Prueba de Evaluación Continua 5

Adolfo Hilario Garcia

13 de diciembre de 2020

El firmante de este trabajo reconoce que todo él es original, de su única autoría, escritura y redacción, y que allí donde han sido empleadas ideas o datos de otros autores, su trabajo ha sido reconocido y ubicado, con suficiente detalle, como para que el lector pueda consultar lo afirmado sobre él.

1 Introducción y enunciado

En el campo de la óptica, la difracción de Fresnel, también conocida como difracción de campo cercano, es el patrón de difracción de ondas electromagnéticas obtenido cerca del objeto causante de difracción (a menudo una apertura). En la difracción de Fresnel generada por una función de apertura rectangular, dos funciones trascendentes, las integrales de Fresnel, que se definen como:

$$C(\omega) = \int_0^{\omega} \cos\left(\frac{\pi}{2}x^2\right)\, dx \tag{1}$$

$$S(\omega) = \int_0^{\omega} \sin\left(\frac{\pi}{2}x^2\right)\, dx \tag{2}$$

La complicación de estas integrales reside en que no son expresables como funciones elementales y requieren de la evaluación numérica. En esta PEC se realizarán distintas evaluaciones con distintos métodos y se analizará la cota de error generado por cada uno de ellos en el orden que sigue:

(a) Se utilizará la *regla trapezoidal compuesta* para distintos $n-$intervalos.

(b) Se utilizará la *integración de Romberg* presentándola como una mejora a la regla trapezoidal compuesta.

(c) Además, se empleará el método de *cuadratura gaussiana* para 64 puntos.

(d) Se representará y se discutirá las distribuciones de los puntos x_i utilizados en la cuadratura gaussiana con respecto a la distribución de los puntos x_i utilizando la regla trapezoidal compuesta. Además, se analizará el error absoluto de la regla trapezoidal compuesta y de la cuadratura gaussiana con respecto a la integración de Romberg.

(e) Por último, se calculará la intensidad relativa del patrón de difracción I/I_0 en determinados puntos de la pantalla utilizando la cuadratura gaussiana de 64 puntos.

Para todos los apartados excepto el (e), fijaremos un valor de $\omega = 5$, por lo que todos los cálculos y representaciones gráficas se encuentran en el intervalo $x_i \, \varepsilon \, [0, 5]$ (excepto en la cuadratura gaussiana, que están en $x_i \, \varepsilon \, [-1, 1]$).

2 Metodología

Para realizar esta PEC se han escrito programas que agilicen la tarea a la hora de realizar cálculos y obtener resultados. Se ha utilizado para escribirlos el lenguaje de programación de Matlab en el software 2020b. He decidido cambiarme definitivamente a Matlab por la mayor agilidad que presenta respecto al lenguaje C, que he utilizado en PECs anteriores.

En el Sección 5 se encuentran listados de código que ilustran la metodología de trabajo con la que será enfocada tanto la PEC como la asignatura en general. La metodología sigue las pautas generales que se enlistan a continuación:

- Los algoritmos que efectuan los métodos numéricos están basados, generalmente, en el pseudocódigo de los algoritmos de Burden y Faires (2011). Dependiendo del método se utilizan funciones propiamente dichas, con sus variables auxiliares, para ser llamadas a lo largo de la sesión de trabajo, o simplemente éstas se definen según las vayamos necesitando. Todo depende de la complejidad del algoritmo.
- Al tener que presentar gran parte resultados en forma de tabla, se ha creado una función, llamada `escribe_matriz_LaTeX`, cuya función es crear un archivo `.tex` con los datos que el programador decida.
- A lo largo del código aparecen comentarios e indicaciones que facilitan la lectura. En Sección 5 se presentan las principales piezas de código que se han utilizado para llevar a cabo los diferentes apartados.

3 Resultados

A continuación se exponen los resultados junto con los razonamientos e interpretaciones pertinentes para cada apartado.

3.1 Apartado (a)

La regla trapezoidal compuesta consiste en dividir en n trapecios el área bajo una función $f(x)$ y cálcular el área de cada uno para luego sumarlas. Expresado matemárticamente:

$$\int_a^b f(x)\,dx = \int_{x_0}^{x_1} f(x)\,dx + \int_{x_1}^{x_2} f(x)\,dx + \cdots + \int_{x_{n-1}}^{x_n} f(x)\,dx$$

dónde $x_0 = a$ y $x_n = b$. A partir de la fórmula para el área de un trapecio podemos obtener la siguiente expresión general:

$$\int_a^b f(x)\,dx = \frac{h}{2}\left[f(x_0) + 2\sum_{i=1}^{n-1} f(x_i) + f(x_n)\right]; \quad \text{donde} \quad h = \frac{b-a}{n} \tag{3}$$

Se puede comprobar que en total se realizan $1 + 2(n-1) + 1 = 2n$ evaluaciones de funciones.

