

Biprisma de Fresnel. Teoría

March 15, 2021

1 Biprisma de Fresnel

Hay diferentes dispositivos que son equivalentes al sistema de dos rendijas originariamente empleado por Thomas Young. Aquí vamos a considerar el dispositivo llamado **biprisma de Fresnel**, el cual nos permitirá variar fácilmente la separación entre las rendijas o fuentes secundarias.

En la figura siguiente se muestra el esquema del dispositivo. Una fuente puntual situada en el eje del sistema e indicada en el dibujo por la letra S , ilumina el biprisma, formado por dos prismas delgados pegados por su base. El efecto de cada uno de estos prismas sobre la parte de la onda que los atraviesa es desviar su dirección de propagación. Esta desviación la podemos calcular mediante un trazado de rayos y lo que sabemos de Óptica Geométrica. Si el prisma es delgado, como estamos considerando, la desviación es constante e igual a,

$$\alpha = (n - 1)\gamma$$

En esta ecuación, α es el ángulo de desviación, o ángulo entre el rayo de salida del prisma y la continuación del rayo de entrada, n es el índice de refracción del material que forma el prisma, y finalmente, γ es el ángulo del prisma. En principio, consideraremos que ambos prismas son iguales, del mismo material y con el mismo ángulo.

El efecto que produce el biprisma es generar dos imágenes virtuales de la fuente real S , una por cada prisma, localizadas en el plano perpendicular al eje y que pasa por S . Estas dos imágenes virtuales actúan como dos fuentes secundarias de radiación, equivalentes a las dos rendijas del dispositivo de Young. La luz que emerge por la parte superior del biprisma lo hace tal como si procediera de S_1 , mientras que la parte del haz refractada por la parte inferior del biprisma se propaga como si procediera de S_2 . Por lo tanto, a la derecha del biprisma tenemos la superposición de dos ondas esféricas procedentes de S_1 y S_2 , respectivamente. El plano donde se encuentran estas fuentes virtuales es equivalente al plano de las rendijas en el experimento de Young.

De este modo, a la derecha del biprisma, el conjunto fuente real S + biprisma es equivalente a dos fuentes S_1 y S_2 localizadas en el mismo plano que S . Si colocamos una pantalla en esta región del espacio, observaremos franjas de interferencia en la zona en la que se superpongan las ondas refractadas por cada parte del biprisma. La interfranja de este patrón será igual que en el caso del experimento de Young:

$$Int = \frac{\lambda D}{a}$$

donde ahora D es la distancia desde el plano de las fuentes S_1 y S_2 (es decir, desde la fuente real S) hasta la pantalla y a es la distancia entre S_1 y S_2 , ya que estas fuentes juegan el mismo papel que las rendijas.

1.1 Separación entre las fuentes virtuales (a)

Como hemos visto, la separación entre las fuentes virtuales, que hemos denominado a , determina la interfranja. Vamos a ver que el dispositivo del biprisma de Fresnel permite modificar fácilmente esta distancia. Para ello, vamos a ver de qué depende esta magnitud.

Las fuentes virtuales S_1 y S_2 son las imágenes de la fuente real S que proporcionan los prismas que componen el biprisma. Esto quiere decir que, desde el punto de vista de la Óptica Geométrica, la prolongación de los rayos de salida de cada prisma se cortan en los puntos donde están situadas estas imágenes. Concretamente, nos podemos fijar en el rayo que, pasando por el prisma superior, lo hace justo pegado a la base del mismo y por tanto al eje del sistema. Este rayo se desviará, al igual que el resto, un ángulo α . Es evidente en la figura que este ángulo es también el que forma la prolongación del rayo de salida con el eje. De este modo, vemos que:

$$\tan(\alpha) = \frac{a/2}{d}$$

donde d es la distancia entre la fuente S y el biprisma. Por tanto, despejando a , obtenemos:

$$a = 2d \tan(\alpha) = 2d \tan[(n-1)\gamma]$$

Es decir, la separación entre las fuentes S_1 y S_2 depende de las propiedades del biprisma (de su índice de refracción y del ángulo de los prismas, que determinan el ángulo de desviación α), y, más importante, de la distancia d entre el biprisma y la fuente puntual que lo ilumina. Cambiando esta distancia modificamos el valor de a y por tanto podemos modificar la interfranja del diagrama interferencial.

