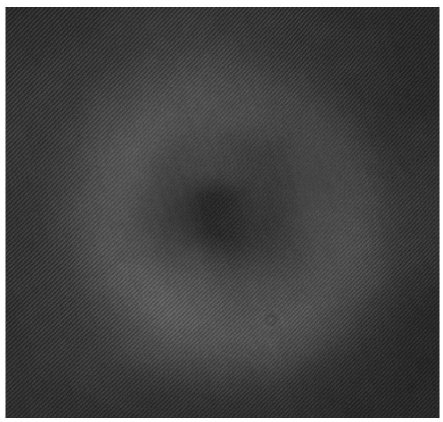
Data extraction: from a picture to a flow!

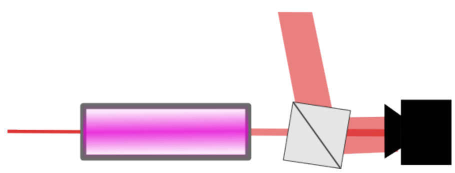
7 May 2024

We consider a laser beam after propagation in a Rubidium cell and want to extract as much information as possible from it using only a camera. We will consider this picture:



1 Field recovery: Off-axis Interferometry

The first thing to do is to be able to recover the complete field $E_s(\overrightarrow{r})e^{i\phi(\overrightarrow{r})}$from just a picture. This is done with an off-axis interferometry setup: the signal beam interfere with a reference beam making a small angle.

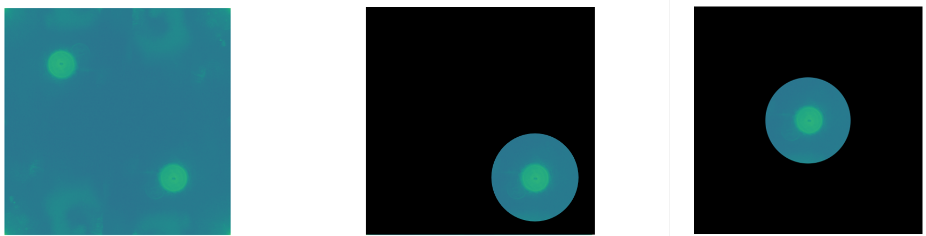


The intensity on the camera is then:

$$I = \alpha \left| E_s(\overrightarrow{r})e^{i\phi(\overrightarrow{r})} + E_r(\overrightarrow{r})e^{i\overrightarrow{k}\cdot\overrightarrow{r}} \right|^2 = I_s + I_r + 2\sqrt{I_r I_s}\cos\left(\overrightarrow{k_\perp}\cdot\overrightarrow{r_\perp} + \phi\right)$$

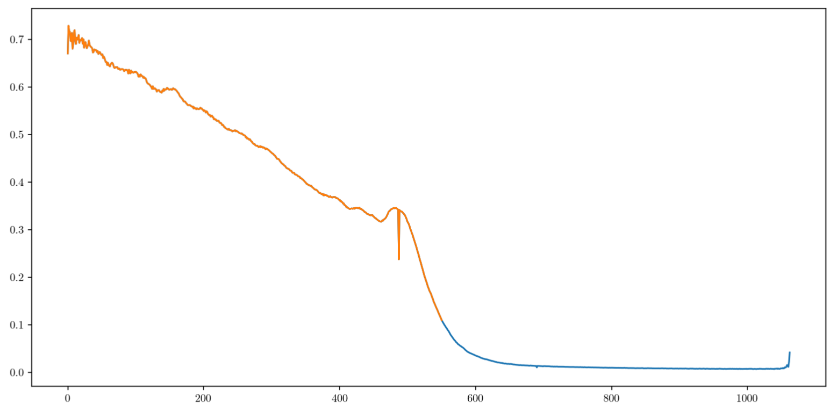
Taking the Fourier transform, we get an order zero due to I_s and I_r, and two order one peaks with the cosinus. Assuming that I_r varies slowly with respect to $\overrightarrow{k_\perp}\cdot\overrightarrow{r_\perp}$, the Fourier transform of the cosinus extract the one of the field:

$$\hat{I}(\overrightarrow{k}) = \hat{I}_s + \hat{I}_r + \mathcal{F}\left[E_s e^{i\phi}\right]\left(\overrightarrow{k} - \overrightarrow{k_\perp}\right) + \mathcal{F}\left[E_s e^{i\phi}\right]\left(\overrightarrow{k} + \overrightarrow{k_\perp}\right)$$