En el caso que nos ocupa, el enunciado nos pide la aproximación mediante la regla trapezoidal compuesta para los siguientes valores de n:

$$n = [1, 2, 4, 6, 8, 16, 32, 64, 128, 256]$$

Al aplicar la ecuación (3), para cada valor de n se obtienen los resultados mostrados en la Tabla 1. Como se puede apreciar, los resultados convergen a un valor determinado según aumentamos el número de particiones.

Tabla 1: *Uso de la regla trapezoidal compuesta en* $[0, 5]$ *para distintos* n

n	n_{ev}	$S(5)$	$C(5)$
1	2	2.500000	2.500000
2	4	0.293291	-1.059699
4	8	0.817116	-2.740093
6	12	-0.063985	-0.556502
8	16	-0.047404	-0.852584
16	32	0.528965	0.321135
32	64	0.499397	0.527926
64	128	0.499200	0.555434
128	256	0.499192	0.561621
256	512	0.499191	0.563131

Como hemos dicho, para cada n se evaluarán un total de $2n$ funciones, como muestra la columna de n_{ev}. Es interesante asegurarse gráficamente de que las aproximaciones convergen a un valor. De acuerdo con las figuras 1 y 2, la estimación gana en precisión según aumentamos el número de subintervalos.

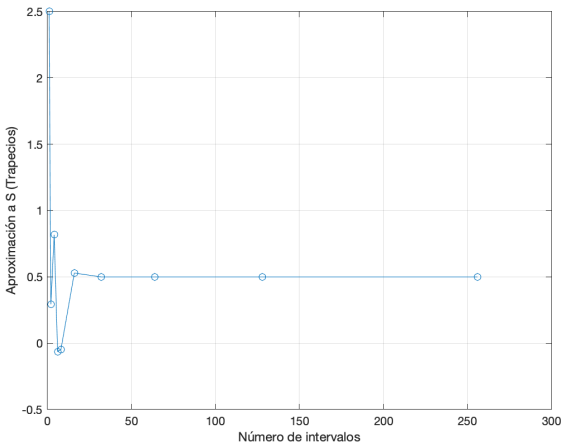

Figura 1: *Representación gráfica del valor de* $S(5)$ *con respecto al número de intervalos*

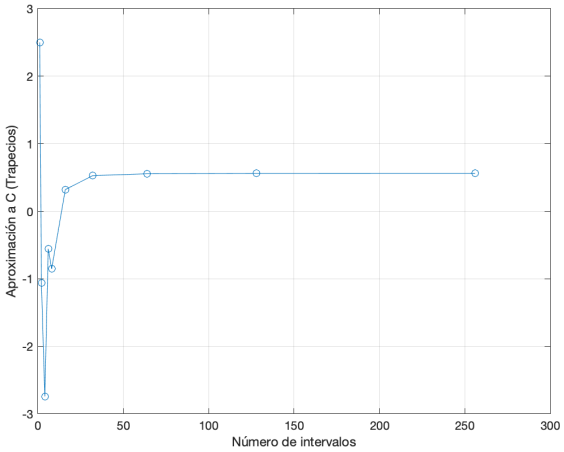

Figura 2: *Representación gráfica del valor de* $C(5)$ *con respecto al número de intervalos*

3.2 Apartado (b)

Para este apartado se han utilizado dos variaciones para la integración de Romberg, que se explican a continuación:

- Primero, se ha estimado el valor de la integral a partir del método expuesto en la tutoría intercampus, en el cual se calcula el valor de la integral mediante la ecuación:

$$I \approx Mejorestimacion = Masexacto + \frac{1}{2^n - 1}(Masexacto - Menosexacto) \qquad (4)$$

  Donde $Menosexacto$ son los valores para cada n obtenidos mediante la regla trapezoidal compuesta en el apartado anterior, y $Masexacto$ corresponde a una nueva regla trapezoidal, ahora haciendo que $n' = 2n$. Podemos utilizar directamente los resultados del apartado anterior, y, como sabemos que la regla trapezoidal compuesta evalúa $2n$ funciones distintas, éste método, por ésta vía, realiza $4n$ evaluaciones.