1.2 Extensión del patrón de interferencia en la pantalla

Para que se produzca interferencia en la pantalla debemos de situarnos en una zona del espacio en donde se solapen los haces que pasan por cada uno de los prismas. Como se puede observar en la figura, esto no ocurre en todo el espacio a la derecha del biprisma. Por lo tanto, la zona en la pantalla en la que observemos interferencia estará limitada en su extensión. Para calcular esta extensión nos podemos fijar en la figura que la zona en la que se solapan los haces viene determinada por el ángulo de desviación α , y que, si llamamos z_{ext} a la altura en la pantalla del extremo del patrón, tenemos,

$$\tan(\alpha) = \frac{z_{ext}}{D-d} \Rightarrow z_{ext} = (D-d) \tan(\alpha)$$

La extensión total del patrón, será, por supuesto $2z_{ext}$.

Hay que resaltar que esta limitación en la extensión del patrón de franjas no es debido en este caso a una longitud de coherencia finita pues en este desarrollo estamos suponiendo que la radiación es monocromática. Si ilumináramos con radiación no monocromática, tendríamos que añadir este efecto para calcular la extensión del patrón. Para ello, localizaríamos el punto en la pantalla en donde $\Delta = l_c$ siendo Δ la diferencia de camino y l_c la longitud de coherencia de la fuente. Si este punto tiene una altura con el eje menor que el valor anteriormente calculado z_{ext} , entonces limitaría más la extensión del patrón. Si por el contrario es mayor, la condición que limitaría la extensión del diagrama interferencial sería la anterior (z_{ext}) es decir, el solapamiento de los dos haces.

Ejecutando el siguiente código se genera una figura interactiva en donde, con un desplazador, se puede variar la distancia entre la fuente y el biprisma (distancia d en nuestro desarrollo anterior). Observa cómo cambia el patrón de franjas:

1. ¿Cambia la Interfranja?. Fija un valor de la distancia fuente-biprisma y calcular con las expresiones anteriores el valor de la interfranja esperado. ¿Obtienes el mismo valor? (Observa que el valor de la interfranja viene calculado directamente en la caja de la derecha que aparece con la figura).
2. Cambia en el apartado de “Datos fijos” del código el valor de D , λ y α , y observa los cambios en el patrón. ¿Se corresponden con lo explicado anteriormente?
3. Cambia por último el valor de la irradiancia de uno de los haces (variables $I1$ e $I2$). ¿Qué esperarías que ocurriera en el patrón?. ¿Se corresponde lo que ocurre con tu idea?

Aparte de este estudio, nos podemos plantear qué cambiaría en las anteriores ecuaciones si fabricáramos un biprisma en donde el prisma superior no es igual al inferior. Por ejemplo, que los prismas que componen el biprisma no tuvieran el mismo ángulo, o se construyeran con materiales con índices de refracción distintos. Razona qué habría que cambiar en el anterior desarrollo y cómo afectaría al patrón de franjas.

```
[1]: import warnings
warnings.filterwarnings('ignore')
import numpy as np
import matplotlib.pyplot as plt
##matplotlib inline
plt.style.use('ggplot')
import ipywidgets as widgets
from IPython.display import display
import io
import base64
from IPython.display import clear_output

#Datos fijos
#####33
D = 1 # Distancia fuente-pantalla
Lambda = 6.33e-7 # longitud de onda de la radiación
```

```

alpha = 0.7*np.pi/180 # ang. del biprisma

I1 = 1 # Consideramos irradiancias normalizadas a un cierto valor.
I2 = 1
#####
k = 2.0*np.pi/Lambda
n = 1.33 # agua
y0 = 3.5# tamaño vertical biprisma
yrepres = 4.5 # in mm. Parametro de representación

fig,ax = plt.subplots(1,1)
ax.plot(0,0,'o',lw=2)
ax.set_xlim(-0.1,D+0.1)
ax.set_ylim(-yrepres*1.5,yrepres*1.5)
ax.set_xlabel("x (m)")
ax.set_ylabel("y (mm)")
ax.set_title('Esquema del montaje experimental')
line1, = ax.plot(np.linspace(-1,D,50),np.zeros(50),'k')
buf = io.BytesIO()
plt.savefig(buf, format='png')
buf.seek(0)
figwidg = widgets.HTML("""<img src='data:image/png;base64,{}/>""".
    ↳format(base64.b64encode(buf.getvalue()).decode('ascii'))))
plt.close(fig)

fig2,ax2 = plt.subplots(1,1)
ax2.set_xlabel("x (mm)")
ax2.set_ylabel("y (mm)")
ax2.set_title('Pantalla')
ax2.pcolormesh(np.zeros((500,500)))
buf2 = io.BytesIO()
plt.savefig(buf2, format='png')
buf2.seek(0)
figwidg2 = widgets.HTML("""<img src='data:image/png;base64,{}/>""".
    ↳format(base64.b64encode(buf2.getvalue()).decode('ascii'))))
figbox = widgets.HBox([figwidg,figwidg2])
plt.close(fig2)
clear_output()

#####
Interfwidg = widgets.FloatText(0,description='Int. (mm)',color='#C13535')

def changeBiprism(x0=0.1):
    a = 2*x0*np.tan((n-1)*alpha) #separacion entre fuentes virtuales
    interfranja = Lambda*D/a

