Tasks:
Part 1

- Choose N when sampling the double slit pair function provided to result in a propagated set of coordinates that is 64cm in extent at L = 10m for an initial region of size 20mm. The scale factor is given by $s = \frac{N\lambda L}{(2l)^2}$. N is the number of samples, L is the distance propagated, and l is half the extent of the initial coordinates
- Compute a Fraunhofer propagation on the samples provided to distance 10m

Part 2

- Use the **numpy.convolve** function to calculate the true convolution of the 10 millimeter boxcar and the provided intensity. In MATLAB, the function to use is **conv**, and instead of *mode* one selects the *shape* parameter. Then run the code to do a calculation of the circular convolution. The *mode/shape* argument of **numpy.convolve/conv** is crucial, you want to pick it so that the number of samples remains N not more not less, so that you can use the same coordinates
- Answer: what differences do you notice between the results of the functions circular_convolve, and **numpy.convolve/conv**? Where are the differences most prominent?

Circular convolutions (FYI, don't need to know yet):
In class we discussed regular convolutions, where the overlap is taken by simply shifting one of the convolved functions across the other. In the discretized case, one can also define a circular convolution where the function that is swept across the other “wraps around” when it hits a boundary, resulting in periodic effects not seen in a regular convolution. This convolution rarely represents a physical process like observation by a finite sized observer, but is easier to compute by the following:

$$(B_j \circledast A_j)_n = DFT_n^{-1}\{DFT\{B_j\} \cdot DFT\{A_j\}\}$$

Where the encircled star denotes the circular convolution rather than a regular convolution. The dot inside the inverse DFT is just multiplication.

I provide example plots to show what your correct results should look like.