- Para poder presentar los resultados en forma de tablas de Romberg, he tenido que hacer un algoritmo específico con las siguientes condiciones y fórmulas para cada elemento de la tabla (Burden y Faires (2011) 215):

$$R_{1,1} = \frac{h_1}{2}[f(a) + f(b)] \qquad h_k = \frac{b-a}{2^{k-1}} \qquad (5)$$

$$R_{k,1} = \frac{1}{2}\left[R_{k-1,1} + h_{k-1}\sum_{i=1}^{2^{k-1}} f(a + (2i-1)h_k)\right] \qquad k = 2, \ldots, n \qquad (6)$$

$$R_{kj} = R_{k,j-1} + \frac{1}{4^{j-1} - 1}(R_{k,j-1} - R_{k-1,j-1}) \qquad k = j, j+1, \ldots;\ j = 2, \ldots, n \qquad (7)$$

  En realidad, la primera y segunda ecuación no son más que la ejecución de la regla trapezoidal compuesta para cada $k = 1, \ldots, n$. Además, la ecuación (4) es en cierto modo similar a la tercera ecuación, lo que me hace pensar que:

$$I \approx Mejorestimacion \approx R_{n,n}$$

  Por último constatar que se requiere de $2 + 2^{k-1}$ evaluaciones de funciones.

Los resultados obtenidos mediante el primer procedimiento se muestran en la Tabla 2. Vemos cómo, de nuevo, los valores de las integrales convergen. Ahora bien, también observamos que en la Tabla 1, la precisión de más de dos cifras decimales llega a partir de $n = 32$ para $S(5)$ y a partir $n = 128$ para $C(5)$. Mediante la integración de Romberg, hemos conseguido esa precisión en el n anterior, esto es, $n = 16$ para S y $n = 64$ para C.

Tabla 2: *Uso de la integración de Romberg en* $[0, 5]$ *para distintos* n

n	$S(5)$	$C(5)$
1	-1.913417	-4.619398
2	0.991724	-3.300225
4	-0.105039	-0.726750
6	1.250992	0.041007
8	0.531225	0.325738
16	0.499397	0.527929
32	0.499200	0.555434
64	0.499192	0.561621
128	0.499191	0.563131
256	0.499191	0.563506

Por lo que respecta a las tablas de Romberg, dado que el resultado converge muy rápidamente he decidido presentar las tablas para $n = 16$. Inexplicablemente, a Matlab le cuesta crear una matriz de más de 20×20 para este algoritmo.

Tabla 3: *Uso de la integración de Romberg para* $S(5)$ *en* $[0, 5]$ *y* $n = 16$

2.500000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.293291	-0.442278	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.817116	0.991724	1.087324	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-0.047404	-0.335578	-0.424064	-0.448055	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.528965	0.721088	0.791532	0.810828	0.815764	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.499397	0.489541	0.474105	0.469066	0.467726	0.467386	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.499200	0.499134	0.499774	0.500181	0.500303	0.500335	0.500343	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.499192	0.499189	0.499193	0.499184	0.499180	0.499179	0.499178	0.499178	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.000000	0.000000	0.000000	0.000000	0.000000
0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.000000	0.000000	0.000000	0.000000
0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.000000	0.000000	0.000000
0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.000000	0.000000
0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.000000
0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191	0.499191

Tabla 4: *Uso de la integración de Romberg para* $C(5)$ *en* $[0, 5]$ *y* $n = 16$

2.500000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-1.059699	-2.246265	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-2.740093	-3.300225	-3.370489	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
-0.852584	-0.223414	-0.018293	0.034916	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.321135	0.712375	0.774761	0.787349	0.790300	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.527926	0.596856	0.589155	0.586209	0.585420	0.585220	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.555434	0.564603	0.562453	0.562029	0.561934	0.561911	0.561905	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.561621	0.563684	0.563622	0.563641	0.563647	0.563649	0.563649	0.563649	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.563131	0.563634	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.563506	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.563600	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.000000	0.000000	0.000000	0.000000	0.000000
0.563623	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.000000	0.000000	0.000000	0.000000
0.563629	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.000000	0.000000	0.000000
0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.000000	0.000000
0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.000000
0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631	0.563631

3.3 Apartado (c)

La cuadratura gaussiana consiste en efectuar el siguiente cambio de variable:

$$I = \int_a^b f(x)\,dx = \int_{-1}^{1} f(t)\,dt \qquad x = \frac{(b-a)t + b + a}{2}, \quad dx = \frac{b-a}{2}dt \tag{8}$$

En nuestro caso, como $b = \omega = 5$, tenemos que

$$C(5) = \frac{5}{2}\int_{-1}^{1} \cos\left(\frac{\pi}{2}\cdot\frac{25}{4}(t+1)^2\right)\,dt \tag{9}$$

$$S(5) = \frac{5}{2}\int_{-1}^{1} \sin\left(\frac{\pi}{2}\cdot\frac{25}{4}(t+1)^2\right)\,dt \tag{10}$$

Ahora la integral se puede expresar como la suma de los productos de una función peso w con n valores por $f(t_i)$, donde t_i son las raíces del polinomio de Legendre de grado n. Como el enunciado exige que $n = 64$, tendremos una suma de 64 términos (y, por tanto, 64 evaluaciones de funciones). Expresado matemáticamente:

$$C(5) \approx \sum_{i=1}^{64} w_i\, c(t_i) = 0.563\,631 \tag{11}$$

$$S(5) \approx \sum_{i=1}^{64} w_i\, s(t_i) = 0.499\,191 \tag{12}$$

donde c y s son los integrandos de las funciones de fresnel con el cambio de variable aplicado. Cabe recordar que los valores w_i y t_i son proporcionados en el curso virtual.