```

```

    xlimit = (D-x0)*np.tan((n-1)*alpha) # distancia desde el eje que ocupa el
    ↪patron de interferencias
    Pasoespacial = interfranja/20
    xpantalla = np.linspace(-xlimit*1.2,xlimit*1.2,int(xlimit/Pasoespacial))
    X,Y = np.meshgrid(xpantalla,xpantalla)
    delta = (k*a*X/D)
    Itotal = I1 + I2 + (2.0*np.sqrt(I1*I2)*np.cos(delta))*(np.abs(X)<xlimit)
    figwidg.observe(repsetup(x0),names='new')
    figwidg2.observe(repfringes(xpantalla,Itotal),names='new')
    Interfwidg.observe(updateInterf(interfranja),names='new')
    return

def updateInterf(interfranja):
    Interfwidg.value=round(interfranja*1e3,2)
    return

def repsetup(x0):
    fig,ax = plt.subplots(1,1,figsize=(13,8))
    ax.plot(0,0,'o',lw=2)
    ax.set_xlim(-0.1,D+0.1)
    ax.set_ylim(-yrepres*1.5,yrepres*1.5)
    ax.set_xlabel("x (m)")
    ax.set_ylabel("y (mm)")
    ax.set_title('Esquema del montaje experimental')
    ax.plot(np.linspace(-1,D,50),np.zeros(50),'k')
    ax.vlines(x0,-y0,y0,'r',lw=2)
    ax.vlines(D,-y0*3,y0*3,lw=4)
    ax.plot(np.linspace(x0,x0+.01,3),(-y0/.01)*np.linspace(x0,x0+.01,3) +
    ↪y0*(1+x0/.01),'r',lw=2)
    ax.plot(np.linspace(x0,x0+.01,3),(y0/.01)*np.linspace(x0,x0+.01,3) -
    ↪y0*(1+x0/.01),'r',lw=2)
    buf = io.BytesIO()
    plt.savefig(buf, format='png')
    buf.seek(0)
    clear_output()
    figwidg.value = ""<img src='data:image/png;base64,{}/>"".format(base64.
    ↪b64encode(buf.getvalue()).decode('ascii'))
    clear_output()
    plt.close(fig)
    return

def repfringes(x,Itotal):
    fig2,ax2 = plt.subplots(1,1,figsize=(4,8))
    ax2.set_ylabel("y (mm)")
    ax2.set_xlim(0,x.max()*1e3)
    ax2.set_ylim(-yrepres,yrepres)

```

```

ax2.set_title('Pantalla (mm)')
ax2.pcolormesh(x*1e3,x*1e3,Itotal.T,cmap = 'gray',vmin=0,vmax=4)
buf2 = io.BytesIO()
plt.savefig(buf2, format='png')
buf2.seek(0)
clear_output()
figwidg2.value = """<img src='data:image/png;base64,{}/>""".format(base64.
↪b64encode(buf2.getvalue()).decode('ascii'))
clear_output()
plt.close(fig2)
return

```

```

Positionwidg = widgets.FloatSlider(value=0.3,min=0.08,max=0.35,step=0.
↪02,description='Fuente-Biprisma (m)',orientation='horizontal')
changeupos = widgets.interactive(changeBiprism,x0=Positionwidg)
resultswidg = widgets.HBox([figbox,Interfwidg])
display(changeupos,resultswidg)

```

[1]: interactive(children=(FloatSlider(value=0.3, description='Fuente-Biprisma (m)', max=0.35, min=0.08, step=0.02)...

[1]: HBox(children=(HBox(children=(HTML(value="<img src='...

[0]: