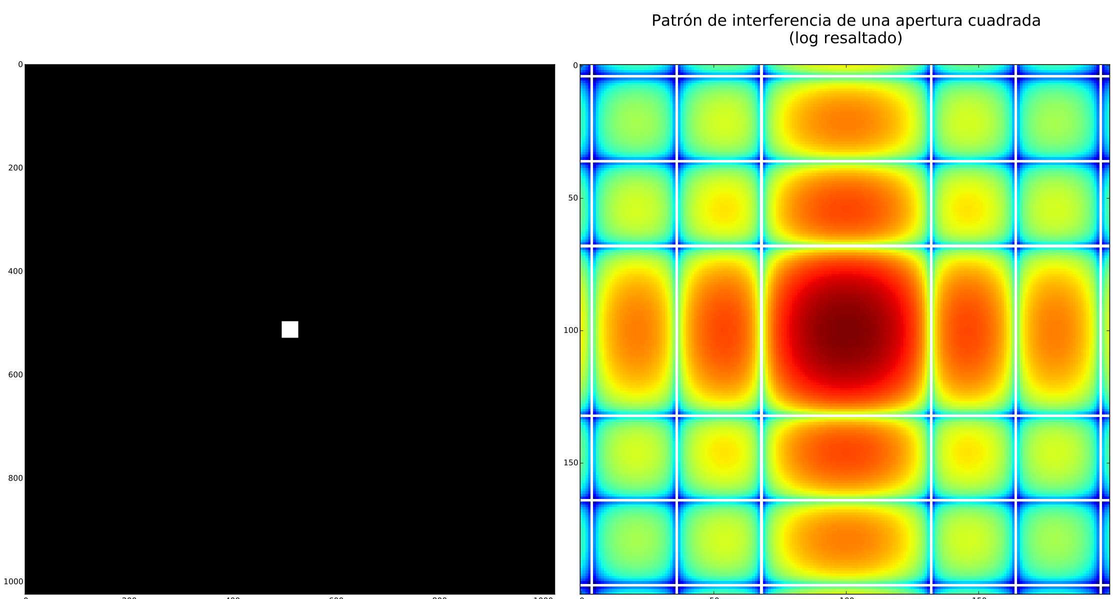
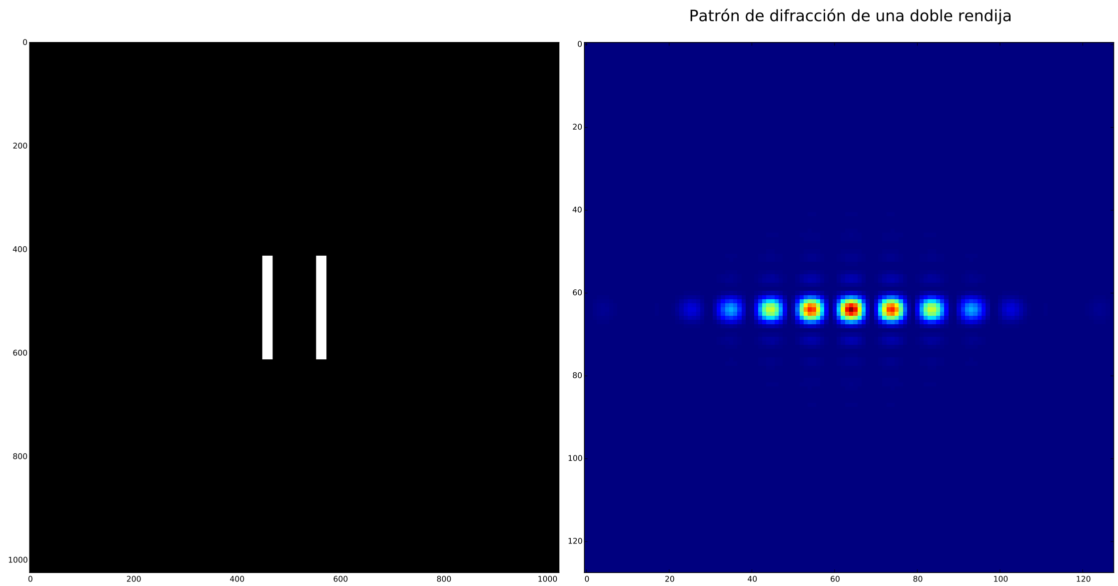
La solución a este taller debe presentarse en un *notebook* llamado `HW4.ipynb` y debe cargarse a su repositorio en GitHub en la carpeta /MC/Tareas/HW4/. Es requisito que en todo lo hecho se pongan comentarios que expliquen lo que se está haciendo. La fecha límite para hacer un commit es el **jueves 25 de junio a las 23:59**. Puede trabajar en parejas.