3.4 Apartado (d)

A continuación vamos a estudiar como se distribuyen los puntos x_i según el método usado para aproximar las integrales. En el caso que nos ocupa, compararemos los puntos x_i del apartado (a) con los puntos x_i del apartado b.Como se muestra tnato en la Tabla 5, como en las figuras 3 y 4, la distribuciones no coinciden en absoluto. Vemos como, mediante la regla trapezoidal compuesta, los puntos de distribuyen de forma equiespaciada y su valor crece de forma lineal. Por otro lado, los puntos utilizados en la cuadratura gaussiana (utilizados realizando el cambio de variable), se distancian según avanza n. Además, no crecen linealmente, sino que siguen una distribución alternante.

Nótese como en la Figura 3 $x_0 = 0$ y $x_{64} = 5$; en cambio, en la Figura 4, $x_{63} \approx 0$ y $x_{64} \approx 5$.

Por último, vamos a analizar y discutir el error absoluto de éstos dos métodos con respecto de la integración de Romberg. Para ello, utilizaremos los valores obtenidos en la Tabla 2 para $n = 64$ como valores de referencia. Siguiendo estas pautas, tenemos que

Tabla 5: *Valores de x_i para cada $k = 1, \ldots, 64$ mediante los dos métodos pedidos.*

i	$x_{i,T}$	$x_{i,G}$
1	0.078125	2.439124
2	0.156250	2.560876
3	0.234375	2.317517
4	0.312500	2.682483
5	0.390625	2.196343
6	0.468750	2.803657
7	0.546875	2.075889
8	0.625000	2.924111
9	0.703125	1.956441
10	0.781250	3.043559
11	0.859375	1.838282
12	0.937500	3.161718
13	1.015625	1.721693
14	1.093750	3.278307
15	1.171875	1.606950
16	1.250000	3.393050
17	1.328125	1.494325
18	1.406250	3.505675
19	1.484375	1.384085
20	1.562500	3.615915
21	1.640625	1.276492
22	1.718750	3.723508
23	1.796875	1.171801
24	1.875000	3.828199
25	1.953125	1.070261
26	2.031250	3.929739
27	2.109375	0.972112
28	2.187500	4.027888
29	2.265625	0.877586
30	2.343750	4.122414
31	2.421875	0.786909
32	2.500000	4.213091
33	2.578125	0.700295
34	2.656250	4.299705
35	2.734375	0.617950
36	2.812500	4.382050
37	2.890625	0.540069
38	2.968750	4.459931
39	3.046875	0.466837
40	3.125000	4.533163
41	3.203125	0.398427
42	3.281250	4.601573
43	3.359375	0.335002
44	3.437500	4.664998
45	3.515625	0.276711
46	3.593750	4.723289
47	3.671875	0.223695
48	3.750000	4.776305
49	3.828125	0.176077
50	3.906250	4.823923
51	3.984375	0.133972
52	4.062500	4.866028
53	4.140625	0.097478
54	4.218750	4.902522
55	4.296875	0.066683
56	4.375000	4.933317
57	4.453125	0.041659
58	4.531250	4.958341
59	4.609375	0.022467
60	4.687500	4.977533
61	4.765625	0.009150
62	4.843750	4.990850
63	4.921875	0.001737
64	5.000000	4.998263

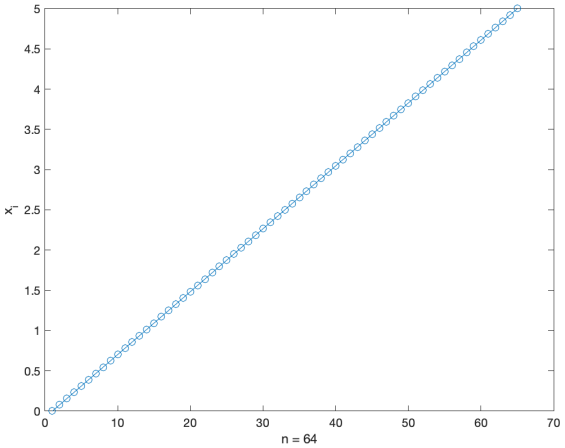

Figura 3: *Representación gráfica de los valores x_i para cada i en la regla trapezoidal*

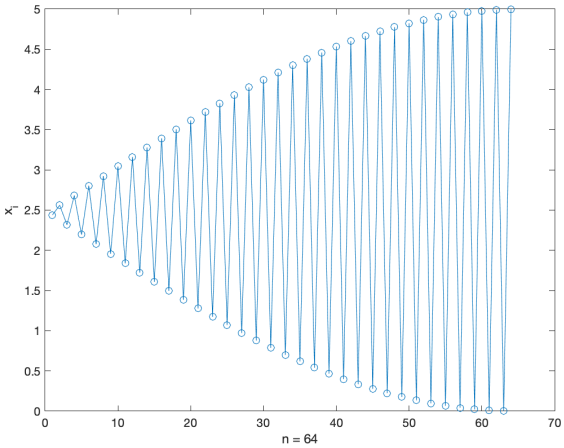

Figura 4: *Representación gráfica de los valores x_i para cada i en la cuadratura gaussiana*

$$\varepsilon_{T,S} = 8.2 \times 10^{-6} \qquad \varepsilon_{T,C} = 0.0062$$
$$\varepsilon_{G,S} = 4.9 \times 10^{-7} \qquad \varepsilon_{G,C} = 0.0020$$

Se puede observar que la aproximación para el seno de Fresnel es cómo mínimo dos órdenes de magnitud más precisa que la del coseno, independientemente del método. Además, podemos decir que la aproximación por cuadratura Gaussiana es más exacta en relación con la integración de Romberg, comparada con la regla trapezoidal compuesta, pues su error absoluto es siempre menor.

3.5 Apartado (e)

Para este ejercicio consideraremos una apertura de $L_x = 4\sqrt{79}/25\,\text{mm}$ por $L_y = 6\sqrt{79}/25\,\text{mm}$ situada a una distancia $q_0 = 400\,\text{mm}$ de la pantalla y con una longitud de onda de apertura de $\lambda = 632\,\text{nm}$. Todos los cálculos se realizarán en milímetros.

El ejercicio nos pide calcular la intensidad relativa I/I_0, que se obtiene mediante la fórmula:

$$\frac{I}{I_0} = \frac{1}{4}\left[(C(u_2) - C(u_1))^2 + (S(u_2) - S(u_1))^2\right]\left[(C(v_2) - C(v_1))^2 + (S(v_2) - S(v_1))^2\right] \quad (13)$$

donde

$$u = x\sqrt{\frac{2}{\lambda q_0}} \qquad v = y\sqrt{\frac{2}{\lambda q_0}} \quad (14)$$

Se puede ver que el término $C(u_2) - C(u_1)$ no es más que la aplicación del teorema fundamental del Cálculo sobre una integral entre u_1 y u_2. Lo mismo sucede con las demás funciones y sus respectivos límites. En el caso que nos ocupa, nos pide utilizar la cuadratura gaussiana y los puntos x_1, x_2, y_1, y_2 que nos proporciona el enunciado. Al realizar el cambio de variable requerido por este método, obtenemos que

$$x = \frac{L_x}{2}\sqrt{\frac{2}{\lambda q_0}}(t+1) \qquad dx = \frac{L_x}{2}\sqrt{\frac{2}{\lambda q_0}}\,dt$$

$$y = \frac{L_y}{2}\sqrt{\frac{2}{\lambda q_0}}(t+1) \qquad dy = \frac{L_y}{2}\sqrt{\frac{2}{\lambda q_0}}\,dt$$

Aplicando estas nuevas variables a las funciones S y C, realizando la cuadratura gaussiana de 64 puntos con las funciones peso y raíces proporcionadas en el curso virtual, y sustituyendo los valores en la ecuación (13), se obtiene que

$$\frac{I}{I_0} = 7.6 \times 10^{-4} \tag{15}$$

3.6 Apartado (f)

Al representar las funciones $S(\omega)$ y $C(\omega)$ en el intervalo $[0, 5]$, se obtiene el patrón de la Figura 5.

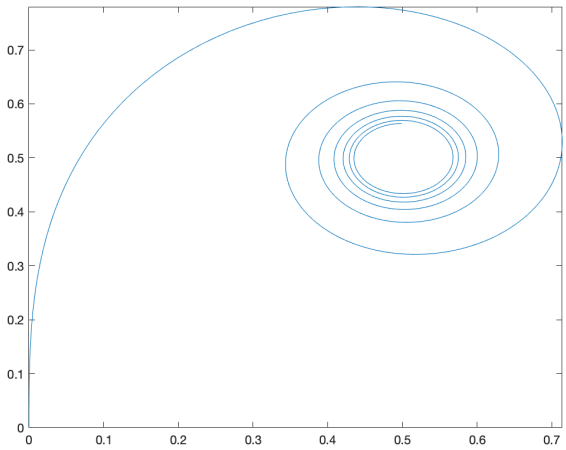

Figura 5: *Representación de las funciones* $(S(\omega), C(\omega))$

Dado que la función es impar, podemos esperar otra espiral antisimétrica a ésta en el intervalo $[-5, 0]$. Además, cuando $\omega \to \infty$, la función tiende hacia el centro de la espiral. Nótese que el último punto dibujado en la función corresponde a los valores de $(S(5), C(5))$.

4 Conclusiones

En esta PEC 5 se ha estudiado la difracción de campo cercano, también llamada difracción de Fresnel, aplicada a una abertura a una distancia determinada de la pantalla, y con una longitud de onda de apertura concreta. Al fijar un valor para la frecuencia angular, podemos aproximar los valores de las funciones de Fresnel para ese valor de ω utilizando distintos métodos númericos, con distintos costes computacionales y precisiones. En esta prueba se ha utilizado la regla trapezoidal compuesta, así como su mejora mediante integración de Romberg, además de la cuadratura gaussiana de 64 puntos.

Por último, cabe recalcar la importancia de las integrales de Fresnel, pues sus aplicaciones van desde la óptica hasta la ingeniería de caminos y estructuras.

5 Listados de código

En esta sección se presentan los diferentes listados de código utilizados en los distintos apartados. Como ya incluyen comentarios, en el documento se hará una breve explicación del cometido y peculiaridades de cada programa.

5.1 Apartado (a)

El listado 1 muestra los pasos a seguir para obtener los vectores `trap_C` y `trap_S`.

Listado 1: *Código para ejecutar la regla trapezoidal compuesta, con comentarios*

```
% El vector "n" contiene el número de particiones que se realizarán en cada
% paso.

n = [1,2,4,6,8,16,32,64,128,256];

% Para obtener una representación más detallada de la convergencia
% n = 50 : 1 : 250;

%Definimos el intervalo de integración
a = 0;
b = 5;

trap_S = [];
trap_C = [];

for i = 1 : numel(n)
    h = (b-a)/n(i); %definimos "h"
    x = a : h : b;  %"x" es el vector de valores h-equiespaciados en [a,b]
    s = sin((pi/2)*x.^2); %definimos la función s
    c = cos((pi/2)*x.^2);   %definimos la función c

    %definimos las aproximaciones por la regla del trapecio

    trap_S(i) = h/2*(s(1) + 2*sum(s(2:end-1)) + s(end));
    trap_C(i) = h/2*(c(1) + 2*sum(c(2:end-1)) + c(end));
end
```

5.2 Apartados (b) y (c)

Aquí, debido a que se han utilizado dos métodos, tenemos dos programas distintos. El listado 2 muestra como, a partir de los vectores `trap_C` y `trap_S`, solamente queda realizar de nuevo la aproximación trapezoidal ahora con $N = 2n$, y aplicar la ecuación (4) para obtener los vectores `Rommberg_C` y `Romberg_S`

El listado 3, por otro lado, ejecuta directamente las tres ecuaciones mostradas en Burden y Faires (2011), pág. 215, en sus respectivos intervalos, para crear la matriz `R` de dimensión n. Cuando en este listado se referencian las funciones `senofres` y `cosenofres`, es porque han sido

definidas como funciones propiamente dichas, a diferencia de los listados anteriores. El listado 4 muestra las funciones definidas para los apartados (b) y (c).

El código utilizado para realizar la cuadratura gaussiana de 64 puntos se muestra en el listado 5. Simplemente se cargan los datos y se manipulan según el procedimiento a seguir. Nótese como se utiliza la función `save` al final del código para poder utilizar estos resultados en otros *live scripts*.

Listado 2: *Código para ejecutar la integración de Romberg por el primer método, con comentarios*

```
N = 2*n %n ya está definido, pero no N

masexacto_S = [];
masexacto_C = [];

for i = 1 : numel(N) % numel(n) = numel(N)
    h = (b-a)/N(i); %redefinimos "h"
    x = a : h : b;  %"x" es el vector de valores h-equiespaciados en [a,b]
    s = sin((pi/2)*x.^2); %definimos la función s
    c = cos((pi/2)*x.^2);   %definimos la función c

    %ejecutamos la regla trapezoidal compuesta para cada N

    masexacto_S(i) = h/2*(s(1) + 2*sum(s(2:end-1)) + s(end));
    masexacto_C(i) = h/2*(c(1) + 2*sum(c(2:end-1)) + c(end));
end

Romberg_S=[]; %definimos las aproximaciones por Romberg
Romberg_C=[];

for i = 1 : numel(n) %utilizamos la fórmula de la tutoría
    Romberg_C(i) = masexacto_C(i) + 1/(2^n(i)-1)*(masexacto_C(i)-trap_C(i));
    Romberg_S(i) = masexacto_S(i) + 1/(2^n(i)-1)*(masexacto_S(i)-trap_S(i));
end
```