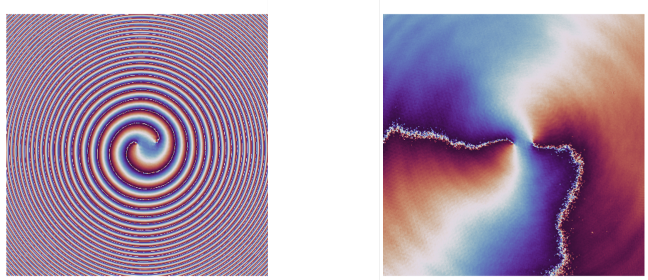


So we start by isolating the first order and then roll it to the center:

By calculating the inverse Fourier transform, we can get the complete complex field. We now need to find the borders of the beam. This is done by calculating the azimutal average of the amplitude and keep only the part where the amplitude is larger than 10%[1] of the largest value. This define our region of interest.



But we see that the phase is dominated by the defocusing phase, thus we also measure with the same method the phase of a pure gaussian beam propagating through the cell and substract the two, to get only the phase due to the vortices.



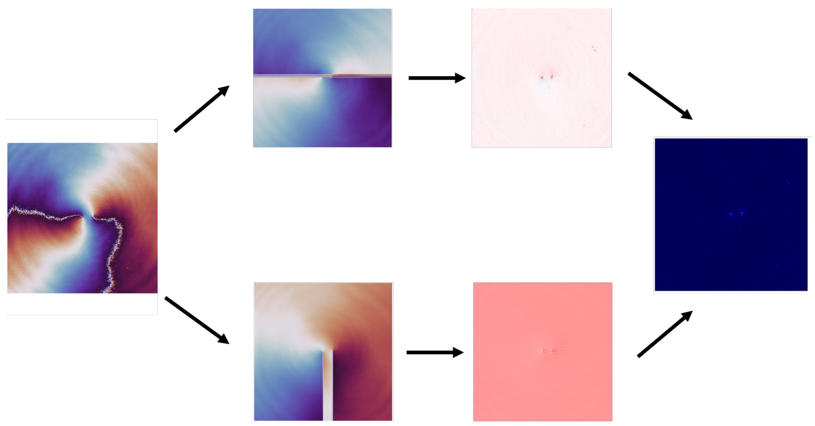
[1]This threshold is an arbitrary value

2 Velocity decomposition

In the hydrodynamic picture, the light intensity can be seen as the fluid density and the velocity is given by the gradient of the phase $\overrightarrow{v} = \nabla\phi$. To calculate this gradient we first need to unwrap the phase such that it is not limited between 0 and 2π.

However, if we unwrap the phase in the whole 2D image and calculate the gradient, we will have divergences due to the singularities of the vortices. To avoid this problem, we unwrap the phase along x and calculate the velocity along y from this picture and the velocity along x from the unwraped phase alon y.

