 8.07505E-06 }, { 0.002822559, 0.003099058, 0.003400281,
0.003724466, 0.004067666, 0.004423173, 0.004778676, 0.005115543, 0.005414521,
0.005616871, 0.005697771, 0.005595446, 0.005206506, 0.004634121, 0.003739126,
0.002638559, 0.001497096, 0.000555413, 6.20061E-05, 9.93236E-05, 0.00043396,

```
0.000593776, 0.000310892, 2.05598E-05, 0.000178233, 0.000112228, 7.85728E-05,
1.70187E-05, 2.01793E-05, 1.33383E-05, 8.09589E-06 }, { 0.003529136, 0.003862883,
0.004221748, 0.004601199, 0.004993164, 0.005385054, 0.005756262, 0.006077308,
0.00631303, 0.006397504, 0.006285068, 0.005909356, 0.005140027, 0.004226446,
0.002997482, 0.00171412, 0.000649647, 7.72559E-05, 0.000101733, 0.000475077,
0.000670357, 0.000382415, 3.27629E-05, 0.000162207, 0.000187139, 3.37592E-05,
7.06647E-05, 6.44791E-05, 2.25282E-05, 1.39108E-05, 4.7423E-06 }, { 0.004647694,
0.005067925, 0.005512062, 0.0059706, 0.006428273, 0.006862498, 0.007239135,
0.007511847, 0.007621425, 0.007497478, 0.007068686, 0.006278101, 0.00498591,
0.003666065, 0.002136995, 0.000843305, 0.000118411, 0.000102509, 0.000532813,
0.000798051, 0.000502995, 6.21051E-05, 0.000151591, 0.00026724, 2.62613E-05,
0.000141036, 4.95979E-05, 3.0657E-05, 3.03247E-05, 1.53061E-05, 6.48456E-06 }, {
0.005415639, 0.005875472, 0.006349324, 0.006821068, 0.007266529, 0.00765157,
0.007928138, 0.008034974, 0.007891658, 0.007447954, 0.006622174, 0.005410038,
0.003734025, 0.002277672, 0.000907316, 0.000128488, 9.84456E-05, 0.000551314,
0.000844685, 0.000544618, 7.41027E-05, 0.000139222, 0.000300429, 3.71824E-05,
0.000149892, 2.28379E-05, 4.3592E-05, 2.49481E-05, 2.37236E-05, 1.23272E-05,
5.11657E-06 }, { 0.006595224, 0.007110813, 0.007623529, 0.008106992, 0.008523713,
0.008823081, 0.008938751, 0.008792627, 0.00828345, 0.007405431, 0.006085939,
0.004421972, 0.002465224, 0.00108244, 0.000170656, 9.03013E-05, 0.000578228,
0.000938005, 0.000645204, 0.0001026, 0.000125937, 0.000345381, 6.81534E-05,
0.000138485, 4.8968E-05, 9.63634E-05, 5.94376E-05, 3.55501E-05, 1.8635E-05,
8.30954E-06, 4.10199E-06 }, { 0.007969455, 0.00852554, 0.009049289, 0.009500076,
0.009823032, 0.009947878, 0.00979007, 0.009261415, 0.008254575, 0.006852892,
0.005034343, 0.003057347, 0.00116687, 0.000231335, 7.79785E-05, 0.000597591,
0.001036862, 0.000766013, 0.00014666, 0.000113733, 0.000387549, 0.000111241,
0.000118805, 0.000102377, 9.85792E-05, 3.11917E-05, 2.5944E-05, 2.9858E-05,
1.50785E-05, 1.00156E-05, 3.50179E-06 }, { 0.009437749, 0.009997826, 0.010479357,
0.010823852, 0.010956678, 0.010788662, 0.01022491, 0.009186621, 0.007594826,
0.005679159, 0.003522128, 0.001572813, 0.000245902, 6.18924E-05, 0.000596968,
0.001126984, 0.000903383, 0.000208053, 9.34657E-05, 0.000422343, 0.000167238,
9.49229E-05, 0.000158221, 7.17682E-05, 3.55543E-05, 4.82187E-05, 2.85235E-05,
2.32931E-05, 1.31657E-05, 6.62875E-06, 3.58475E-06 }, { 0.011033903, 0.011546371,
0.011912757, 0.01205396, 0.011876094, 0.011278084, 0.010173562, 0.008528581,
0.006347683, 0.004062187, 0.001897473, 0.000434701, 6.58164E-05, 0.000578238,
0.00121042, 0.001062359, 0.000298191, 7.01878E-05, 0.000447604, 0.00023448,
6.93399E-05, 0.000211051, 4.60677E-05, 7.92079E-05, 8.3572E-05, 5.48083E-05,
3.21901E-05, 1.90746E-05, 1.48044E-05, 7.62906E-06, 2.86984E-06 }, { 0.012707391,
0.013095329, 0.013244829, 0.013056688, 0.012424958, 0.01125515, 0.009505036,
0.007239619, 0.004616873, 0.002288546, 0.000594241, 3.68745E-05, 0.000617332,
0.001280231, 0.001242876, 0.000422773, 4.84196E-05, 0.000457094, 0.000323797,
4.90816E-05, 0.000255899, 3.39628E-05, 0.000128443, 7.81191E-05, 3.83523E-05,
3.3981E-05, 2.8064E-05, 1.56696E-05, 1.00483E-05, 6.38266E-06, 2.49053E-06 }, {
0.014305335, 0.014462072, 0.014264931, 0.013602316, 0.012375252, 0.010531059,
0.008125703, 0.005386772, 0.002680797, 0.000800777, 4.18847E-05, 0.000474032,
0.001377995, 0.001432418, 0.000590384, 3.77882E-05, 0.000439986, 0.000429137,
3.71008E-05, 0.000283908, 4.42365E-05, 0.000163406, 5.21193E-05, 3.41985E-05,
3.38244E-05, 2.7063E-05, 2.31023E-05, 1.55632E-05, 1.08914E-05, 5.21366E-06,
2.06876E-06 }, { 0.015748957, 0.015543744, 0.014852984, 0.013571018, 0.01164008,
0.009101834, 0.00617448, 0.003294799, 0.001024309, 7.08962E-05, 0.000393006,
0.001324127, 0.001588492, 0.000808171, 5.31409E-05, 0.000392849, 0.000545498,
4.95501E-05, 0.000286503, 8.32617E-05, 0.000171547, 3.57667E-05, 6.74469E-05,
6.62556E-05, 4.36821E-05, 2.65266E-05, 2.01535E-05, 1.63526E-05, 7.30241E-06,
5.01638E-06, 2.11388E-06 } };
```

```
matriz300 = new double[31, 31]
{ { 4.70614E-06, 5.23526E-06, 5.83516E-06, 6.51573E-06, 7.28844E-06,
8.1679E-06, 9.16875E-06, 1.03078E-05, 1.1645E-05, 1.30832E-05, 1.4764E-05,
1.66799E-05, 1.90898E-05, 2.13424E-05, 2.41476E-05, 2.73254E-05, 3.09175E-05,
3.49905E-05, 3.95895E-05, 4.48005E-05, 5.06657E-05, 5.74392E-05, 6.52223E-05,
7.43131E-05, 8.52091E-05, 9.84386E-05, 0.000115093, 0.00013682, 0.000165519,
0.000203702, 0.000236753 }, { 6.80273E-06, 7.5674E-06, 8.4343E-06, 9.41766E-06,
```

1.0534E-05, 1.18044E-05, 1.32498E-05, 1.48945E-05, 1.68245E-05, 1.88994E-05,
2.13231E-05, 2.40838E-05, 2.75529E-05, 3.07919E-05, 3.48196E-05, 3.93733E-05,
4.4508E-05, 5.03114E-05, 5.68372E-05, 6.41916E-05, 7.2412E-05, 8.18189E-05,
9.24966E-05, 0.000104762, 0.000119121, 0.00013596, 0.000156023, 0.000179767,
0.00020526, 0.000221428, 0.000160176 }, { 9.15628E-06, 1.01852E-05, 1.13515E-05,
1.26743E-05, 1.41758E-05, 1.58841E-05, 1.78272E-05, 2.00373E-05, 2.26298E-05,
2.54151E-05, 2.86662E-05, 3.23658E-05, 3.70085E-05, 4.13361E-05, 4.67065E-05,
5.2762E-05, 5.95663E-05, 6.72218E-05, 7.57799E-05, 8.53512E-05, 9.59431E-05,
0.000107904, 0.000121237, 0.000136174, 0.000153033, 0.000171713, 0.000191911,
0.000211437, 0.00022188, 0.000195694, 5.87785E-05 }, { 1.18929E-05, 1.32287E-05,
1.47428E-05, 1.64597E-05, 1.84081E-05, 2.06242E-05, 2.3144E-05, 2.60087E-05,
2.93668E-05, 3.2972E-05, 3.71759E-05, 4.19537E-05, 4.79386E-05, 5.35054E-05,
6.03951E-05, 6.81358E-05, 7.67931E-05, 8.64745E-05, 9.72118E-05, 0.000109096,
0.000122067, 0.000136444, 0.000152061, 0.000168918, 0.000186887, 0.000204967,
0.000221032, 0.000229003, 0.00021326, 0.000138238, 4.35317E-06 }, { 1.55469E-05,
1.72923E-05, 1.92701E-05, 2.15124E-05, 2.40563E-05, 2.69485E-05, 3.02356E-05,
3.39703E-05, 3.83449E-05, 4.30366E-05, 4.85003E-05, 5.46996E-05, 6.24472E-05,
6.96332E-05, 7.84958E-05, 8.8406E-05, 9.94214E-05, 0.000111641, 0.000125049,
0.00013968, 0.000155348, 0.00017226, 0.000189946, 0.000207962, 0.00022539,
0.000239802, 0.000246441, 0.000234601, 0.000183884, 7.1478E-05, 9.88625E-06 }, {
1.99726E-05, 2.22133E-05, 2.47519E-05, 2.76291E-05, 3.0892E-05, 3.46001E-05,
3.88119E-05, 4.35935E-05, 4.91887E-05, 5.5182E-05, 6.21501E-05, 7.00396E-05,
7.98711E-05, 8.89571E-05, 0.000100113, 0.000112513, 0.000126186, 0.000141194,
0.000157433, 0.000174819, 0.000192954, 0.000211805, 0.000230416, 0.000247642,
0.000261374, 0.000267324, 0.000257769, 0.000218554, 0.000132869, 1.87178E-05,
2.80013E-05 }, { 2.5382E-05, 2.82272E-05, 3.14497E-05, 3.51008E-05, 3.92395E-05,
4.394E-05, 4.9275E-05, 5.53261E-05, 6.23979E-05, 6.99607E-05, 7.87359E-05,
8.86451E-05, 0.000100948, 0.000112267, 0.000126085, 0.000141325, 0.000157958,
0.000175965, 0.000195087, 0.000215032, 0.000235076, 0.000254762, 0.000272434,
0.000285954, 0.000291665, 0.000283277, 0.000250751, 0.000180541, 7.29396E-05,
7.90451E-07, 1.72094E-05 }, { 3.27507E-05, 3.64181E-05, 4.05704E-05, 4.52727E-05,
5.05998E-05, 5.66457E-05, 6.35013E-05, 7.12677E-05, 8.03298E-05, 9.00016E-05,
0.000101195, 0.000113792, 0.000129359, 0.000143598, 0.000160854, 0.000179693,
0.000199976, 0.000221532, 0.000243838, 0.000266254, 0.000287543, 0.000306565,
0.000320675, 0.000326405, 0.000318278, 0.000288389, 0.000226571, 0.00012743,
2.30658E-05, 1.66318E-05, 9.40563E-07 }, { 4.26358E-05, 4.7404E-05, 5.28E-05,
5.89075E-05, 6.58214E-05, 7.36611E-05, 8.25402E-05, 9.25836E-05, 0.000104279,
0.000116731, 0.000131094, 0.00014719, 0.000166962, 0.000184911, 0.00020646,
0.000229674, 0.000254219, 0.000279653, 0.000305025, 0.000329134, 0.000349992,
0.000365433, 0.000371548, 0.000363354, 0.000333787, 0.000275055, 0.000181811,
6.76902E-05, 1.10692E-06, 3.85771E-05, 1.09839E-05 }, { 5.53376E-05, 6.15161E-05,
6.85042E-05, 7.6408E-05, 8.53473E-05, 9.54718E-05, 0.000106921, 0.000119848,
0.000134863, 0.000150796, 0.0001691, 0.000189498, 0.000214363, 0.000236716,
0.000263215, 0.000291257, 0.000320176, 0.000349076, 0.000376349, 0.00039996,
0.000416898, 0.000423631, 0.000414929, 0.000384645, 0.000325566, 0.00023407,
0.000117691, 1.81271E-05, 1.47548E-05, 3.30174E-05, 1.23418E-05 }, { 7.21018E-05,
8.01351E-05, 8.92147E-05, 9.94745E-05, 0.000111065, 0.000124173, 0.000138968,
0.00015563, 0.000174921, 0.000195307, 0.0002186, 0.000244372, 0.000275468,
0.00030306, 0.000335216, 0.000368409, 0.000401429, 0.000432651, 0.000459496,
0.000478742, 0.000486121, 0.000476509, 0.000443835, 0.000382188, 0.000287626,
0.000167422, 5.05532E-05, 1.46347E-06, 4.59438E-05, 6.64051E-06, 1.27365E-06 }, {
9.32565E-05, 0.000103619, 0.000115321, 0.000128529, 0.000143428, 0.000160245,
0.000179181, 0.00020044, 0.000224951, 0.000250715, 0.000279948, 0.00031199,
0.000350134, 0.000383391, 0.00042125, 0.000458973, 0.000494516, 0.000525186,
0.000547093, 0.000555553, 0.000545101, 0.00050936, 0.000443082, 0.000343873,
0.000216899, 8.77057E-05, 6.80798E-06, 2.5372E-05, 5.33806E-05, 6.48523E-06,
1.09951E-05 }, { 0.000123998, 0.00013773, 0.000153218, 0.000170673, 0.000190326,
0.000212454, 0.000237292, 0.000265062, 0.000296905, 0.000330139, 0.0003675,
0.000407935, 0.000455187, 0.000495379, 0.000539596, 0.000581315, 0.000617175,
0.000642862, 0.000652721, 0.000640389, 0.000599718, 0.000524359, 0.000413461,
0.000273952, 0.000127882, 2.24172E-05, 9.74228E-06, 6.5098E-05, 2.26325E-05,
2.55876E-05, 1.83505E-06 }, { 0.000161072, 0.000178832, 0.000198833, 0.000221331,