5.3 Apartado (d) y tratamiento de datos

Para la primera parte del ejercicio, creamos una matriz con columnas $i = 1, \ldots, n$, los puntos x_i obtenidos por la regla trapezoidal y los puntos obtenidos mediante la cuadratura gaussiana. Se sigue un procedimiento idéndico al mostrado en el listado 6, donde se escribe la matriz a partir de la cual se generará la Tabla 1. Para generar la tabla en formato LaTeX, se crea la función `escribe_matriz_LaTeX`, similar a la utilizada en la PEC 2, y que se muestra en el listado 7.

Para calcular el error absoluto, se procede como en el listado 8, es decir, haciendo el valor absoluto de la resta entre el valor de referencia y la aproximación por cada método.

5.4 Apartado (e)

En este apartado he decidio usar estructuras de datos para pasar a las funciones, pues los *live script*s daban problemas para trabajar con variables globales comunes. En el listado 9 se muestra el procedimiento seguido para obtener la intensidad relativa. La primera parte no es más que la aplicación de la cuadratura gaussiana ya expuesta en el listado 5. Ahora las

Listado 3: *Código para ejecutar la integración de Romberg por el segundo método, con comentarios*

```
a = 0; %condiciones iniciales
b = 5;
n=16;

for k = 1 : n %definimos cada h_k
    h(k) = (b-a)/power(2,k-1);
end

for k = 2 : n
    R_C(1,1) = h(1)/2*(cosenofres(a)+cosenofres(b)); %primer elemento
    R_S(1,1) = h(1)/2*(senofres(a)+senofres(b));


    suma_C = 0;
    suma_S = 0;
    for i = 1 : power(2,k-2) &sumatorio del segundo paso
        suma_C = suma_C + cosenofres(a+(2*i-1)*h(k));
        suma_S = suma_S + senofres(a+(2*i-1)*h(k));
    end
        R_C(k,1) = 1/2*(R_C(k-1,1) + h(k-1)*suma_C); %primera columna
        R_S(k,1) = 1/2*(R_S(k-1,1) + h(k-1)*suma_S);
end

for j = 2 : n %calculamos cada elemento a partir de los anteriores
    for k = j : n
        R_C(k,j) = R_C(k,j-1) +
                (1/(power(4,j-1)-1))*(R_C(k,j-1)-R_C(k-1,j-1));
        R_S(k,j) = R_S(k,j-1) +
                (1/(power(4,j-1)-1))*(R_S(k,j-1)-R_S(k-1,j-1));
    end
end
```

Listado 4: *Funciones utilizadas en los apartados* (b) *y* (c)*.*

```
function c = cosenofres(x)
    c = cos((pi/2)*x^2);
end

function s = senofres(x)
    s = sin((pi/2)*x^2);
end

function ct= cosenofres(t)
    ct = cos((pi/2)*(25/4)*(t+1)^2);
end

function st = senofres(t)
    st = sin((pi/2)*(25/4)*(t+1)^2);
end
```

funciones que se nombran aparecen en el listado 10, que no son más que las funciones seno y coseno de frensel para u y v, con sus respectivos cambios de variable. Así, se calcula `Gauss_Cu` que es el código para $(C(u_2) - C(u_1))$, es decir, la aproximación de la integral que queremos. Lo mismo se hace para cada integral a calcular.

Finalmente, se calcula la intensidad relativa aplicando la ecuación (13) para obtener `I_I_0` $= I/I_0$.

Listado 5: *Código para ejecutar la cuadratura gaussiana de 64 puntos*

```
n = 64;

datos = readmatrix("QG64.dat"); %leemos los datos
r = datos(:,1); %raíces del polinomio de Legendre
w = datos(:,2); %valores de las función peso

c_r = []; %vector con la función \cos(\pi/2*x^2) en cada r_i
s_r = []; %vector con la función \sin(\pi/2*x^2) en cada r_i
for i = 1 : n
    c_r(i) = cosenofres(r(i));
    s_r(i) = senofres(r(i));
end

suma_c=0;
suma_s=0;
for i = 1 : n  %sumamos y multiplicamos por w_i
    suma_c = suma_c + w(i)*c_r(i);
    suma_s = suma_s + w(i)*s_r(i);
end

%recordar que dx = 5/2*dt
Gauss_C = 5/2*suma_c;
Gauss_S = 5/2*suma_s;

x=[]; %recogiendo los puntos x_i para representarlos
for i = 1 : n
    x(i) = 5/2*(r(i)+1);
end

save("resultados_apartado_c","Gauss_S","Gauss_C","x")
```

.