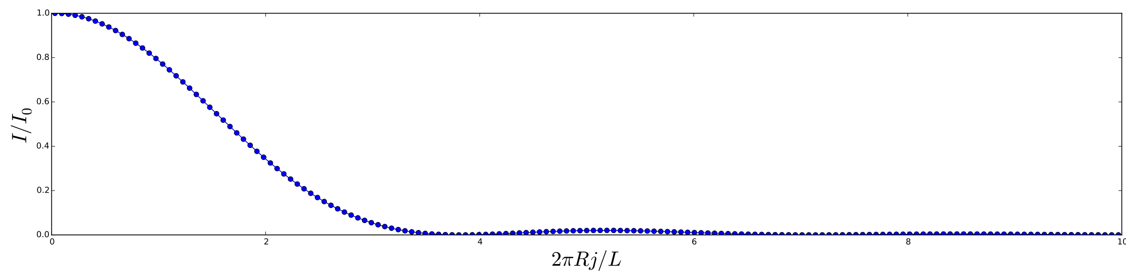
1. (Difracción de Fraunhofer) El patrón de difracción producido producido por una apertura en el régimen de Fraunhofer (ver Born & Wolf, Principles of Optics capítulo 8) puede encontrarse a través de una integral de Fourier y aproximarse numéricamente a través de una DFT. En este ejercicio exploramos los patrones de difracción producidos por aperturas de diferentes formas usando diferentes *pupil functions* (ver Born. pág 385). Imaginamos que las aperturas se encuentran en el centro de una placa de lado $L = 1$ u y la representamos por medio de un array cuadrado de 1024 elementos en cada dirección.
   (a) $\boxed{\text{20 pt}}$ Comenzando con un array de ceros de lado 1024 manipularlo con ciclos o usando *slice notation* para dejar todo en cero excepto elementos en un cuadrado de lado 32 en el centro.
   (b) $\boxed{\text{20 pt}}$ Calcular la DFT del array resultante y aplicar sobre él la función fftshift, esto hace un cambio de coordenadas que arroja el patrón de interferencia esperado. De lo resultante tomar del centro un cuadrado de lado 200 y producir una imagen del logaritmo del módulo al cuadrado usando `imshow` con la representación de la apertura a su izquierda.

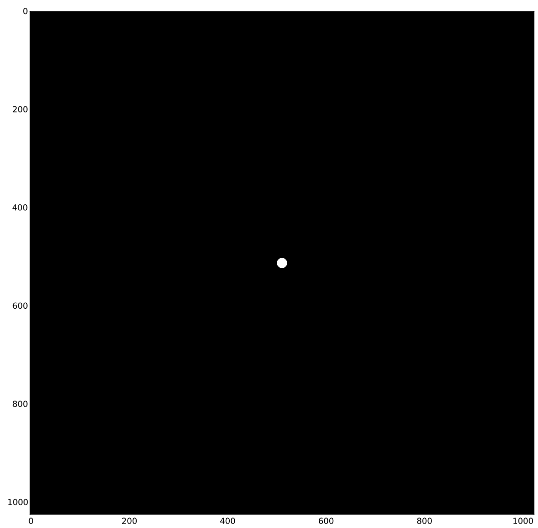


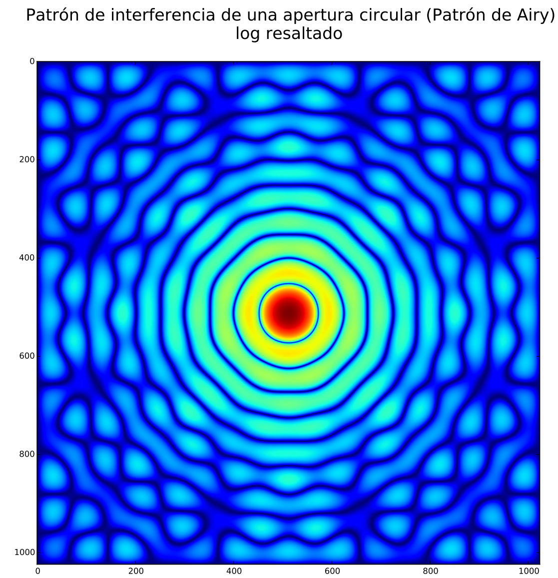
(c) $\boxed{\text{20 pt}}$ Repita los dos anteriores literales para una doble rendija cada una de ellas con una altura de 200, grosor 20 y con sus ejes en las columnas 460 y 564. Esta vez tome un cuadrado de lado 128 para la gráfica del patrón de difracción y no resalte con `log`.



(d) $\boxed{\text{20 pt}}$ Ahora hágalo para una apertura circular de radio 0.01 u y además de lo anterior tome ahora la fila central y tomando como abscisas `2*pi*radio/1.*linspace(-512,512,1024)` reproduzca la gráfica mostrada abajo; las abscisas están así elegidas para hacer comparable nuestro resultado con lo mostrado en la pág. 396 de los *Principles* de Born. I_0 es la amplitud máxima.







(e) $\boxed{20 \text{ pt}}$ Usando las mismas ordenadas y abscisas del anterior literal calcular los máximos y mínimos entre 0. y 12.0. Al hacerlo **debe** utilizar interpolación en algún momento. Compare sus resultados con la tabla mostrada en la pág. 397 de los *Principles*:

TABLE XX

The first few maxima and minima of the function

$$y = \left(\frac{2J_1(x)}{x}\right)^2.$$

x	$\left(\frac{2J_1(x)}{x}\right)^2$	
0	1	Max.
$1{\cdot}220\pi = 3{\cdot}833$	0	Min.
$1{\cdot}635\pi = 5{\cdot}136$	0·0175	Max.
$2{\cdot}233\pi = 7{\cdot}016$	0	Min.
$2{\cdot}679\pi = 8{\cdot}417$	0·0042	Max.
$3{\cdot}238\pi = 10{\cdot}174$	0	Min.
$3{\cdot}699\pi = 11{\cdot}620$	0·0016	Max.

COMENTARIOS:

- Si al aplicar `log` encuentra una advertencia sobre el intento de hacer `log(0)` sume 1 a todo antes de usarlo.
- El panel para la apertura circular se hizo con la ayuda de `subplot2grid`:

```
plt.figure(figsize=(20,15))
plt.subplot2grid((3,2),(0,0),colspan=2,rowspan=1)

...

plt.subplot2grid((3,2),(1,0),rowspan=2)

...

plt.subplot2grid((3,2),(1,1),rowspan=2)

...

plt.tight_layout()
plt.show()
```