0.000246598, 0.000274958, 0.00030666, 0.000341914, 0.00038205, 0.000423547,
0.000469617, 0.000518623, 0.000574429, 0.000620226, 0.000668035, 0.000709193,
0.000738575, 0.000749902, 0.000735844, 0.00068914, 0.00060516, 0.000481276,
0.000327329, 0.000166634, 4.11575E-05, 3.22073E-06, 5.48792E-05, 6.6894E-05,
2.47885E-06, 8.58067E-06, 8.3145E-06 }, { 0.000209826, 0.000232831, 0.000258689,
0.000287701, 0.000320181, 0.000356485, 0.000396849, 0.000441417, 0.000491675,
0.000542981, 0.000598975, 0.000657108, 0.000720865, 0.000770368, 0.000817661,
0.000851458, 0.000864422, 0.000848207, 0.000794651, 0.000697804, 0.000558918,
0.000385365, 0.000205161, 6.10852E-05, 2.57405E-06, 4.36828E-05, 9.0894E-05,
2.23032E-05, 2.88319E-05, 9.15893E-06, 2.996E-06 }, { 0.000272996, 0.000302711,
0.000336025, 0.000373283, 0.00041482, 0.000460996, 0.000511972, 0.00056773,
0.000629802, 0.000692082, 0.000758445, 0.000824963, 0.000893822, 0.000942506,
0.000981349, 0.000996278, 0.00097772, 0.000916144, 0.000805229, 0.000645175,
0.000450261, 0.000245643, 8.04816E-05, 4.69118E-06, 3.60462E-05, 0.000101011,
6.30631E-05, 5.5398E-06, 2.58812E-05, 1.36118E-05, 3.21113E-06 }, { 0.000350797,
0.000388624, 0.000430892, 0.000477965, 0.000530157, 0.000587765, 0.000650761,
0.000718798, 0.000793217, 0.00086609, 0.000941088, 0.001012306, 0.00107916,
0.001118141, 0.001135081, 0.001113943, 0.001044141, 0.000917964, 0.0007366,
0.000514327, 0.000286328, 9.85977E-05, 7.5181E-06, 3.13222E-05, 0.000106379,
9.72788E-05, 8.31342E-06, 4.14373E-05, 3.91631E-06, 6.54365E-06, 4.31195E-06 }, {
0.000454303, 0.000502699, 0.000556547, 0.000616183, 0.000681828, 0.000753599,
0.000831087, 0.000913336, 0.001001102, 0.001084061, 0.001164995, 0.001235154,
0.001289064, 0.001305069, 0.001280863, 0.001200752, 0.00105646, 0.00084825,
0.000593937, 0.000331223, 0.000118804, 1.06886E-05, 3.00557E-05, 0.000113271,
0.000125683, 2.95384E-05, 2.40579E-05, 2.70857E-05, 2.50849E-05, 7.62446E-06,
3.72371E-06 }, { 0.00069728, 0.000770515, 0.000851589, 0.000940788, 0.001038133,
0.001143343, 0.001255169, 0.001371309, 0.001491326, 0.001599422, 0.001696808,
0.001768718, 0.001800234, 0.001773063, 0.001670871, 0.001480941, 0.001203285,
0.000857811, 0.000495651, 0.000191969, 2.68123E-05, 3.07585E-05, 0.000137292,
0.000183261, 7.82872E-05, 1.12326E-05, 6.68897E-05, 9.17717E-06, 8.78542E-06,
1.00108E-05, 3.02317E-06 }, { 0.000744009, 0.000820443, 0.000904386, 0.000995779,
0.001094135, 0.001198447, 0.001306434, 0.001414396, 0.001519536, 0.00160535,
0.001668957, 0.00169356, 0.001655879, 0.001559333, 0.001373296, 0.001105039,
0.000776785, 0.000436709, 0.000159397, 1.66149E-05, 3.17679E-05, 0.000134058,
0.000174591, 7.98336E-05, 5.76379E-06, 6.29998E-05, 1.18483E-05, 3.09335E-05,
1.80641E-05, 5.22101E-06, 2.71002E-06 }, { 0.000936111, 0.001029755, 0.001131594,
0.001241037, 0.00135676, 0.001476514, 0.001596168, 0.001709471, 0.00180997,
0.001877951, 0.001905116, 0.001870765, 0.001740275, 0.001548344, 0.001248427,
0.000879942, 0.000498242, 0.000184016, 2.0239E-05, 3.38579E-05, 0.000145923,
0.000198332, 0.000102922, 6.77034E-06, 6.00441E-05, 3.69371E-05, 2.66086E-05,
5.75829E-06, 6.87514E-06, 4.46117E-06, 2.68918E-06 }, { 0.001172006, 0.001285248,
0.001406798, 0.001535131, 0.001667536, 0.001799779, 0.001924935, 0.002033098,
0.002112461, 0.002140882, 0.002103064, 0.001976782, 0.001718392, 0.001411808,
0.000999821, 0.000570275, 0.000214935, 2.50908E-05, 3.49329E-05, 0.000160183,
0.00022414, 0.000126508, 1.06039E-05, 5.50609E-05, 6.18584E-05, 1.169E-05,
2.31165E-05, 2.13418E-05, 7.38285E-06, 4.6409E-06, 1.58854E-06 }, { 0.001545953,
0.00168886, 0.001839626, 0.001995048, 0.002149981, 0.00229682, 0.002424067,
0.002516125, 0.002553081, 0.00251129, 0.002366846, 0.002100761, 0.001666333,
0.001223211, 0.000710803, 0.000278666, 3.83359E-05, 3.55515E-05, 0.000180342,
0.000267228, 0.000166353, 1.99596E-05, 5.19843E-05, 8.87326E-06, 8.76518E-06,
4.69365E-05, 1.70898E-05, 1.05395E-05, 1.00464E-05, 5.12801E-06, 2.14082E-06 }, {
0.00180517, 0.001962004, 0.002123325, 0.002283681, 0.002434904, 0.002565465,
0.002659146, 0.002695295, 0.002646826, 0.002496929, 0.00221826, 0.001809746,
0.001245921, 0.000757203, 0.00029914, 4.12326E-05, 3.46042E-05, 0.000187477,
0.000283302, 0.000179804, 2.35264E-05, 4.83366E-05, 0.000100016, 1.18512E-05,
5.06221E-05, 7.73716E-06, 1.40051E-05, 8.12512E-06, 7.92474E-06, 4.06353E-06,
1.71535E-06 }, { 0.002203864, 0.002380338, 0.002555498, 0.002720398, 0.002862332,
0.002964163, 0.003003455, 0.002953845, 0.002781219, 0.002483952, 0.002037935,
0.001476663, 0.000818744, 0.000356158, 5.43996E-05, 3.23463E-05, 0.00019794,
0.000315428, 0.000212839, 3.22767E-05, 4.45216E-05, 0.000115489, 2.13824E-05,
4.77461E-05, 1.54256E-05, 3.19245E-05, 2.01211E-05, 1.16202E-05, 6.22589E-06,
2.72353E-06, 1.36618E-06 }, { 0.002671172, 0.00286233, 0.003042018, 0.003196404,

```
0.003306834, 0.003349451, 0.003295617, 0.003115587, 0.002773309, 0.002297758,  
0.001682357, 0.001015788, 0.000382429, 7.32452E-05, 2.867E-05, 0.000206421,  
0.000349996, 0.000252593, 4.57665E-05, 4.11899E-05, 0.000130379, 3.47054E-05,  
4.21015E-05, 3.23405E-05, 3.40308E-05, 1.11974E-05, 9.12157E-06, 9.97557E-06,  
5.02382E-06, 3.48361E-06, 1.2197E-06 }, { 0.003174823, 0.003368362, 0.003534405,  
0.003652962, 0.003698577, 0.003640922, 0.003447882, 0.003093097, 0.002550544,  
0.00189967, 0.001170011, 0.000515287, 7.67128E-05, 2.35501E-05, 0.000208804,  
0.000382469, 0.000298038, 6.44566E-05, 3.50952E-05, 0.000143351, 5.20405E-05,  
3.49749E-05, 5.0516E-05, 2.61914E-05, 1.1167E-05, 1.50878E-05, 9.20275E-06,  
7.82286E-06, 4.5738E-06, 2.15306E-06, 1.13538E-06 }, { 0.003727687, 0.003905881,  
0.004032975, 0.004081829, 0.004020349, 0.003814185, 0.00343438, 0.002870435,  
0.00212569, 0.001349414, 0.000620276, 0.000135859, 2.5936E-05, 0.000205829,  
0.000413962, 0.000351102, 9.19151E-05, 2.76994E-05, 0.000154054, 7.28769E-05,  
2.69689E-05, 6.83755E-05, 1.7809E-05, 2.4133E-05, 2.76433E-05, 1.84311E-05,  
1.03569E-05, 6.33248E-06, 5.06849E-06, 2.41637E-06, 1.01372E-06 }, { 0.004314392,  
0.004450273, 0.004502472, 0.004436857, 0.004217245, 0.003811853, 0.003207579,  
0.002429012, 0.001533518, 0.000746465, 0.000184561, 1.40253E-05, 0.000223603,  
0.000442626, 0.000412281, 0.000129908, 2.00072E-05, 0.0001607, 0.00010102,  
2.01235E-05, 8.47449E-05, 1.26821E-05, 4.0344E-05, 2.83716E-05, 1.45232E-05,  
1.23074E-05, 9.22176E-06, 5.34538E-06, 3.03955E-06, 2.10933E-06, 8.28259E-07 }, {  
0.004884183, 0.004939521, 0.004870015, 0.004637307, 0.004208006, 0.003565648,  
0.002732495, 0.001791019, 0.000871855, 0.000247155, 1.34249E-05, 0.000179363,  
0.00048144, 0.000478186, 0.000181334, 1.4512E-05, 0.000159891, 0.00013512,  
1.44466E-05, 9.70106E-05, 1.32483E-05, 5.3888E-05, 2.11776E-05, 1.11588E-05,  
1.07384E-05, 9.59538E-06, 7.82019E-06, 5.78382E-06, 3.90223E-06, 1.83223E-06,  
7.25826E-07 }, { 0.00540995, 0.005336635, 0.005091017, 0.004637274, 0.003957459,  
0.003069767, 0.002055197, 0.001070591, 0.000313419, 1.88347E-05, 0.00015678,  
0.00047506, 0.000532292, 0.000248931, 1.50477E-05, 0.00015029, 0.000174783,  
1.43647E-05, 0.000103214, 2.17638E-05, 6.10281E-05, 1.42295E-05, 1.88045E-05,  
1.99243E-05, 1.33125E-05, 9.0956E-06, 7.28093E-06, 4.93398E-06, 2.71988E-06,  
1.78861E-06, 6.53887E-07 } };
```

```
matriz600 = new double[31, 31]  
{ { 2.35E-06, 2.62E-06, 2.92E-06, 3.26E-06, 3.64E-06, 4.08E-06,  
4.58E-06, 5.15E-06, 5.82E-06, 6.54E-06, 7.38E-06, 8.34E-06, 9.54E-06, 1.07E-05,  
1.21E-05, 1.37E-05, 1.55E-05, 1.75E-05, 1.98E-05, 2.24E-05, 2.53E-05, 2.87E-05,  
3.26E-05, 3.72E-05, 4.26E-05, 4.92E-05, 5.75E-05, 6.84E-05, 8.28E-05, 1.02E-04,  
1.18E-04 }, { 3.40E-06, 3.78E-06, 4.21E-06, 4.71E-06, 5.26E-06, 5.90E-06, 6.62E-  
06, 7.45E-06, 8.41E-06, 9.45E-06, 1.07E-05, 1.20E-05, 1.38E-05, 1.54E-05, 1.74E-  
05, 1.97E-05, 2.23E-05, 2.52E-05, 2.84E-05, 3.21E-05, 3.62E-05, 4.09E-05, 4.62E-  
05, 5.24E-05, 5.96E-05, 6.80E-05, 7.80E-05, 8.99E-05, 1.03E-04, 1.11E-04, 8.01E-  
05 }, { 4.57E-06, 5.09E-06, 5.67E-06, 6.33E-06, 7.09E-06, 7.94E-06, 8.91E-06,  
1.00E-05, 1.13E-05, 1.27E-05, 1.43E-05, 1.62E-05, 1.85E-05, 2.07E-05, 2.34E-05,  
2.64E-05, 2.98E-05, 3.36E-05, 3.79E-05, 4.27E-05, 4.80E-05, 5.40E-05, 6.06E-05,  
6.81E-05, 7.65E-05, 8.59E-05, 9.60E-05, 1.06E-04, 1.11E-04, 9.78E-05, 2.94E-05 },  
{ 5.94E-06, 6.61E-06, 7.37E-06, 8.23E-06, 9.20E-06, 1.03E-05, 1.16E-05, 1.30E-05,  
1.47E-05, 1.65E-05, 1.86E-05, 2.10E-05, 2.40E-05, 2.68E-05, 3.02E-05, 3.41E-05,  
3.84E-05, 4.32E-05, 4.86E-05, 5.45E-05, 6.10E-05, 6.82E-05, 7.60E-05, 8.45E-05,  
9.34E-05, 1.02E-04, 1.11E-04, 1.15E-04, 1.07E-04, 6.91E-05, 2.18E-05 }, { 7.77E-  
06, 8.64E-06, 9.63E-06, 1.08E-05, 1.20E-05, 1.35E-05, 1.51E-05, 1.70E-05, 1.92E-  
05, 2.15E-05, 2.42E-05, 2.73E-05, 3.12E-05, 3.48E-05, 3.92E-05, 4.42E-05, 4.97E-  
05, 5.58E-05, 6.25E-05, 6.98E-05, 7.77E-05, 8.61E-05, 9.50E-05, 1.04E-04, 1.13E-  
04, 1.20E-04, 1.23E-04, 1.17E-04, 9.19E-05, 3.57E-05, 4.94E-06 }, { 9.98E-06,  
1.11E-05, 1.24E-05, 1.38E-05, 1.54E-05, 1.73E-05, 1.94E-05, 2.18E-05, 2.46E-05,  
2.76E-05, 3.11E-05, 3.50E-05, 3.99E-05, 4.45E-05, 5.01E-05, 5.63E-05, 6.31E-05,  
7.06E-05, 7.87E-05, 8.74E-05, 9.65E-05, 1.06E-04, 1.15E-04, 1.24E-04, 1.31E-04,  
1.34E-04, 1.29E-04, 1.09E-04, 6.64E-05, 9.36E-06, 1.40E-05 }, { 1.27E-05, 1.41E-  
05, 1.57E-05, 1.75E-05, 1.96E-05, 2.20E-05, 2.46E-05, 2.77E-05, 3.12E-05, 3.50E-  
05, 3.94E-05, 4.43E-05, 5.05E-05, 5.61E-05, 6.30E-05, 7.07E-05, 7.90E-05, 8.80E-  
05, 9.75E-05, 1.08E-04, 1.18E-04, 1.27E-04, 1.36E-04, 1.43E-04, 1.46E-04, 1.42E-  
04, 1.25E-04, 9.03E-05, 3.65E-05, 3.95E-07, 8.60E-06 }, { 1.64E-05, 1.82E-05,  
2.03E-05, 2.26E-05, 2.53E-05, 2.83E-05, 3.17E-05, 3.56E-05, 4.02E-05, 4.50E-05,
```

5.06E-05, 5.69E-05, 6.47E-05, 7.18E-05, 8.04E-05, 8.98E-05, 1.00E-04,
0.000110766, 0.00012192, 0.000133128, 0.000143773, 1.53E-04, 1.60E-04, 1.63E-04,
1.59E-04, 1.44E-04, 1.13E-04, 6.37E-05, 1.15E-05, 8.32E-06, 4.70E-07 }, { 2.13E-
05, 2.37E-05, 2.64E-05, 2.94E-05, 3.29E-05, 3.68E-05, 4.13E-05, 4.63E-05, 5.21E-
05, 5.84E-05, 6.55E-05, 7.36E-05, 8.35E-05, 9.25E-05, 0.000103229, 0.000114836,
0.00012711, 0.000139827, 0.000152514, 0.000164569, 0.000174998, 0.000182719,
0.000185776, 1.82E-04, 1.67E-04, 1.38E-04, 9.09E-05, 3.38E-05, 5.53E-07, 1.93E-
05, 5.49E-06 }, { 2.76E-05, 3.07E-05, 3.42E-05, 3.82E-05, 4.27E-05, 4.77E-05,
5.34E-05, 5.99E-05, 6.74E-05, 7.54E-05, 8.45E-05, 9.47E-05, 0.000107179,
0.000118356, 0.000131607, 0.000145629, 0.000160089, 0.00017454, 0.000188177,
0.000199983, 0.000208453, 0.000211819, 0.000207468, 0.000192326, 0.000162785,
1.17E-04, 5.88E-05, 9.06E-06, 7.38E-06, 1.65E-05, 6.17E-06 }, { 3.60E-05, 4.00E-
05, 4.46E-05, 4.97E-05, 5.55E-05, 6.21E-05, 6.95E-05, 7.78E-05, 8.75E-05, 9.76E-
05, 0.000109294, 0.000122182, 0.000137731, 0.000151529, 0.000167608, 0.000184206,
0.000200717, 0.00021633, 0.000229753, 0.000239377, 0.000243067, 0.00023826,
0.000221922, 0.000191097, 0.000143814, 8.37E-05, 2.53E-05, 7.32E-07, 2.30E-05,
3.32E-06, 6.37E-07 }, { 4.66E-05, 5.18E-05, 5.76E-05, 6.42E-05, 7.17E-05, 8.01E-
05, 8.96E-05, 0.000100205, 0.000112464, 0.000125348, 0.000139967, 0.000155991,
0.000175066, 0.000191696, 0.000210628, 0.000229491, 0.000247264, 0.0002626,
0.000273555, 0.000277785, 0.000272559, 0.000254687, 0.000221546, 0.000171939,
0.000108449, 4.39E-05, 3.40E-06, 1.27E-05, 2.67E-05, 3.24E-06, 5.50E-06 }, {
6.19E-05, 6.88E-05, 7.66E-05, 8.53E-05, 9.51E-05, 1.06E-04, 0.000118624,
0.000132513, 0.000148439, 0.00016506, 0.000183744, 0.000203965, 0.000227595,
0.000247694, 0.000269805, 0.000290667, 0.000308599, 0.000321444, 0.000326374,
0.000320207, 0.00029987, 0.000262187, 0.000206735, 0.000136976, 6.39E-05, 1.12E-
05, 4.87E-06, 3.26E-05, 1.13E-05, 1.28E-05, 9.17E-07 }, { 8.05E-05, 8.94E-05,
9.94E-05, 1.11E-04, 1.23E-04, 1.37E-04, 1.53E-04, 0.000170937, 0.000191011,
0.000211765, 0.000234805, 0.000259313, 0.000287221, 0.000310123, 0.000334031,
0.000354613, 0.000369306, 0.00037497, 0.000367941, 0.000344586, 0.000302592,
0.000240644, 0.000163665, 8.33E-05, 2.06E-05, 1.61E-06, 2.74E-05, 3.34E-05,
1.24E-06, 4.29E-06, 4.16E-06 }, { 1.05E-04, 1.16E-04, 1.29E-04, 1.44E-04, 1.60E-
04, 1.78E-04, 1.98E-04, 2.21E-04, 0.000245825, 0.000271485, 0.00029949,
0.000328563, 0.000360448, 0.000385204, 0.000408855, 0.000425756, 0.000432239,
0.00042413, 0.000397349, 0.000348919, 0.000279468, 0.000192684, 0.000102576,
3.05E-05, 1.29E-06, 2.18E-05, 4.54E-05, 1.11E-05, 1.44E-05, 4.58E-06, 1.50E-06 },
{ 1.36E-04, 1.51E-04, 1.68E-04, 1.87E-04, 2.07E-04, 0.000230457, 2.56E-04, 2.84E-
04, 3.15E-04, 0.000346044, 0.000379236, 0.000412503, 0.000446942, 0.000471289,
0.000490715, 0.000498181, 0.0004889, 0.000458107, 0.00040264, 0.000322602,
0.000225133, 0.000122815, 4.02E-05, 2.34E-06, 1.80E-05, 5.05E-05, 3.15E-05,
2.77E-06, 1.29E-05, 6.80E-06, 1.60E-06 }, { 1.75E-04, 1.94E-04, 2.15E-04, 2.39E-
04, 2.65E-04, 2.94E-04, 0.000325355, 3.59E-04, 3.97E-04, 4.33E-04, 0.000470576,
0.000506197, 0.000539633, 0.000559129, 0.000567601, 0.00055703, 0.000522121,
0.000459019, 0.000368321, 0.000257167, 0.000143155, 4.93E-05, 3.76E-06, 1.57E-05,
5.32E-05, 4.86E-05, 4.15E-06, 2.07E-05, 1.96E-06, 3.27E-06, 2.16E-06 }, { 2.27E-
04, 2.51E-04, 2.78E-04, 3.08E-04, 3.41E-04, 3.77E-04, 0.000415531, 0.000456677,
5.01E-04, 5.42E-04, 0.000582562, 0.000617655, 0.000644619, 0.000652625,
0.000640518, 0.00060045, 0.000528285, 0.000424156, 0.000296974, 0.000165599,
5.94E-05, 5.34E-06, 1.50E-05, 5.67E-05, 6.28E-05, 1.48E-05, 1.20E-05, 1.35E-05,
1.25E-05, 3.81E-06, 1.86E-06 }, { 3.48E-04, 3.85E-04, 4.26E-04, 4.70E-04, 5.19E-
04, 5.72E-04, 6.28E-04, 0.000685704, 0.000745746, 8.00E-04, 8.49E-04,
0.000884508, 0.000900272, 0.000886682, 0.000835567, 0.000740569, 0.0006017,
0.000428919, 0.000247806, 9.60E-05, 1.34E-05, 1.54E-05, 6.87E-05, 9.16E-05,
3.91E-05, 5.62E-06, 3.34E-05, 4.59E-06, 4.39E-06, 5.01E-06, 1.51E-06 }, { 3.72E-
04, 4.10E-04, 4.52E-04, 4.98E-04, 5.47E-04, 5.99E-04, 6.53E-04, 7.07E-04,
0.000759908, 0.000802842, 8.35E-04, 8.47E-04, 0.000828121, 0.000779825,
0.000686766, 0.000552586, 0.000388405, 0.000218328, 7.97E-05, 8.29E-06, 1.59E-05,
6.71E-05, 8.73E-05, 3.99E-05, 2.89E-06, 3.15E-05, 5.91E-06, 1.55E-05, 9.03E-06,
2.61E-06, 1.36E-06 }, { 4.68E-04, 5.15E-04, 5.66E-04, 6.20E-04, 6.78E-04, 7.38E-
04, 7.98E-04, 8.55E-04, 0.000905226, 0.000939242, 0.000952833, 9.36E-04, 8.70E-
04, 0.000774343, 0.00062431, 0.000439991, 0.000249084, 9.20E-05, 1.01E-05, 1.70E-
05, 7.30E-05, 9.92E-05, 5.14E-05, 3.38E-06, 3.01E-05, 1.84E-05, 1.33E-05, 2.88E-
06, 3.44E-06, 2.23E-06, 1.34E-06 }, { 5.86E-04, 6.43E-04, 7.03E-04, 7.68E-04,

```
8.34E-04, 9.00E-04, 9.63E-04, 1.02E-03, 1.06E-03, 0.001070835, 0.00105191,
0.00098872, 8.59E-04, 7.06E-04, 0.000499941, 0.000285086, 0.000107392, 1.25E-05,
1.75E-05, 8.02E-05, 0.000112102, 6.32E-05, 5.29E-06, 2.76E-05, 3.09E-05, 5.87E-
06, 1.15E-05, 1.07E-05, 3.69E-06, 2.32E-06, 7.95E-07 }, { 7.73E-04, 8.44E-04,
9.20E-04, 9.98E-04, 1.08E-03, 1.15E-03, 1.21E-03, 1.26E-03, 1.28E-03, 1.26E-03,
0.001183923, 0.001050759, 0.000833371, 6.12E-04, 3.55E-04, 0.000139218, 1.91E-05,
1.78E-05, 9.03E-05, 0.00013367, 8.31E-05, 9.95E-06, 2.61E-05, 4.43E-05, 4.38E-06,
2.35E-05, 8.57E-06, 5.28E-06, 5.02E-06, 2.57E-06, 1.07E-06 }, { 9.03E-04, 9.81E-
04, 1.06E-03, 1.14E-03, 1.22E-03, 1.28E-03, 1.33E-03, 1.35E-03, 0.00132418,
1.25E-03, 1.11E-03, 0.000905173, 0.000623018, 0.000378505, 1.49E-04, 2.05E-05,
1.74E-05, 9.39E-05, 0.000141732, 8.98E-05, 1.17E-05, 2.43E-05, 5.00E-05, 5.90E-
06, 2.53E-05, 3.88E-06, 6.98E-06, 4.05E-06, 3.96E-06, 2.03E-06, 8.58E-07 }, {
1.10E-03, 1.19E-03, 1.28E-03, 1.36E-03, 1.43E-03, 1.48E-03, 1.50E-03, 1.48E-03,
1.39E-03, 0.001242705, 1.02E-03, 7.38E-04, 0.000409229, 0.000177859, 2.71E-05,
1.63E-05, 9.92E-05, 0.000157842, 0.000106312, 1.60E-05, 2.24E-05, 5.78E-05,
1.06E-05, 2.39E-05, 7.67E-06, 1.60E-05, 1.01E-05, 5.80E-06, 3.11E-06, 1.36E-06,
6.83E-07 }, { 1.34E-03, 1.43E-03, 1.52E-03, 1.60E-03, 1.65E-03, 1.68E-03, 1.65E-
03, 1.56E-03, 1.39E-03, 1.15E-03, 0.000841375, 5.08E-04, 1.91E-04, 3.64E-05,
1.45E-05, 0.000103553, 1.75E-04, 1.26E-04, 2.27E-05, 2.08E-05, 6.52E-05, 1.72E-
05, 2.12E-05, 1.61E-05, 1.71E-05, 5.64E-06, 4.58E-06, 4.99E-06, 2.51E-06, 1.75E-
06, 6.13E-07 }, { 1.59E-03, 1.69E-03, 1.77E-03, 1.83E-03, 1.85E-03, 1.82E-03,
1.73E-03, 1.55E-03, 1.28E-03, 9.50E-04, 5.85E-04, 0.000257217, 3.81E-05, 1.19E-
05, 0.000104869, 0.00019155, 0.000148866, 3.20E-05, 1.77E-05, 7.18E-05, 2.58E-05,
1.76E-05, 2.51E-05, 1.32E-05, 5.56E-06, 7.50E-06, 4.59E-06, 3.91E-06, 2.30E-06,
1.08E-06, 5.64E-07 }, { 1.87E-03, 1.96E-03, 2.02E-03, 2.04E-03, 2.01E-03, 1.91E-
03, 1.72E-03, 1.44E-03, 1.06E-03, 6.74E-04, 3.10E-04, 6.75E-05, 1.32E-05, 1.04E-
04, 2.07E-04, 0.000175394, 4.56E-05, 1.41E-05, 7.72E-05, 3.62E-05, 1.37E-05,
3.41E-05, 9.03E-06, 1.20E-05, 1.38E-05, 9.22E-06, 5.16E-06, 3.17E-06, 2.54E-06,
1.20E-06, 5.08E-07 }, { 2.16E-03, 2.23E-03, 2.26E-03, 2.22E-03, 2.11E-03, 1.91E-
03, 1.61E-03, 1.22E-03, 7.66E-04, 3.72E-04, 9.16E-05, 7.12E-06, 0.000112661,
0.000222055, 2.06E-04, 6.44E-05, 1.02E-05, 8.07E-05, 5.02E-05, 1.03E-05, 4.23E-
05, 6.43E-06, 2.00E-05, 1.43E-05, 7.35E-06, 6.20E-06, 4.61E-06, 2.68E-06, 1.51E-
06, 1.05E-06, 4.14E-07 }, { 2.45E-03, 2.48E-03, 2.44E-03, 2.32E-03, 2.11E-03,
1.79E-03, 1.37E-03, 8.95E-04, 4.35E-04, 1.23E-04, 6.72E-06, 9.07E-05, 2.42E-04,
0.000239083, 8.99E-05, 7.39E-06, 8.06E-05, 6.72E-05, 7.36E-06, 4.86E-05, 6.59E-
06, 2.69E-05, 1.08E-05, 5.59E-06, 5.36E-06, 4.83E-06, 3.92E-06, 2.92E-06, 1.95E-
06, 9.23E-07, 3.68E-07 }, { 2.71E-03, 2.68E-03, 2.55E-03, 2.32E-03, 1.98E-03,
1.54E-03, 1.03E-03, 5.34E-04, 1.55E-04, 9.25E-06, 7.97E-05, 2.39E-04,
0.000266206, 1.23E-04, 7.47E-06, 7.61E-05, 8.70E-05, 7.15E-06, 5.19E-05, 1.06E-
05, 3.06E-05, 7.27E-06, 9.25E-06, 9.85E-06, 6.60E-06, 4.56E-06, 3.67E-06, 2.43E-
06, 1.38E-06, 8.91E-07, 3.22E-07 } };
```

```
matriz800 = new double[31, 31]
{ { 1.76288E-06, 1.9615E-06, 2.18664E-06, 2.44201E-06, 2.73193E-06,
3.06186E-06, 3.4373E-06, 3.86456E-06, 4.36611E-06, 4.90551E-06, 5.53588E-06,
6.25444E-06, 7.15819E-06, 8.00299E-06, 9.05499E-06, 1.02467E-05, 1.15938E-05,
1.31212E-05, 1.48459E-05, 1.68E-05, 1.89995E-05, 2.15395E-05, 2.44582E-05,
2.78673E-05, 3.19533E-05, 3.69144E-05, 4.31599E-05, 5.13076E-05, 6.20695E-05,
7.63883E-05, 8.87825E-05 }, { 2.54825E-06, 2.83528E-06, 3.16063E-06, 3.52962E-06,
3.94847E-06, 4.42506E-06, 4.96728E-06, 5.58418E-06, 6.30809E-06, 7.08631E-06,
7.99528E-06, 9.03068E-06, 1.03317E-05, 1.15464E-05, 1.30568E-05, 1.47646E-05,
1.66901E-05, 1.88664E-05, 2.13137E-05, 2.40716E-05, 2.71543E-05, 3.06819E-05,
3.46861E-05, 3.92857E-05, 4.46705E-05, 5.09849E-05, 5.85087E-05, 6.74127E-05,
7.69726E-05, 8.30357E-05, 6.0066E-05 }, { 3.42988E-06, 3.8161E-06, 4.25381E-06,
4.75019E-06, 5.31354E-06, 5.95441E-06, 6.68332E-06, 7.51233E-06, 8.48469E-06,
9.52937E-06, 1.07487E-05, 1.21362E-05, 1.38773E-05, 1.55003E-05, 1.75143E-05,
1.97852E-05, 2.23369E-05, 2.52078E-05, 2.84171E-05, 3.20064E-05, 3.59785E-05,
4.04637E-05, 4.54638E-05, 5.10653E-05, 5.73873E-05, 6.43926E-05, 7.19668E-05,
7.92889E-05, 8.32053E-05, 7.33854E-05, 2.20419E-05 }, { 4.45499E-06, 4.95644E-06,
5.52467E-06, 6.16893E-06, 6.89995E-06, 7.73132E-06, 8.67655E-06, 9.75109E-06,
1.10107E-05, 1.23628E-05, 1.39395E-05, 1.57314E-05, 1.79759E-05, 2.00636E-05,
2.26474E-05, 2.55503E-05, 2.87968E-05, 3.24275E-05, 3.64541E-05, 4.09107E-05,
```