Combining the two we get the velocity amplitude:



We now want to separate the two important parts of the flow: the vortices and the sound waves, that is to say incompressible and compressible flow. However, by its definition as a gradient, the velocity field is incompressible, thus we introduce a density weighted velocity $u = \sqrt{\rho}\,\nabla\phi$, that can be decomposed as:

$$\overrightarrow{u} = \overrightarrow{u}_{comp} + \overrightarrow{u}_{inc} = \nabla\varphi + \nabla_{\wedge}\overrightarrow{\Psi}$$

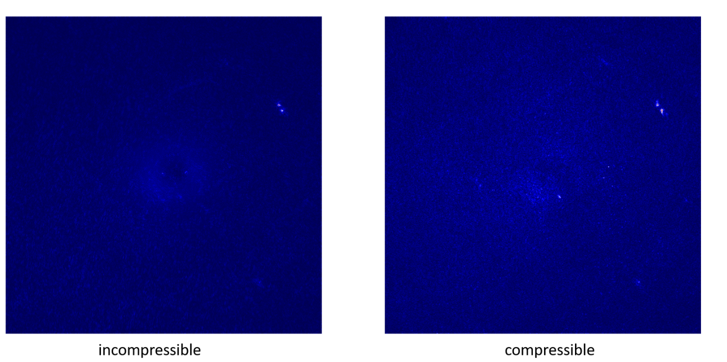
Going to Fourier space, we get:

$$\overrightarrow{U}(\overrightarrow{k}) = i\overrightarrow{k}U_{\varphi}(\overrightarrow{k}) + i\overrightarrow{k}_{\wedge}\overrightarrow{U}_{\boldsymbol{\Psi}}(\overrightarrow{k})$$

So the compressible part is simply obtained by dotting with the wavevector

$$U_{\varphi}(\overrightarrow{k}) = -i\frac{\overrightarrow{k}\cdot\overrightarrow{U}(\overrightarrow{k})}{k^2}$$

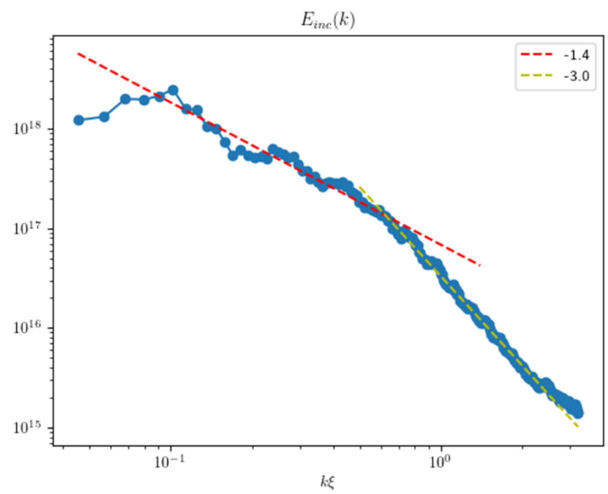
And the incompressible part immediatly follow by difference.



And finally, we can calculate the compressible and incompressible energy spectrum by taking the azimutal average of the square of each component of the velocity:

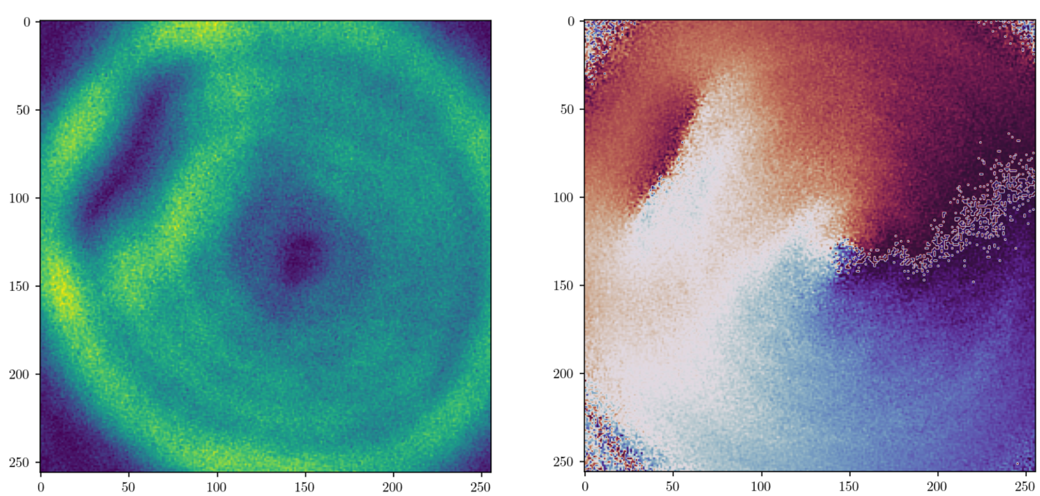
$$E_{inc} = \frac{m}{2} k \int_0^{2\pi} |U_{inc}|^2 \mathrm{d}\Omega_k$$