Listado 6: *Creando una matriz y exportándola*

```
M(:,1) = n;
M(:,2) = trap_S;
M(:,3) = trap_C;

cabeceras = "$n$ & $S(\omega)$ & $C(\omega)$";
columnasTabular =  "*3{r}";
escribe_matriz_LaTeX(M, cabeceras, columnasTabular, {'%d','%f','%f'},
        "apartado_a.tex");
```

Listado 7: *Función que escribe matrices en formato LaTeX*

```
function escribe_matriz_LaTeX(M,cabeceras,columnasTabular,
    formatoColumnas,nombreArchivo)

[n,m] = size(M);

str = sprintf("\\begin{tabular}{%s}\n\t\\toprule\n\t" + ...
"%s\\\\\n\t\\midrule\n\t", columnasTabular, cabeceras);
for i = 1 : n
    for j = 1 : m
        contenido = ['%s ',formatoColumnas{j}];
        str = sprintf(contenido, str, M(i,j));
        if j<m
            str = sprintf("%s &", str);
        else
            str = sprintf("%s \\\\\n\t",str);
        end
    end
end
str = sprintf("%s\\bottomrule\n\\end{tabular}\n",str);

fileID = fopen(nombreArchivo,"w");
fprintf(fileID,"%s",str);

fprintf("\nSe ha escrito la matriz en el archivo '%s'\n",nombreArchivo);
fprintf("\n");
fprintf("%s",str);

end
```

Listado 8: *Calculando el error absoluto*

```
load("resultados_apartado_c.mat")
Eabs_C_T = abs(Romberg_C(8)-trap_C(8));
Eabs_S_T = abs(Romberg_S(8)-trap_S(8));
Eabs_C_G = abs(Romberg_C(8)-Gauss_C);
Eabs_S_G = abs(Romberg_S(8)-Gauss_S);
```

Listado 9: *Cuadratura gaussiana y cáculo de la intensidad relativa*

```
n = 64;

%Estructura de datos para pasar a las funciones
param.n = n;
param.Lx = 4*sqrt(79)/25;
param.Ly = 6*sqrt(79)/25;
param.q_0 = 400;
param.lambda = 0.000632;

datos = readmatrix("QG64.dat");
r = datos(:,1);
w = datos(:,2);

cu_r = [];
su_r = [];
cv_r = [];
sv_r = [];
for i = 1 : n
    cu_r(i) = cosenofresu(r(i),param);
    su_r(i) = senofresu(r(i),param);

    cv_r(i) = cosenofresv(r(i),param);
    sv_r(i) = senofresv(r(i),param);
end

suma_cu=0;
suma_su=0;
suma_cv=0;
suma_sv=0;
for i = 1 : n
    suma_cu = suma_cu + w(i)*cu_r(i);
    suma_su = suma_su + w(i)*su_r(i);

    suma_cv = suma_cv + w(i)*cv_r(i);
    suma_sv = suma_sv + w(i)*sv_r(i);
end

Gauss_Cu = 4*sqrt(79)/50*suma_cu;
Gauss_Su = 4*sqrt(79)/50*suma_su;

Gauss_Cv = 6*sqrt(79)/50*suma_cv;
Gauss_Sv = 6*sqrt(79)/50*suma_sv;

I_I0 = 1/4*(Gauss_Cu^2 + Gauss_Su^2)*(Gauss_Cv^2 + Gauss_Sv^2);
```

Listado 10: *Funciones utilizadas en el apartado (e).*

```
function ctu = cosenofresu(t,param)
    n = param.n;
    Lx = param.Lx;
    Ly = param.Ly;
    q_0 = param.q_0;
    lambda = param.lambda;

    ctu = cos((pi/2)*(Lx/2*sqrt(2/(lambda*q_0)))^2*(t+1)^2);
end

function ctv = cosenofresv(t,param)
    n = param.n;
    Lx = param.Lx;
    Ly = param.Ly;
    q_0 = param.q_0;
    lambda = param.lambda;

    ctv = cos((pi/2)*(Ly/2*sqrt(2/(lambda*q_0)))^2*(t+1)^2);
end

function stu = senofresu(t,param)
    n = param.n;
    Lx = param.Lx;
    Ly = param.Ly;
    q_0 = param.q_0;
    lambda = param.lambda;

    stu = sin((pi/2)*(Lx/2*sqrt(2/(lambda*q_0)))^2*(t+1)^2);
end

function stv = senofresv(t,param)
    n = param.n;
    Lx = param.Lx;
    Ly = param.Ly;
    q_0 = param.q_0;
    lambda = param.lambda;

    stv = sin((pi/2)*(Ly/2*sqrt(2/(lambda*q_0)))^2*(t+1)^2);
end
```