4.5775E-05, 5.11664E-05, 5.7023E-05, 6.33443E-05, 7.00829E-05, 7.68628E-05,
8.28874E-05, 8.58765E-05, 7.99727E-05, 5.18394E-05, 1.63241E-06 }, { 5.82378E-06,
6.47895E-06, 7.22123E-06, 8.06265E-06, 9.01708E-06, 1.01021E-05, 1.13352E-05,
1.27361E-05, 1.43769E-05, 1.61366E-05, 1.81858E-05, 2.05107E-05, 2.34163E-05,
2.61113E-05, 2.9435E-05, 3.31514E-05, 3.72824E-05, 4.18648E-05, 4.68931E-05,
5.23799E-05, 5.82555E-05, 6.45977E-05, 7.12298E-05, 7.79861E-05, 8.45217E-05,
8.9926E-05, 9.2416E-05, 8.79756E-05, 6.89568E-05, 2.68042E-05, 3.70743E-06 }, {
7.48164E-06, 8.32275E-06, 9.27549E-06, 1.03551E-05, 1.15794E-05, 1.29705E-05,
1.45504E-05, 1.6344E-05, 1.84427E-05, 2.06906E-05, 2.3304E-05, 2.62629E-05,
2.995E-05, 3.33575E-05, 3.75414E-05, 4.21914E-05, 4.7319E-05, 5.29474E-05,
5.90373E-05, 6.55572E-05, 7.2358E-05, 7.94272E-05, 8.64063E-05, 9.28662E-05,
9.80161E-05, 0.000100247, 9.66641E-05, 8.19583E-05, 4.98259E-05, 7.01903E-06,
1.05006E-05 }, { 9.50802E-06, 1.05761E-05, 1.17855E-05, 1.31555E-05, 1.47083E-05,
1.64718E-05, 1.84731E-05, 2.07429E-05, 2.33954E-05, 2.6232E-05, 2.95232E-05,
3.32395E-05, 3.78536E-05, 4.20984E-05, 4.72807E-05, 5.29958E-05, 5.92335E-05,
6.59866E-05, 7.31575E-05, 8.06374E-05, 8.8154E-05, 9.55363E-05, 0.000102164,
0.000107234, 0.000109375, 0.00010623, 9.40325E-05, 6.77033E-05, 2.73522E-05,
2.96436E-07, 6.45341E-06 }, { 1.22684E-05, 1.36451E-05, 1.52034E-05, 1.6968E-05,
1.89667E-05, 2.12349E-05, 2.38067E-05, 2.67199E-05, 3.01189E-05, 3.37465E-05,
3.79447E-05, 4.26693E-05, 4.85076E-05, 5.38474E-05, 6.0319E-05, 6.73839E-05,
7.49906E-05, 8.30746E-05, 9.14397E-05, 9.9846E-05, 0.00010783, 0.000114963,
0.000120254, 0.000122403, 0.000119356, 0.000108147, 8.49646E-05, 4.77863E-05,
8.64941E-06, 6.23726E-06, 3.52686E-07 }, { 1.59715E-05, 1.77613E-05, 1.97865E-05,
2.20784E-05, 2.46725E-05, 2.76137E-05, 3.09445E-05, 3.47119E-05, 3.90989E-05,
4.3769E-05, 4.91562E-05, 5.51929E-05, 6.26083E-05, 6.934E-05, 7.74212E-05,
8.61271E-05, 9.53323E-05, 0.00010487, 0.000114385, 0.000123427, 0.000131249,
0.00013704, 0.000139333, 0.00013626, 0.000125172, 0.000103147, 6.81795E-05,
2.53834E-05, 4.15054E-07, 1.44668E-05, 4.11923E-06 }, { 2.07297E-05, 2.3049E-05,
2.56718E-05, 2.86377E-05, 3.19918E-05, 3.57902E-05, 4.00855E-05, 4.49343E-05,
5.05663E-05, 5.65424E-05, 6.34076E-05, 7.10582E-05, 8.03836E-05, 8.87669E-05,
9.8705E-05, 0.000109222, 0.000120067, 0.000130905, 0.000141133, 0.000149988,
0.00015634, 0.000158865, 0.000155602, 0.000144245, 0.000122089, 8.77767E-05,
4.41336E-05, 6.79714E-06, 5.53368E-06, 1.23812E-05, 4.62794E-06 }, { 2.701E-05,
3.00256E-05, 3.34333E-05, 3.72835E-05, 4.16324E-05, 4.65501E-05, 5.21003E-05,
5.83505E-05, 6.55866E-05, 7.3233E-05, 8.19695E-05, 9.16357E-05, 0.000103298,
0.000113646, 0.000125706, 0.000138155, 0.000150538, 0.000162248, 0.000172316,
0.000179533, 0.000182301, 0.000178696, 0.000166442, 0.000143323, 0.000107861,
6.27831E-05, 1.89565E-05, 5.48945E-07, 1.72299E-05, 2.48971E-06, 4.77719E-07 }, {
3.49351E-05, 3.88254E-05, 4.32174E-05, 4.81738E-05, 5.37642E-05, 6.00738E-05,
6.71777E-05, 7.51523E-05, 8.43466E-05, 9.40102E-05, 0.000104975, 0.000116993,
0.000131299, 0.000143772, 0.000157971, 0.000172119, 0.000185449, 0.000196951,
0.000205167, 0.00020834, 0.00020442, 0.000191016, 0.00016616, 0.000128955,
8.13371E-05, 3.28882E-05, 2.55236E-06, 9.516E-06, 2.00177E-05, 2.4327E-06,
4.1233E-06 }, { 4.6452E-05, 5.16071E-05, 5.74204E-05, 6.39709E-05, 7.13451E-05,
7.96475E-05, 8.89656E-05, 9.93829E-05, 0.000111328, 0.000123794, 0.000137807,
0.000152973, 0.000170696, 0.000185771, 0.000202354, 0.000218001, 0.00023145,
0.000241084, 0.000244782, 0.000240157, 0.000224903, 0.000196641, 0.000155051,
0.000102732, 4.79537E-05, 8.40486E-06, 3.65467E-06, 2.44137E-05, 8.48576E-06,
9.59577E-06, 6.88028E-07 }, { 6.03421E-05, 6.70096E-05, 7.45168E-05, 8.29598E-05,
9.24411E-05, 0.000103082, 0.000114975, 0.000128201, 0.000143257, 0.000158823,
0.000176103, 0.000194485, 0.000215416, 0.000232593, 0.000250525, 0.000265961,
0.000276982, 0.00028123, 0.000275957, 0.000258441, 0.000226945, 0.000180484,
0.000122749, 6.24849E-05, 1.54313E-05, 1.20852E-06, 2.05833E-05, 2.50846E-05,
9.30072E-07, 3.21653E-06, 3.1179E-06 }, { 7.86087E-05, 8.72457E-05, 9.69515E-05,
0.00010784, 0.000120028, 0.00013365, 0.000148794, 0.000165514, 0.000184367,
0.000203613, 0.000224617, 0.000246423, 0.000270337, 0.000288906, 0.000306644,
0.00031932, 0.000324183, 0.000318101, 0.000298014, 0.000261691, 0.000209602,
0.000144513, 7.69316E-05, 2.29026E-05, 9.65213E-07, 1.63861E-05, 3.40874E-05,
8.36076E-06, 1.08145E-05, 3.43642E-06, 1.12416E-06 }, { 0.000102278, 0.000113435,
0.00012594, 0.000139924, 0.000155511, 0.000172838, 0.000191964, 0.000212883,
0.00023617, 0.000259534, 0.000284428, 0.00030938, 0.00033521, 0.000353471,
0.000368041, 0.00037364, 0.00036668, 0.000343584, 0.000301983, 0.000241953,