$$E_{comp} = \frac{m}{2} k \int_0^{2\pi} |U_{comp}|^2 \mathrm{d}\Omega_k$$

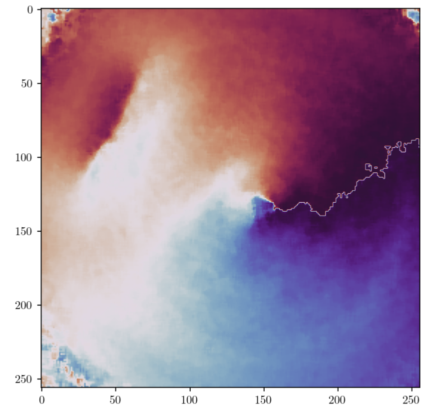


3 Vortex and soliton detection

We have now all the information about our flow, but it could be interesting to find the position of objects in it: mostly vortices and solitons. For the example, we will use this picture with one central vortex and a soliton

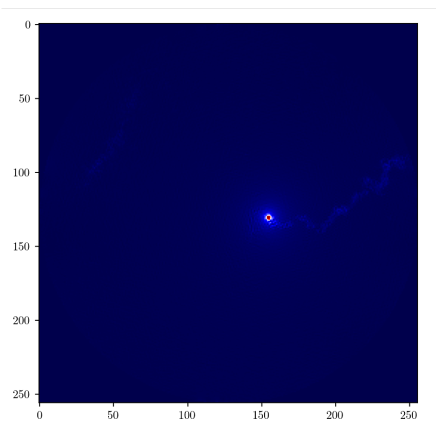


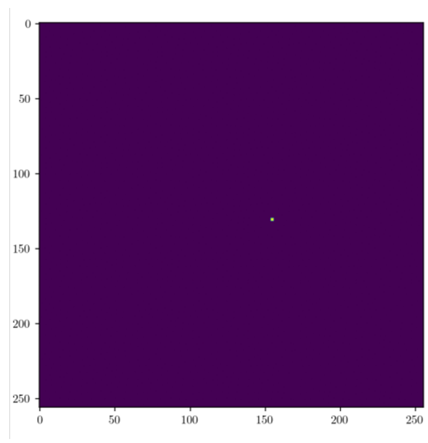
First, we have a very noisy picture, we need to filter it. However we cannot use a gaussian filter because since the soliton is a smooth phase jump, it will be smoothed out by the gaussian. The solution is to use a median filter on the phase:



To detect the vortices, a possibility would be to calculate the phase circulation around each pixel. If it is a multiple of 2π, we have a vortex! But this is costly in term of computation, and in the case with a few vortices[2] we can do something easier. The vortices are indeed singularities of the velocity, and so we can find them by calculating the velocity.

Once the velocity is calculated we keep only the regions above a given threshold, decompose them in connected components and take as vortex position the position of the maximum in each connected component:

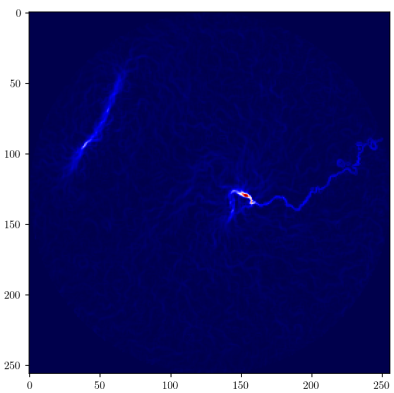




[2]But in the case with a lot of vortices, nothing is better than this circulation calculation because the other method will often not distinguish close vortices.