0.00016885, 9.21109E-05, 3.01739E-05, 1.75801E-06, 1.35239E-05, 3.7885E-05,
2.36447E-05, 2.07957E-06, 9.70285E-06, 5.10266E-06, 1.20366E-06 }, { 0.000131432,
0.000145635, 0.000161503, 0.000179171, 0.000198758, 0.000220376, 0.000244014,
0.000269541, 0.000297461, 0.000324801, 0.000352936, 0.000379652, 0.000404731,
0.000419354, 0.000425708, 0.000417779, 0.000391596, 0.000344269, 0.000276243,
0.000192876, 0.000107366, 3.69642E-05, 2.81666E-06, 1.17543E-05, 3.99023E-05,
3.64768E-05, 3.11442E-06, 1.55433E-05, 1.47016E-06, 2.4563E-06, 1.61643E-06 }, {
0.000170222, 0.000188395, 0.000208611, 0.000230996, 0.000255635, 0.000282569,
0.000311647, 0.000342509, 0.000375439, 0.000406564, 0.000436929, 0.00046325,
0.000483474, 0.000489478, 0.000480398, 0.000450346, 0.00039622, 0.00031812,
0.000222731, 0.000124198, 4.45369E-05, 4.0036E-06, 1.12823E-05, 4.24926E-05,
4.71308E-05, 1.10681E-05, 9.0306E-06, 1.01509E-05, 9.40751E-06, 2.85667E-06,
1.39621E-06 }, { 0.00026128, 0.000288782, 0.000319222, 0.000352708, 0.000389247,
0.000428734, 0.000470701, 0.000514283, 0.000559318, 0.000599878, 0.000636417,
0.000663397, 0.000675221, 0.000665028, 0.000626689, 0.000555438, 0.000451281,
0.000321691, 0.000185852, 7.19631E-05, 1.00438E-05, 1.15507E-05, 5.15127E-05,
6.87297E-05, 2.93414E-05, 4.21895E-06, 2.50859E-05, 3.44585E-06, 3.2927E-06,
3.75558E-06, 1.13373E-06 }, { 0.000278817, 0.000307525, 0.000339046, 0.00037336,
0.000410284, 0.000449439, 0.00048997, 0.000530489, 0.000569946, 0.000602149,
0.000626018, 0.00063525, 0.000621111, 0.000584886, 0.000515088, 0.000414447,
0.000291305, 0.000163743, 5.97424E-05, 6.21933E-06, 1.19348E-05, 5.0307E-05,
6.54806E-05, 2.99168E-05, 2.16513E-06, 2.3637E-05, 4.43455E-06, 1.15977E-05,
6.77306E-06, 1.96053E-06, 1.017E-06 }, { 0.000350847, 0.000386024, 0.000424272,
0.000465369, 0.000508819, 0.000553778, 0.000598695, 0.000641225, 0.000678946,
0.00070446, 0.000714655, 0.000701764, 0.000652796, 0.000580776, 0.000468243,
0.000329995, 0.000186809, 6.89606E-05, 7.57245E-06, 1.27272E-05, 5.47725E-05,
7.43907E-05, 3.85669E-05, 2.53576E-06, 2.2542E-05, 1.38323E-05, 9.99502E-06,
2.16297E-06, 2.58424E-06, 1.6736E-06, 1.00801E-06 }, { 0.000439322, 0.000481869,
0.000527529, 0.000575729, 0.000625452, 0.000675109, 0.0007221, 0.000762708,
0.000792501, 0.000803169, 0.000788974, 0.000741576, 0.000644601, 0.000529549,
0.000374959, 0.000213809, 8.05356E-05, 9.38263E-06, 1.31415E-05, 6.01421E-05,
8.40799E-05, 4.74014E-05, 3.96403E-06, 2.06879E-05, 2.31756E-05, 4.40165E-06,
8.65098E-06, 7.99684E-06, 2.7636E-06, 1.74054E-06, 5.96124E-07 }, { 0.000579595,
0.000633301, 0.000689949, 0.000748337, 0.000806534, 0.000861684, 0.000909471,
0.00094404, 0.000957916, 0.000942225, 0.000887997, 0.000788111, 0.000625051,
0.000458752, 0.000266489, 0.000104401, 1.43306E-05, 1.33883E-05, 6.77392E-05,
0.000100259, 6.23288E-05, 7.45552E-06, 1.9553E-05, 3.326E-05, 3.28812E-06,
1.75975E-05, 6.43123E-06, 3.96537E-06, 3.76479E-06, 1.92398E-06, 8.01882E-07 }, {
0.000676931, 0.00073589, 0.000796523, 0.000856783, 0.000913604, 0.000962654,
0.000997846, 0.001011423, 0.000993219, 0.000936927, 0.000832287, 0.000678913,
0.00046727, 0.000283868, 0.000112044, 1.53985E-05, 1.30502E-05, 7.04541E-05,
0.000106308, 6.73555E-05, 8.77598E-06, 1.82048E-05, 3.74996E-05, 4.4234E-06,
1.90089E-05, 2.90753E-06, 5.23104E-06, 3.03976E-06, 2.97221E-06, 1.52179E-06,
6.43493E-07 }, { 0.000826663, 0.00089303, 0.00095889, 0.001020881, 0.00107423,
0.0011125, 0.001127264, 0.001108624, 0.001043771, 0.000932109, 0.000764601,
0.000553855, 0.000306906, 0.00013337, 2.03007E-05, 1.2223E-05, 7.44385E-05,
0.000118395, 7.97224E-05, 1.20276E-05, 1.67998E-05, 4.33205E-05, 7.96594E-06,
1.79679E-05, 5.74969E-06, 1.19621E-05, 7.55659E-06, 4.34798E-06, 2.33511E-06,
1.01997E-06, 5.12298E-07 }, { 0.001002278, 0.001074201, 0.001141794, 0.001199858,
0.001241384, 0.001257406, 0.001237168, 0.001169501, 0.001040874, 0.000862203,
0.000631052, 0.000380782, 0.000143146, 2.73126E-05, 1.08638E-05, 7.77022E-05,
0.000131423, 9.46077E-05, 1.70386E-05, 1.55819E-05, 4.89377E-05, 1.29201E-05,
1.58893E-05, 1.20556E-05, 1.28067E-05, 4.23301E-06, 3.44085E-06, 3.74091E-06,
1.88421E-06, 1.31146E-06, 4.59768E-07 }, { 0.001191727, 0.001264588, 0.001327082,
0.001371695, 0.001388855, 0.001367167, 0.001294567, 0.001161166, 0.00095722,
0.000712639, 0.000438585, 0.000192866, 2.85582E-05, 8.9568E-06, 7.87033E-05,
0.000143697, 0.000111632, 2.39762E-05, 1.33265E-05, 5.38542E-05, 1.93662E-05,
1.32531E-05, 1.88504E-05, 9.91397E-06, 4.16448E-06, 5.6192E-06, 3.44091E-06,
2.93517E-06, 1.72366E-06, 8.06765E-07, 4.22771E-07 }, { 0.001399907, 0.001467035,
0.001514901, 0.001533296, 0.00151015, 0.001432557, 0.001289651, 0.00107753,
0.000797522, 0.000505831, 0.000232103, 5.0585E-05, 9.90065E-06, 7.7725E-05,
0.000155655, 0.000131528, 3.41682E-05, 1.05738E-05, 5.79579E-05, 2.71134E-05,

```
1.02776E-05, 2.55516E-05, 6.7845E-06, 8.95927E-06, 1.03506E-05, 6.91298E-06,  
3.86954E-06, 2.3739E-06, 1.90316E-06, 9.00755E-07, 3.81416E-07 }, { 0.001621115,  
0.001672345, 0.001692018, 0.001667291, 0.001584562, 0.0014319, 0.001204434,  
0.000911508, 0.00057483, 0.000279248, 6.86668E-05, 5.35018E-06, 8.45902E-05,  
0.000166622, 0.000154502, 4.82699E-05, 7.68191E-06, 6.05891E-05, 3.75935E-05,  
7.71715E-06, 3.17376E-05, 4.82833E-06, 1.50193E-05, 1.07253E-05, 5.51821E-06,  
4.65248E-06, 3.45345E-06, 2.01298E-06, 1.13142E-06, 7.90199E-07, 3.10272E-07 }, {  
0.001836335, 0.001857216, 0.001830993, 0.001743232, 0.001581399, 0.001339367,  
0.001025637, 0.000671415, 0.000326041, 9.18845E-05, 5.045E-06, 6.81615E-05,  
0.000181428, 0.000179311, 6.73659E-05, 5.5583E-06, 6.04868E-05, 5.03242E-05,  
5.53718E-06, 3.64428E-05, 4.93769E-06, 2.01571E-05, 8.10237E-06, 4.19436E-06,  
4.01866E-06, 3.62386E-06, 2.94054E-06, 2.19164E-06, 1.46628E-06, 6.92875E-07,  
2.76414E-07 }, { 0.002035371, 0.00200767, 0.001914915, 0.001743651, 0.001487206,  
0.001152589, 0.000770527, 0.000400305, 0.000116395, 6.91779E-06, 5.99007E-05,  
0.000179521, 0.00019966, 9.24919E-05, 5.59582E-06, 5.71479E-05, 6.5203E-05,  
5.35771E-06, 3.89749E-05, 7.95692E-06, 2.29976E-05, 5.468E-06, 6.91761E-06,  
7.37639E-06, 4.94562E-06, 3.42266E-06, 2.75531E-06, 1.81958E-06, 1.03671E-06,  
6.67389E-07, 2.41295E-07 } };
```

```
matriz1000 = new double[31, 31]  
{ { 1.41021E-06, 1.56912E-06, 1.74924E-06, 1.95354E-06, 2.18548E-06,  
2.44943E-06, 2.74979E-06, 3.09161E-06, 3.49285E-06, 3.92438E-06, 4.42868E-06,  
5.00353E-06, 5.72653E-06, 6.40237E-06, 7.24398E-06, 8.19734E-06, 9.27502E-06,  
1.04969E-05, 1.18767E-05, 1.344E-05, 1.51996E-05, 1.72316E-05, 1.95666E-05,  
2.22939E-05, 2.55627E-05, 2.95315E-05, 3.45279E-05, 4.10461E-05, 4.96556E-05,  
6.11106E-05, 7.1026E-05 }, { 2.03847E-06, 2.26811E-06, 2.5284E-06, 2.8236E-06,  
3.15869E-06, 3.53997E-06, 3.97375E-06, 4.46728E-06, 5.04642E-06, 5.669E-06,  
6.39618E-06, 7.22451E-06, 8.26531E-06, 9.23708E-06, 1.04454E-05, 1.18116E-05,  
1.33521E-05, 1.50931E-05, 1.70509E-05, 1.92573E-05, 2.17234E-05, 2.45455E-05,  
2.77489E-05, 3.14286E-05, 3.57364E-05, 4.07879E-05, 4.68069E-05, 5.39302E-05,  
6.15781E-05, 6.64286E-05, 4.80528E-05 }, { 2.74373E-06, 3.05272E-06, 3.40291E-06,  
3.80003E-06, 4.25072E-06, 4.76342E-06, 5.34656E-06, 6.00978E-06, 6.78768E-06,  
7.62343E-06, 8.59887E-06, 9.7089E-06, 1.11018E-05, 1.24002E-05, 1.40114E-05,  
1.58281E-05, 1.78695E-05, 2.01662E-05, 2.27337E-05, 2.56051E-05, 2.87828E-05,  
3.2371E-05, 3.63711E-05, 4.08522E-05, 4.59099E-05, 5.15141E-05, 5.75734E-05,  
6.34311E-05, 6.65642E-05, 5.87083E-05, 1.76335E-05 }, { 3.56377E-06, 3.96495E-06,  
4.41955E-06, 4.93498E-06, 5.51981E-06, 6.18492E-06, 6.94113E-06, 7.80077E-06,  
8.80844E-06, 9.8902E-06, 1.11515E-05, 1.2585E-05, 1.43807E-05, 1.60508E-05,  
1.81179E-05, 2.04402E-05, 2.30375E-05, 2.5942E-05, 2.91632E-05, 3.27285E-05,  
3.662E-05, 4.09331E-05, 4.56184E-05, 5.06754E-05, 5.60663E-05, 6.14903E-05,  
6.63099E-05, 6.87012E-05, 6.39782E-05, 4.14715E-05, 1.30593E-06 }, { 4.65873E-06,  
5.18289E-06, 5.77675E-06, 6.4499E-06, 7.21348E-06, 8.08154E-06, 9.06801E-06,  
1.01887E-05, 1.15014E-05, 1.29092E-05, 1.45485E-05, 1.64085E-05, 1.8733E-05,  
2.0889E-05, 2.35479E-05, 2.65211E-05, 2.98259E-05, 3.34918E-05, 3.75144E-05,  
4.19039E-05, 4.66044E-05, 5.16782E-05, 5.69839E-05, 6.23889E-05, 6.76174E-05,  
7.19408E-05, 7.39328E-05, 7.03805E-05, 5.51655E-05, 2.14434E-05, 2.96595E-06 }, {  
5.98493E-06, 6.65786E-06, 7.42008E-06, 8.28385E-06, 9.26326E-06, 1.03762E-05,  
1.16402E-05, 1.30751E-05, 1.4754E-05, 1.65523E-05, 1.86431E-05, 2.10102E-05,  
2.39599E-05, 2.6686E-05, 3.0033E-05, 3.3753E-05, 3.78552E-05, 4.23579E-05,  
4.72298E-05, 5.24458E-05, 5.78864E-05, 6.35418E-05, 6.91251E-05, 7.4293E-05,  
7.84129E-05, 8.01978E-05, 7.73313E-05, 6.55667E-05, 3.98608E-05, 5.61522E-06,  
8.40047E-06 }, { 7.60594E-06, 8.46041E-06, 9.428E-06, 1.05241E-05, 1.17664E-05,  
1.31772E-05, 1.47783E-05, 1.65941E-05, 1.87161E-05, 2.09854E-05, 2.36184E-05,  
2.65915E-05, 3.02828E-05, 3.36786E-05, 3.78245E-05, 4.23966E-05, 4.73868E-05,  
5.27893E-05, 5.8526E-05, 6.45099E-05, 7.05232E-05, 7.64291E-05, 8.1731E-05,  
8.57872E-05, 8.75004E-05, 8.4984E-05, 7.52261E-05, 5.41627E-05, 2.18817E-05,  
2.37149E-07, 5.16272E-06 }, { 9.81409E-06, 1.09155E-05, 1.21622E-05, 1.35739E-05,  
1.5173E-05, 1.69876E-05, 1.9045E-05, 2.13756E-05, 2.40949E-05, 2.6997E-05,  
3.03556E-05, 3.41353E-05, 3.8806E-05, 4.30779E-05, 4.82551E-05, 5.39071E-05,  
5.99925E-05, 6.64597E-05, 7.31518E-05, 7.98769E-05, 8.62638E-05, 9.19706E-05,  
9.62036E-05, 9.79227E-05, 9.54847E-05, 8.65178E-05, 6.79717E-05, 3.8229E-05,  
6.91952E-06, 4.98982E-06, 2.82148E-07 }, { 1.27764E-05, 1.42083E-05, 1.58286E-05,
```

1.76621E-05, 1.97375E-05, 2.20905E-05, 2.47552E-05, 2.77692E-05, 3.12789E-05,
3.5015E-05, 3.93248E-05, 4.41542E-05, 5.00865E-05, 5.54719E-05, 6.19369E-05,
6.89017E-05, 7.62659E-05, 8.38964E-05, 9.15084E-05, 9.87417E-05, 0.000104999,
0.000109632, 0.000111466, 0.000109008, 0.000100138, 8.25175E-05, 5.45436E-05,
2.03067E-05, 3.32041E-07, 1.15734E-05, 3.29539E-06 }, { 1.65828E-05, 1.84383E-05,
2.05366E-05, 2.29095E-05, 2.55928E-05, 2.86316E-05, 3.20679E-05, 3.5947E-05,
4.04527E-05, 4.52337E-05, 5.07259E-05, 5.68464E-05, 6.43068E-05, 7.10134E-05,
7.8964E-05, 8.73774E-05, 9.60538E-05, 0.000104724, 0.000112907, 0.00011999,
0.000125072, 0.000127092, 0.000124482, 0.000115396, 9.76715E-05, 7.02214E-05,
3.53069E-05, 5.43769E-06, 4.42697E-06, 9.90498E-06, 3.70234E-06 }, { 2.16066E-05,
2.40193E-05, 2.67456E-05, 2.98258E-05, 3.33051E-05, 3.72394E-05, 4.16797E-05,
4.66799E-05, 5.24689E-05, 5.8586E-05, 6.55753E-05, 7.33084E-05, 8.26385E-05,
9.0917E-05, 0.000100565, 0.000110524, 0.000120431, 0.000129798, 0.000137853,
0.000143627, 0.000145841, 0.000142957, 0.000133154, 0.000114659, 8.62889E-05,
5.02265E-05, 1.51651E-05, 4.39163E-07, 1.3784E-05, 1.99175E-06, 3.8218E-07 }, {
2.79464E-05, 3.10588E-05, 3.45726E-05, 3.85379E-05, 4.30104E-05, 4.80582E-05,
5.37414E-05, 6.01212E-05, 6.74768E-05, 7.52078E-05, 8.39796E-05, 9.35941E-05,
0.000105039, 0.000115018, 0.000126377, 0.000137695, 0.000148359, 0.000157561,
0.000164134, 0.000166672, 0.000163536, 0.000152813, 0.000132928, 0.000103164,
6.50696E-05, 2.63105E-05, 2.04186E-06, 7.61287E-06, 1.60142E-05, 1.94619E-06,
3.29865E-06 }, { 3.71594E-05, 4.12837E-05, 4.59346E-05, 5.11752E-05, 5.70748E-05,
6.37169E-05, 7.11716E-05, 7.95056E-05, 8.90618E-05, 9.90348E-05, 0.000110246,
0.000122379, 0.000136557, 0.000148617, 0.000161884, 0.000174401, 0.000185161,
0.000192868, 0.000195826, 0.000192126, 0.000179923, 0.000157313, 0.000124041,
8.21857E-05, 3.83629E-05, 6.72382E-06, 2.9238E-06, 1.9531E-05, 6.78854E-06,
7.67663E-06, 5.50417E-07 }, { 4.82709E-05, 5.36052E-05, 5.96113E-05, 6.6366E-05,
7.39514E-05, 8.24643E-05, 9.19794E-05, 0.00010256, 0.000114605, 0.000127058,
0.000140883, 0.000155588, 0.000172333, 0.000186075, 0.00020042, 0.00021277,
0.000221586, 0.000224985, 0.000220767, 0.000206754, 0.000181556, 0.000144387,
9.81992E-05, 4.99878E-05, 1.23449E-05, 9.66855E-07, 1.64668E-05, 2.00677E-05,
7.44081E-07, 2.57316E-06, 2.49431E-06 }, { 6.28834E-05, 6.97934E-05, 7.75585E-05,
8.62696E-05, 9.60205E-05, 0.000106919, 0.000119034, 0.00013241, 0.000147493,
0.000162891, 0.000179694, 0.000197139, 0.000216271, 0.000231125, 0.000245316,
0.000255457, 0.000259347, 0.000254482, 0.000238412, 0.000209354, 0.000167682,
0.000115611, 6.15451E-05, 1.83218E-05, 7.72168E-07, 1.31091E-05, 2.727E-05,
6.68847E-06, 8.6517E-06, 2.74922E-06, 8.99362E-07 }, { 8.18181E-05, 9.07439E-05,
0.000100749, 0.000111936, 0.000124407, 0.000138269, 0.00015357, 0.000170306,
0.000188936, 0.000207627, 0.000227543, 0.000247505, 0.000268169, 0.000282778,
0.000294434, 0.000298914, 0.000293345, 0.000274868, 0.000241587, 0.000193563,
0.00013508, 7.36885E-05, 2.41388E-05, 1.40635E-06, 1.08194E-05, 3.03083E-05,
1.89156E-05, 1.66376E-06, 7.76216E-06, 4.08204E-06, 9.62903E-07 }, { 0.00010514,
0.000116503, 0.000129198, 0.000143334, 0.000159004, 0.000176299, 0.00019521,
0.000215633, 0.000237969, 0.000259841, 0.00028235, 0.000303724, 0.000323787,
0.000335485, 0.000340569, 0.000334226, 0.000313279, 0.000275416, 0.000220995,
0.000154301, 8.58922E-05, 2.95709E-05, 2.25321E-06, 9.40386E-06, 3.19223E-05,
2.91813E-05, 2.49139E-06, 1.24349E-05, 1.1762E-06, 1.96515E-06, 1.29312E-06 }, {
0.000136171, 0.00015071, 0.000166884, 0.000184794, 0.000204505, 0.000226054,
0.000249317, 0.000274008, 0.000300353, 0.000325253, 0.000349546, 0.000370603,
0.000386783, 0.000391586, 0.000384322, 0.00036028, 0.000316978, 0.000254497,
0.000178185, 9.93575E-05, 3.56289E-05, 3.20267E-06, 9.0264E-06, 3.39949E-05,
3.77046E-05, 8.85406E-06, 7.2249E-06, 8.12044E-06, 7.52604E-06, 2.28522E-06,
1.11696E-06 }, { 0.000209014, 0.000231018, 0.000255372, 0.000282162, 0.000311395,
0.000342986, 0.000376561, 0.000411429, 0.000447458, 0.000479907, 0.000509139,
0.000530724, 0.000540183, 0.000532028, 0.000501357, 0.000444354, 0.000361027,
0.000257353, 0.000148681, 5.75693E-05, 8.03456E-06, 9.24129E-06, 4.12115E-05,
5.49841E-05, 2.34723E-05, 3.37548E-06, 2.00688E-05, 2.75688E-06, 2.63408E-06,
3.00453E-06, 9.06989E-07 }, { 0.000223045, 0.000246013, 0.000271232, 0.000298685,
0.000328226, 0.000359552, 0.000391979, 0.000424396, 0.000455963, 0.000481726,
0.000500822, 0.000508207, 0.000496896, 0.000467916, 0.000412075, 0.00033156,
0.000233045, 0.000130993, 4.77924E-05, 4.97494E-06, 9.54887E-06, 4.02473E-05,
5.23849E-05, 2.39324E-05, 1.73228E-06, 1.89102E-05, 3.54724E-06, 9.27806E-06,
5.41841E-06, 1.56855E-06, 8.13637E-07 }, { 0.000280669, 0.000308813, 0.000339414,


```

0.000372295, 0.000407057, 0.000443026, 0.000478962, 0.000512988, 0.000543166,
0.000563579, 0.000571735, 0.000561422, 0.000522246, 0.000464628, 0.000374598,
0.000263997, 0.000149446, 5.51663E-05, 6.05715E-06, 1.01832E-05, 4.38204E-05,
5.95134E-05, 3.08522E-05, 2.02846E-06, 1.80348E-05, 1.1065E-05, 7.99681E-06,
1.73055E-06, 2.06768E-06, 1.33891E-06, 8.06388E-07 }, { 0.000351449, 0.000385491,
0.000422022, 0.000460586, 0.000500368, 0.000540096, 0.000577692, 0.00061018,
0.000634016, 0.000642552, 0.000631195, 0.000593274, 0.00051569, 0.000423645,
0.000299969, 0.000171045, 6.44254E-05, 7.50486E-06, 1.05151E-05, 4.81172E-05,
6.72652E-05, 3.79193E-05, 3.17064E-06, 1.65522E-05, 1.85395E-05, 3.52217E-06,
6.91995E-06, 6.39717E-06, 2.21064E-06, 1.39245E-06, 4.7692E-07 }, { 0.000463669,
0.000506639, 0.000551963, 0.000598679, 0.000645241, 0.000689365, 0.000727598,
0.000755255, 0.000766357, 0.000753803, 0.000710418, 0.000630504, 0.000500049,
0.000367004, 0.000213188, 8.35163E-05, 1.14624E-05, 1.07133E-05, 5.41966E-05,
8.02094E-05, 4.98605E-05, 5.96304E-06, 1.56452E-05, 2.66073E-05, 2.63055E-06,
1.40778E-05, 5.14604E-06, 3.17291E-06, 3.01171E-06, 1.53923E-06, 6.41461E-07 }, {
0.000541544, 0.000588718, 0.000637231, 0.000685446, 0.000730907, 0.000770152,
0.000798308, 0.000809171, 0.000794606, 0.000749568, 0.00066585, 0.000543143,
0.000373818, 0.000227091, 8.96287E-05, 1.23158E-05, 1.04436E-05, 5.63703E-05,
8.50494E-05, 5.38811E-05, 7.0186E-06, 1.45675E-05, 2.99994E-05, 3.53775E-06,
1.52083E-05, 2.32632E-06, 4.18385E-06, 2.43147E-06, 2.37778E-06, 1.21733E-06,
5.14805E-07 }, { 0.00066134, 0.000714443, 0.000767139, 0.000816739, 0.000859424,
0.000890044, 0.000901857, 0.000886943, 0.000835055, 0.000745716, 0.000611698,
0.000443089, 0.000245519, 0.000106687, 1.62359E-05, 9.78282E-06, 5.95607E-05,
9.47214E-05, 6.37736E-05, 9.61854E-06, 1.34448E-05, 3.4657E-05, 6.37029E-06,
1.43773E-05, 4.59811E-06, 9.56924E-06, 6.04579E-06, 3.47793E-06, 1.86811E-06,
8.15912E-07, 4.09838E-07 }, { 0.00080185, 0.0008594, 0.000913484, 0.000959944,
0.000993169, 0.001005989, 0.000989796, 0.000935655, 0.000832741, 0.000689788,
0.00050485, 0.000304619, 0.000114504, 2.18428E-05, 8.69635E-06, 6.21757E-05,
0.000105146, 7.56807E-05, 1.36251E-05, 1.2472E-05, 3.91523E-05, 1.03317E-05,
1.27162E-05, 9.64109E-06, 1.02474E-05, 3.38802E-06, 2.75364E-06, 2.99273E-06,
1.50738E-06, 1.0494E-06, 3.67927E-07 }, { 0.000953437, 0.001011739, 0.001061745,
0.001097442, 0.001111173, 0.00109382, 0.001035729, 0.000928992, 0.000765812,
0.000570124, 0.00035086, 0.000154275, 2.28367E-05, 7.17139E-06, 6.29816E-05,
0.00011497, 8.92996E-05, 1.91718E-05, 1.0669E-05, 4.3088E-05, 1.5486E-05,
1.0609E-05, 1.50759E-05, 7.93553E-06, 3.33051E-06, 4.49354E-06, 2.75226E-06,
2.34821E-06, 1.37933E-06, 6.45391E-07, 3.38074E-07 }, { 0.001120021, 0.001173738,
0.001212041, 0.00122676, 0.001208239, 0.001146151, 0.001031804, 0.000862076,
0.000638036, 0.000404656, 0.000185658, 4.04509E-05, 7.92882E-06, 6.22055E-05,
0.000124544, 0.000105216, 2.73204E-05, 8.46789E-06, 4.63751E-05, 2.16806E-05,
8.22991E-06, 2.04371E-05, 5.43264E-06, 7.16315E-06, 8.27968E-06, 5.53042E-06,
3.09496E-06, 1.89909E-06, 1.52263E-06, 7.20351E-07, 3.05187E-07 }, { 0.001297044,
0.001338041, 0.001353784, 0.001333997, 0.001267796, 0.001145636, 0.000963622,
0.000729236, 0.000459853, 0.000223367, 5.49078E-05, 4.28453E-06, 6.77071E-05,
0.000133328, 0.000123596, 3.85949E-05, 6.15409E-06, 4.84866E-05, 3.00611E-05,
6.18186E-06, 2.5388E-05, 3.86617E-06, 1.20102E-05, 8.58428E-06, 4.41797E-06,
3.72372E-06, 2.76255E-06, 1.6108E-06, 9.04761E-07, 6.32124E-07, 2.48203E-07 }, {
0.001469293, 0.001486004, 0.001465018, 0.001394786, 0.001265279, 0.001071599,
0.000820555, 0.000537122, 0.00026079, 7.347E-05, 4.03666E-06, 5.45718E-05,
0.000145184, 0.000143448, 5.38627E-05, 4.45232E-06, 4.84142E-05, 4.02429E-05,
4.43556E-06, 2.91571E-05, 3.94887E-06, 1.61232E-05, 6.4895E-06, 3.35604E-06,
3.21461E-06, 2.9003E-06, 2.35283E-06, 1.75437E-06, 1.17312E-06, 5.54579E-07,
2.21337E-07 }, { 0.00162861, 0.00160644, 0.001532205, 0.001395141, 0.001189914,
0.000922139, 0.000616413, 0.000320189, 9.30621E-05, 5.52763E-06, 4.79731E-05,
0.000143682, 0.00015973, 7.3953E-05, 4.47474E-06, 4.57555E-05, 5.2146E-05,
4.28509E-06, 3.11924E-05, 6.3561E-06, 1.84031E-05, 4.38079E-06, 5.52786E-06,
5.8966E-06, 3.95434E-06, 2.73868E-06, 2.20542E-06, 1.45419E-06, 8.30162E-07,
5.33711E-07, 1.92865E-07 } };
}
}

```

Anexo 2: Vocabulario de programación

Tipo (Type): forma de la representación concreta de un concepto, es decir, es un atributo que especifica el tipo de dato que el objeto puede contener.