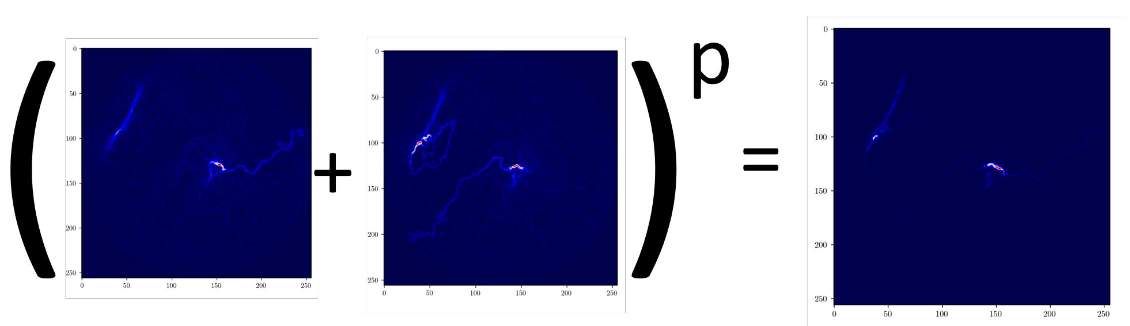
It is now time to detect the soliton by detecting the edge in the phase map! It is done using a variation of Canny edge detection algorithm. First we calculate the gradients in the picture by convolving the picture with a Sobel Kernel with better rotationnal symetry (the other direction gradient is obtained by transposing the kernel).

$$\begin{bmatrix} 3 & 10 & 3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{bmatrix}$$

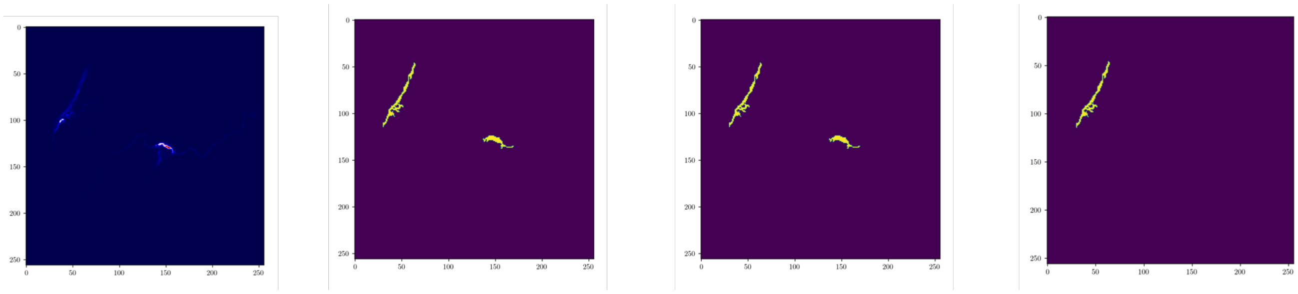


We see that the phase cut in the image results in a line of large gradient. To remove it, the solution is to do the same but after adding π to the initial phase, which rotates the phase cut.

Since the phase cut is non physical, we want to remove it and keep only the real features of the image that should be in the two versions of the these gradient. Therefore, we renormalize the picture and calculate the new gradient as $\nabla = (\nabla_1 + \nabla_2)^p$ with $p > 1$. The common features give hence larger than 1 sum enhance by the power, while the phase cut appearing in only one of the two picture, gives a lower than 1 sum and thus is reduced by the power:



Then to detect the edges, we use double threshold: the points where the gradient is smaller than a lower threshold are set to zero, while the one where the gradient are larger than the higher treshold are set to 2: these are real edges. The other points are set to 1 and called weak edges.Then all the weak edges in a connected component with strong edges are converted to strong edges and the others are removed. However, this also gives edges at the position of the vortices, and thus we remove the edges connected to the position of a vortices.



The remaining pixels are the position of the soliton!