Clase (Class): el propósito de la clase es ofrecer al programador una herramienta para la creación de nuevos tipos, que puedan utilizarse de una forma tan cómoda como los primitivos (ejemplo de tipo primitivo, int (tipo entero)). Por lo tanto, una clase es un tipo definido por el usuario, más concretamente se diseña un nuevo tipo para suministrar una definición de un concepto que no tiene contrapartida directa dentro de los tipos primitivos. La clase incluye también funciones asociadas al tipo por lo que amplía las posibilidades de las estructuras (struct) que solo permiten construir tipos que agrupan variables.

Campo (Field): variable de cualquier tipo que se declara directamente en una clase o struct. Los campos son miembros de su tipo contenedor.

Propiedad (Property): miembro que ofrece un mecanismo flexible para leer, escribir o calcular el valor de un campo privado.

Método (Method): bloque de código que contiene una serie de instrucciones.

Constructor: cada vez que se crea una clase o struct, se llama a su constructor. Los constructores permiten al programador establecer valores predeterminados, limitar la creación de instancias y escribir código flexible y fácil de leer.

Evento (Event): tipo especial de delegado multidifusión que sólo se puede invocar desde dentro de la clase o estructura dónde se declaran. Los eventos se utilizan mucho en el diseño de interfaz gráfica de una aplicación, ya que permiten obtener datos relativos a las acciones efectuadas por el usuario de la aplicación.

Struct: tipo de valor que se suele utilizar para encapsular pequeños grupos de variables relacionadas.

Instancia (Instance): variable de tipo referencia. Difiere de las variables clásicas por el hecho de que la instancia no contiene directamente datos, sino una referencia de la ubicación en la memoria de la máquina dónde se encuentran los datos.

Anexo 3: Complemento sobre la teoría de la difracción

En este anexo se amplía la teoría expuesta en la sección 2 y se exponen algunos de los casos típicos en los que se manifiesta el fenómeno de la difracción.

La interferencia de dos fuentes de ondas luminosas fue demostrada por Thomas Young mediante el conocido experimento de la doble rendija. Como se puede observar en la figura 29, la luz de la fuente luminosa incide sobre una primera barrera la cual sólo posee una estrecha rendija S_0 . Las ondas que salen de esta primera rendija llegan a una segunda barrera, la cual está provista de dos rendijas S_1 y S_2 . Estas dos rendijas actúan como un par de fuentes luminosas, por lo tanto, la luz de S_1 y S_2 genera, sobre la pantalla de visionado, un patrón visible de bandas paralelas denominadas franjas. Para que el experimento se produzca adecuadamente las rendijas han de tener un tamaño igual o inferior al de la longitud de onda de la fuente luminosa utilizada.

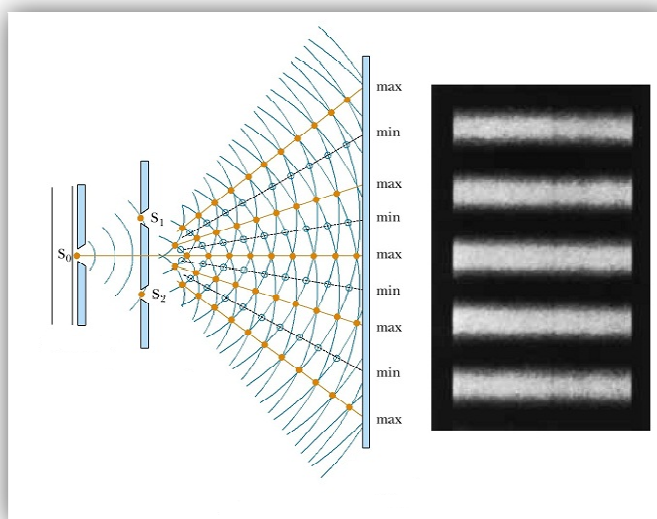


Figura 29. Experimento de la doble rendija de Young

Cuando la interferencia de las dos fuentes luminosas S_1 y S_2 es constructiva, se obtiene un máximo, que da lugar a una línea brillante en la pantalla, y cuando la interferencia es destructiva, se obtiene un mínimo, produciendo una franja oscura tal y como se muestra en la figura 29.

Este experimento demostró que cuando las ondas luminosas atraviesan una pequeña abertura se observa un patrón de interferencia, en vez de un punto concreto de luz, lo cual indica que la luz no viaja en línea recta cuando pasa a través de rendijas estrechas, sino que se dispersa, de ahí, el patrón de interferencia obtenido con dicho experimento. Este fenómeno se le conoce como difracción, el cual, se explica a través del modelo ondulatorio de la luz.

La teoría ondulatoria de la luz fue propuesta por Huygens pero no tuvo aceptación inmediata, ya que se creía que el comportamiento de la luz se explicaba, únicamente, a través de la teoría corpuscular formulada por Newton. Pero gracias al experimento de la doble rendija, se pudo demostrar que la luz posee naturaleza ondulatoria.

Gracias a las ecuaciones formuladas por Maxwell, se pudo entender la luz como una onda electromagnética que se propaga a través del medio de forma indefinida. Las ondas electromagnéticas se propagan indefinidamente ya que, un campo magnético variable con el tiempo genera un campo eléctrico variable, que a su vez crea un campo magnético variable, y así sucesivamente. La siguiente imagen (véase figura 29), muestra lo anteriormente mencionado, además, se observa que dichos campos son perpendiculares a la dirección de propagación y entre sí.

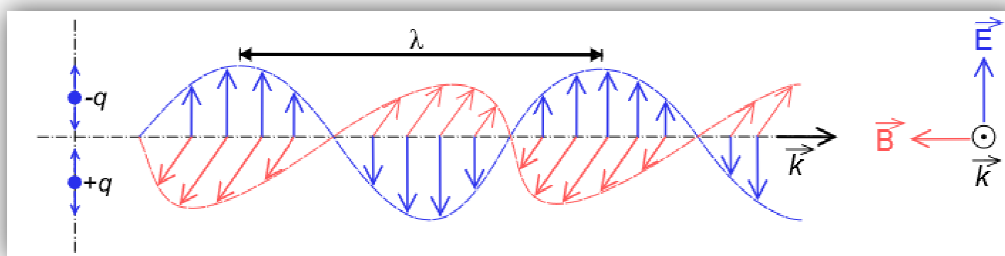


Figura 30. Evolución de una onda electromagnética a lo largo del tiempo

Actualmente, para la física moderna, la luz tiene una naturaleza dual: corpuscular y ondulatoria. Se dice que la luz tiene naturaleza dual ya que en algunos casos actúa como una onda y en otros como una partícula (las entidades que componen la luz se les denominan fotones, los cuales no son considerados diferentes de las partículas materiales ya que, estas mismas también poseen comportamiento dual).

Añadir, que el fenómeno de difracción no solo ocurre cuando la luz atraviesa rendijas estrechas, sino que también tiene lugar cuando pasa alrededor de obstáculos o por bordes afilados.

Otro aspecto importante a comentar, es el patrón de difracción de Fraunhofer de una sola rendija unidimensional (véase figura 30). Para observar dicho patrón se tienen que dar las siguientes condiciones:

- Los haces de luz que pasan a través de la rendija han de ser prácticamente paralelos entre sí, además de monocromáticos.
- Colocar la pantalla muy lejos de la abertura o usar una lente convergente tras la rendija.

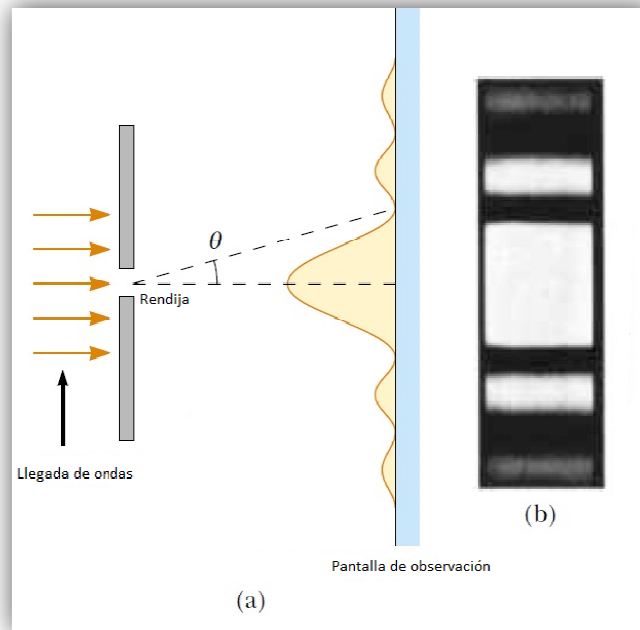


Figura 31. a) Patrón de difracción de Fraunhofer de una única rendija unidimensional. b) Fotografía de un patrón de difracción de Fraunhofer de una sola rendija

El patrón de difracción de Fraunhofer obtenido es la representación de la intensidad luminosa con respecto al seno de θ . Como se puede observar en la figura 31, la mayor parte de la intensidad luminosa se concentra en el punto central de la pantalla de observación, dando lugar al máximo central. Además, existen máximos secundarios de intensidad decreciente, a cada lado del máximo central, entre los cuales (incluyendo al máximo principal) aparecen mínimos de intensidad que corresponden a interferencias destructivas, y de ahí las partes oscuras.

Anexo 4: Ficha de especificaciones de la tarjeta de adquisición



Technical Sales

(866) 531-6285
orders@ni.com

[Requirements and Compatibility](#) | [Ordering Information](#) | [Detailed Specifications](#) | [Pinouts/Front Panel Connections](#)

For user manuals and dimensional drawings, visit the product page resources tab on ni.com.

Last Revised: 2012-10-23 10:17:31.0

NI USB-6501

Low-Cost USB Digital I/O Device



- Small, portable digital I/O device
- 24 digital I/O lines, one 32-bit counter
- Overvoltage protection, 8.5 mA current drive
- Full-speed USB (12 Mbit/s) bus interface
- Built-in screw terminals with removable connectors for easier connectivity
- OEM version available with 34-pin IDC mass termination connectivity

Overview

The NI USB-6501 is a portable digital I/O device that provides reliable data acquisition and control at a low price. With plug-and-play USB connectivity, the USB-6501 is simple enough for home/academic applications but robust and versatile enough for laboratory/industrial applications. The board-only version is ideal for embedded OEM applications.

[Back to Top](#)

Requirements and Compatibility

- •
•
•
•

-

OS Information

Linux® ■ NI-DAQmx ANSI C

Mac OS X LabVIEW

Windows 2000/XP LabWindows/CVI

Windows 7 Measurement Studio

Windows Vista x64/x86 Visual Basic

Driver Information

Software Compatibility

Visual Studio
Visual Studio .NET

[Back to Top](#)

Comparison Tables

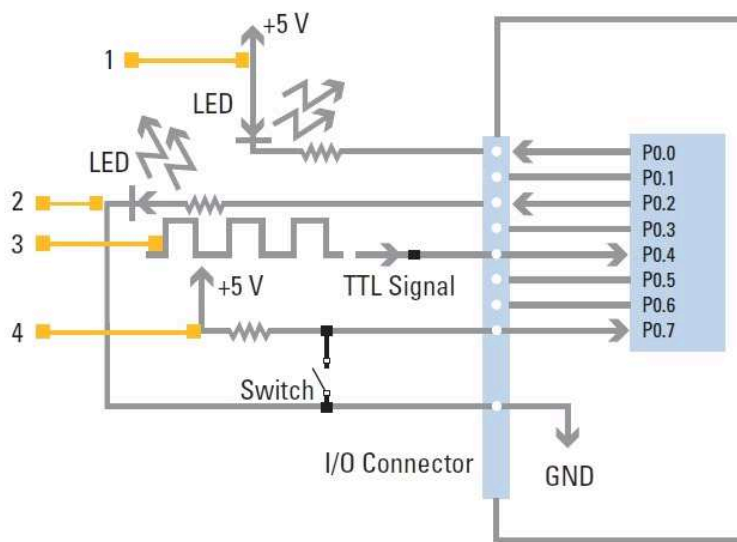
Product	Bus	Digital I/O Lines		Counters	Current Drive	Logic Level	Industrial Feature Set
NI 6501	USB	24	1	8.5 mA	5 V TTL/CMOS	No	

Application and Technology

Hardware

The USB-6501 is a full-speed USB device that provides 24 DIO lines and a 32-bit counter. The 24 digital lines are arranged in three ports: P0.<0..7>, P1.<0..7>, and P2.<0..7>. P2.7 can also function as a 32-bit counter. You can individually program each of the USB-6501 DIO lines as a DI or DO line and use the DIO lines to monitor or control digital signals. All samples of the DI lines and updates of the DO lines are software-timed. The default configuration of the USB-6501 DIO ports is open-drain for 5 V operation with an onboard 4.7 k Ω pull-up resistor. You can add an external, user-provided, pull-up resistor to increase the source current drive up to 8.5 mA per line. Each DIO signal is protected against overvoltage, undervoltage, and overcurrent conditions as well as ESD events. At system startup and reset, the hardware sets all DIO lines to high-impedance inputs.

The data acquisition device does not drive the signal high or low. Each line has a weak pull-up resistor connected to it. You can configure P2.7 as the source for a 32-bit counter. In this mode, the device counts high to low transitions (falling edges) on P2.7. You can arm and disarm the counter and read or reset the counter through software. The USB-6501 supplies a nominal 5 V from two pins, one on each screw-terminal block. You can use the voltage source, which is provided by the USB host, to power external components. The USB-6501 features current-limiting short-circuit protection for safe USB host power use.



1. P0.0 configured as an open-drain digital output driving an LED
2. P0.2 configured as a push-pull digital output driving an LED
3. P0.4 configured as a digital input receiving a TTL signal from a gated inverter
4. P0.7 configured as a digital input receiving a 0 V or 5 V signal from a switch

Figure 1. Example of Connecting a Load

Software

The USB-6501 is shipped with NI-DAQmx measurement services software, a high-performance multithreaded driver for interactive configuration and data acquisition. Use it to develop customized data acquisition applications with NI LabVIEW or C-based development environments. Examples include an interactive control panel to help you get started with your application quickly.

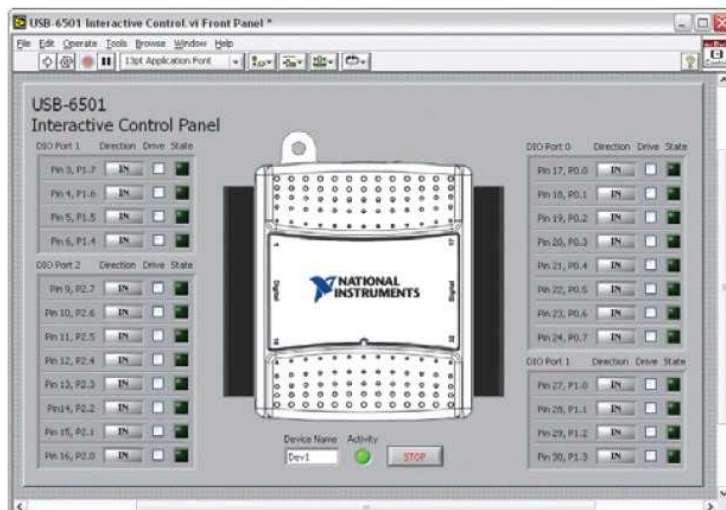


Figure 2. Interactive Control Panel Example Program Shipped with the USB-6501

Recommended Accessories

The USB-6501 has built-in screw terminals for connectivity; no additional accessories are required.

Board-Only Version for OEMs

A board-only version of the USB-6501 (part number 192317-50) is available to OEMs for use in embedded applications. The OEM version offers a 34-pin IDC ribbon cable header for mass termination instead of screw-terminal connectivity. You can mount the OEM version using standoffs (not included in kit) with your own enclosure. For information on special configurations and pricing, visit ni.com/oem.

[Back to Top](#)

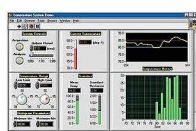
Ordering Information

For a complete list of accessories, visit the product page on ni.com.

Products	Part Number	Recommended Accessories	Part Number
----------	-------------	-------------------------	-------------

[Back to Top](#)

LabVIEW Professional Development System for Windows

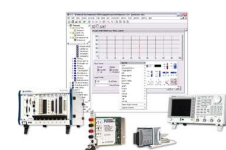


.NET objects The mark LabWindows is used under a
Capability to build DLLs, executables, and license from Microsoft Corporation.
MSI installers

Software Recommendations

- Advanced software tools for large project Real-time advanced 2D graphs and charts development Complete hardware compatibility with IVI,
- Automatic code generation using DAQ VISA, DAQ, GPIB, and serial
- Assistant and Instrument I/O Assistant Analysis tools for array manipulation, signal Tight integration with a wide range of processing statistics, and curve fitting
- hardware Simplified cross-platform communication with
- Advanced measurement analysis and digital network variables
- signal processing Measurement Studio .NET tools (included in
- Open connectivity with DLLs, ActiveX, and LabWindows/CVI Full only)

NI LabWindows™/CVI for Windows



[Back to Top](#)



Support and Services

System Assurance Programs

NI system assurance programs are designed to make it even easier for you to own an NI system. These programs include configuration and deployment services for your NI PXI, CompactRIO, or Compact FieldPoint system. The NI Basic System Assurance Program provides a simple integration test and ensures that your system is delivered completely assembled in one box. When you configure your system with the NI Standard System Assurance Program, you can select from available NI system driver sets and application development environments to create customized, reorderable software configurations. Your system arrives fully assembled and tested in one box with your software preinstalled. When you order your system with the standard program, you also receive system-specific documentation including a bill of materials, an integration test report, a recommended maintenance plan, and frequently asked question documents. Finally, the standard program reduces the total cost of owning an NI system by providing three years of warranty coverage and calibration service. Use the online product advisors at ni.com/advisor to find a system assurance program to meet your needs.

Calibration

NI measurement hardware is calibrated to ensure measurement accuracy and verify that the device meets its published specifications. To ensure the ongoing accuracy of your measurement hardware, NI offers basic or detailed recalibration service that provides ongoing ISO 9001 audit compliance and confidence in your measurements. To learn more about NI calibration services or to locate a qualified service center near you, contact your local sales office or visit ni.com/calibration.

Technical Support

Get answers to your technical questions using the following National Instruments resources.

- **Support** - Visit ni.com/support to access the NI KnowledgeBase, example programs, and tutorials or to contact our applications engineers who are located in NI sales offices around the world and speak the local language.
- **Discussion Forums** - Visit forums.ni.com for a diverse set of discussion boards on topics you care about.
- **Online Community** - Visit community.ni.com to find, contribute, or collaborate on customer-contributed technical content with users like you.

Repair

While you may never need your hardware repaired, NI understands that unexpected events may lead to necessary repairs. NI offers repair services performed by highly trained technicians who quickly return your device with the guarantee that it will perform to factory specifications. For more information, visit ni.com/repair.

Training and Certifications

The NI training and certification program delivers the fastest, most certain route to increased proficiency and productivity using NI software and hardware. Training builds the skills to more efficiently develop robust, maintainable applications, while certification validates your knowledge and ability.

- **Classroom training in cities worldwide** - the most comprehensive hands-on training taught by engineers.
- **On-site training at your facility** - an excellent option to train multiple employees at the same time.
- **Online instructor-led training** - lower-cost, remote training if classroom or on-site courses are not possible.
- **Course kits** - lowest-cost, self-paced training that you can use as reference guides.
- **Training memberships** and training credits - to buy now and schedule training later.

Visit ni.com/training for more information.

Extended Warranty

NI offers options for extending the standard product warranty to meet the life-cycle requirements of your project. In addition, because NI understands that your requirements may change, the extended warranty is flexible in length and easily renewed. For more information, visit ni.com/warranty.

OEM

NI offers design-in consulting and product integration assistance if you need NI products for OEM applications. For information about special pricing and services for OEM customers, visit ni.com/oem.

Alliance

Our Professional Services Team is comprised of NI applications engineers, NI Consulting Services, and a worldwide National Instruments Alliance Partner program of more than 700 independent consultants and integrators. Services range from start-up assistance to turnkey system integration. Visit ni.com/alliance.

[Back to Top](#)

Detailed Specifications

The following specifications are typical at 25 °C, unless otherwise noted. All voltages are relative to COM unless otherwise noted.

Digital I/O

Number of lines

P0.<0..7>	8
P1.<0..7>	8
P2.<0..7>	8

Direction control	Input or output, software-selectable
Output driver type	Active drive (push-pull) or open collector (open-drain), software selectable
Pull-up resistor	4.7 k Ω VBus (nominally 5 V)
Absolute voltage range	– 0.5 to 5.8 V with respect to GND
Power-on state	Input (high impedance)
Digital Logic Levels	
Input low voltage	–0.3 V min, 0.8 V max
Input high voltage	2.0 V min, 5.8 V max
Input leakage current	50.0 μ A max
Output low voltage	
Open collector (open-drain) or active drive (push-pull)	
$I_{OL} = 2$ mA	0.4 V max
$I_{OL} = 8.5$ mA	0.8 V max
Output high voltage	
Active drive (push-pull ¹)	
$I_{OH} = -2$ mA	2.8 V min, 3.6 V max
$I_{OH} = -8.5$ mA	2.0 V min, 3.5 V max
Open collector (open-drain), $I_{OH} = -0.4$ mA, nominal	2.0 V min, 5.0 V max
Open collector (open-drain), $I_{OH} = -7.5$ mA, with external pull-up resistor	2.0 V min
Counter	
Number of counters	1 (P2.7 can be configured as a counter)
Resolution	32 bits
Counter measurements	Falling edge counting
Maximum input frequency	5 MHz
Minimum high pulse width	100 ns
Minimum low pulse width	100 ns
Bus Interface	
USB specification	USB 2.0 Full-Speed (12 Mb/s)
External Voltage	
+5 V output	
Voltage	4.00 V min, 5.25 V max
Current	230 mA max
Power Requirements	
USB	
Input voltage	4.50 to 5.25 VDC, in configured state
Active current	80 mA typical, 500 mA max
Suspend current	500 μ A max, all DIO lines disconnected

Physical Characteristics

Dimensions

Without connectors	6.35 cm x 8.51 cm x 2.31 cm (2.50 in. x 3.35 in. x 0.91 in.)
With connectors	8.18 cm x 8.51 cm x 2.31 cm (3.22 in. x 3.35 in. x 0.91 in.)
I/O connectors	USB series B receptacle, (2) 16 position (screw terminal) plug headers
Screw-terminal wiring	16 to 28 AWG copper conductor wire with 10 mm (0.39 in.) of insulation stripped from the end
Torque for screw terminals	0.22 – 0.25 N · m (2.0 – 2.2 lb · in.)
Weight	84 g (3 oz)

Safety

If you need to clean the module, wipe it with a dry towel.

Overvoltage Protection

Connect only voltages that are within these limits.

Channel-to-COM (one channel) ±30 V max, Measurement Category I

Channels-to-COM (one port, all channels) ±8.9 V max, Measurement Category I

Measurement Category I is for measurements performed on circuits not directly connected to the electrical distribution system referred to as MAINS ² voltage. This category is for measurements of voltages from specially protected secondary circuits. Such voltage measurements include signal levels, special equipment, limited-energy parts of equipment, circuits powered by regulated low-voltage sources, and electronics.

Caution Do not use this module for connection to signals or for measurements within Measurement Categories II, III, or IV.

Safety Standards

This product is designed to meet the requirements of the following standards of safety for electrical equipment for measurement, control, and laboratory use:

- IEC 61010-1, EN 61010-1
- UL 61010-1, CSA 61010-1

Note For UL and other safety certifications, refer to the product label or the *Online Product Certification* section.

Hazardous Locations

This device is not certified for use in hazardous locations.

Environmental

The NI USB-6501 device is intended for indoor use only.

Operating temperature (IEC 60068-2-1 and IEC 60068-2-2)	0 to 55 °C
Operating humidity (IEC 60068-2-56)	5 to 95% RH, noncondensing
Maximum altitude	2,000 m (at 25 °C ambient temperature)
Storage temperature (IEC 60068-2-1 and IEC 60068-2-2)	– 40 to 85 °C
Storage humidity (IEC 60068-2-56)	5 to 90% RH, noncondensing

Pollution Degree (IEC 60664) 2

Electromagnetic Compatibility

This product is designed to meet the requirements of the following standards of EMC for electrical equipment for measurement, control, and laboratory use:

- EN 61326 EMC requirements; Minimum Immunity
- EN 55011 Emissions; Group 1, Class A
- CE, C-Tick, ICES, and FCC Part 15 Emissions; Class A

Note For EMC compliance, operate this device with double-shielded cables.

CE Compliance

This product meets the essential requirements of applicable European Directives, as amended for CE marking, as follows:

- 2006/95 /EC; Low-Voltage Directive (safety)
- 2004/108 /EC; Electromagnetic Compatibility Directive (EMC)

Note For the standards applied to assess the EMC of this product, refer to the *Online Product Certification* section.

Online Product Certification

Refer to the product Declaration of Conformity (DoC) for additional regulatory compliance information. To obtain product certifications and the DoC for this product, visit ni.com/certification, search by module number or product line, and click the appropriate link in the Certification column.

Environmental Management

National Instruments is committed to designing and manufacturing products in an environmentally responsible manner. NI recognizes that eliminating certain hazardous substances from our products is beneficial not only to the environment but also to NI customers.

For additional environmental information, refer to the *NI and the Environment* Web page at ni.com/environment. This page contains the environmental regulations and directives with which NI complies, as well as other environmental information not included in this document.

Waste Electrical and Electronic Equipment (WEEE)



EU Customers At the end of their life cycle, all products *must* be sent to a WEEE recycling center. For more information about WEEE recycling centers and National Instruments WEEE initiatives, visit ni.com/environment/weee.htm.

电子信息产品污染控制管理办法 (中国 RoHS)



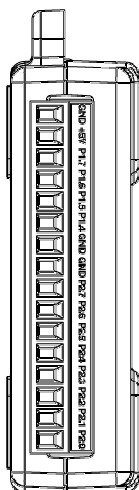
中国客户 National Instruments 符合中国电子信息产品中限制使用某些有害物质指令 (RoHS)。关于 National Instruments 中国 RoHS 合规性信息，请登录 ni.com/environment/rohs_china。(For information about China RoHS compliance, go to ni.com/environment/rohs_china.)

¹ The total current sourced by all DO lines simultaneously should not exceed 65 mA.

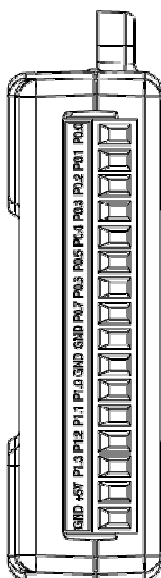
² MAINS is defined as the (hazardous live) electrical supply system to which equipment is designed to be connected for the purpose of powering the equipment. Suitably rated measuring circuits may be connected to the MAINS for measuring purposes.

[Back to Top](#)

Pinouts/Front Panel Connections



NI USB-6501 Digital Terminal Assignments, Left



NI USB-6501 Digital Terminal Assignments, Right

[Back to Top](#)