Optical cavity for spectral cleaning of a laser and matter-wave diffraction experiments

for a cold-atom based quantum simulator

by

Pauline Guesdon

March 27th - July 28th

Supervisors : Dr. Benjamin Pasquiou, Dr. Martin Robert de Saint Vincent,

Dr. Bruno Laburthe-Tolra

Sorbonne Université Sciences

Université Paris 13

CONTENTS

List of Figures **2**

1 Introduction **3**

2 Programming to fit diffraction patterns **6**

2.1 Band theory : about diffraction [2][3][4] 6

2.2 Programming 8

3 Cavity **11**

3.1 Spin manipulation by Raman spectroscopy 12

3.2 Introduction to cavity 14

3.3 Confocal cavity 14

3.4 Experiment 15

3.4.1 Rough alignement 16

3.4.2 Overlap the two beams 18

3.4.3 Adjusting the confocality and transmission 18

3.4.4 Pound-Drever-Hall technique for the lock 19

3.4.5 Analysis 21

4 Conclusion **23**

5 Supplement **24**

References **30**

LIST OF FIGURES

1.1 Experimental set-up 4
1.2 Transitions on the experiment 5
2.1 Diffraction pattern on a CCD camera 8
2.2 Exemple of a diffraction fit 10
2.3 Predicted population evolution of diffraction orders 11
3.1 Raman transition 12
3.2 Hermite-Gauss modes 14
3.3 Confocal cavity 15
3.4 Experimental set-up of mirrors gluing 16
3.5 Even and odd resonant peaks 18
3.6 Pound-Drever-Hall technique 19
3.7 Experimental sidebands and carrier - signal modulation 20
3.8 Experimental error signal by lock-in amplifier 21

1. INTRODUCTION

The experiment led by the Magnetic Quantum Gas team with whom I'm working does cold atoms researches focused on quantum simulators : these experimental simulators are made to figure out complex systems that are described by hamiltonians unsolved by classical computers.
This simulation is about trapping in a 3D optical lattice strontium 87 fermions of purely nuclear spin $F = 9/2$ that has ten spin states in its fondamental form. The interest of this trapping is to obtain the Fermi-Hubbard model describing fermions in optical lattices. In the case of deep lattices of depth U, where $t/U << 1$, t the tunneling, we can approximate this system as a Mott isolator at low temperature. [1] We can then consider that the interaction between the neighbours atoms is a second order pertubation because it costs much energy to have two particles in the same well. The interaction that comes from this is called *superexchange interaction*

$$H_{superexchange} = \frac{J}{2} \sum_{i,j} \vec{S_i}.\vec{S_j} \tag{1}$$

$J = \frac{t^2}{2}$, $\vec{S_i}.\vec{S_j}$ the spin vectors of two neighbours atoms j and i.

We cool the atoms in many steps :

1. First, the atoms are obtained by heating solid strontium at temperature $T = 760$K to have a flux of 6.10^{12} atoms/s with a mean velocity $v \simeq 500$m/s. [2]
2. Then the gaz arising from this first part is collimated by transerve 2D optical molasses that are laser Doppler cooling the atoms. They absorb anisotropic light and re-emit isotropic light because of the momentum conservation.
3. The previous step is followed by a Zeeman slower where they reach a velocity of 20m/s. This is a combination of a counterpropagating laser responsible for absorption and emission cycles, and a magnetic field that shifts the energy levels of the atoms in order to issue resonance despite modification.

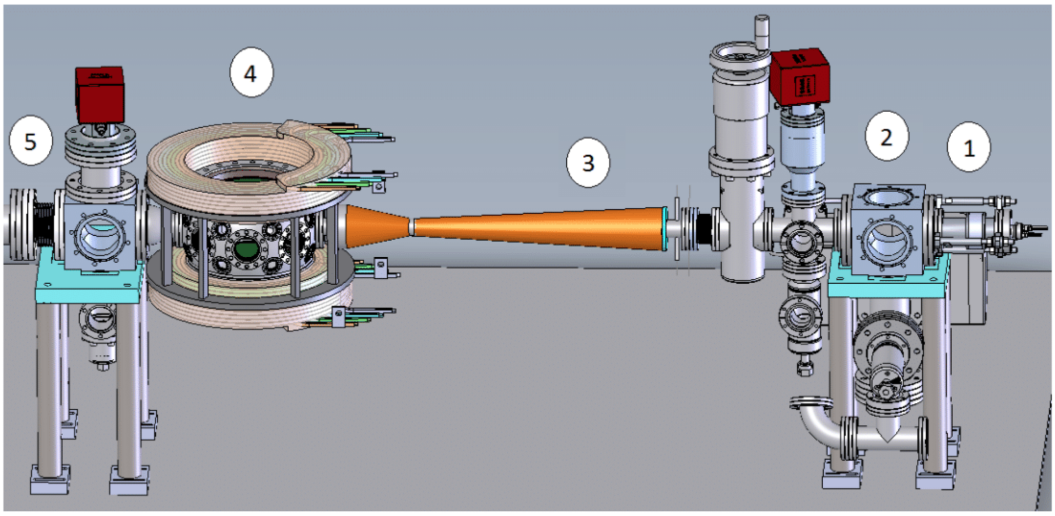


Figure 1.1: Experimental set-up of the cooling and trapping. (1) The atoms going out of an oven that creates an atomic gaz at a flux 6.10^{12} atoms/s (2) Cooling by 2D optical molasses (3) Zeeman slower of the atoms (4) Trapping in 3D MOT (5) Trapping with a dipolar trap and cooling by evaporation

Then we trapp the atoms

4. They are trapped in a 3D Magneto Optical Trap (MOT) on the broad-band transition ($\Gamma \simeq 30.5$MHz) at 461nm, ${}^1S_0 -{}^1 P_1$ transition visible on image 1.2 (blue narrow). A MOT combines Doppler cooling and Zeeman slowing, made of counterpropagating lasers. Particules are cooled down at temperature $T \simeq 1$mK. They are then transfered to a second narrow MOT ($\gamma \simeq 7.4$kHz) on the transition ${}^1S_0 -{}^3 P_1$.

5. Finally they are transfered in a dipolar trap where they are cooled by evaporation. By this technique we remove the warmer atoms by progressively reducing the depth of the trap. At this stage we have what we call a *Fermi sea*.

   We can now trapp them on a 1D lattice at 1064 nm and a 532 nm 2D lattice. They result from retroreflected lasers. They are creating interfering plans, trapping this way the atoms on potential wells.

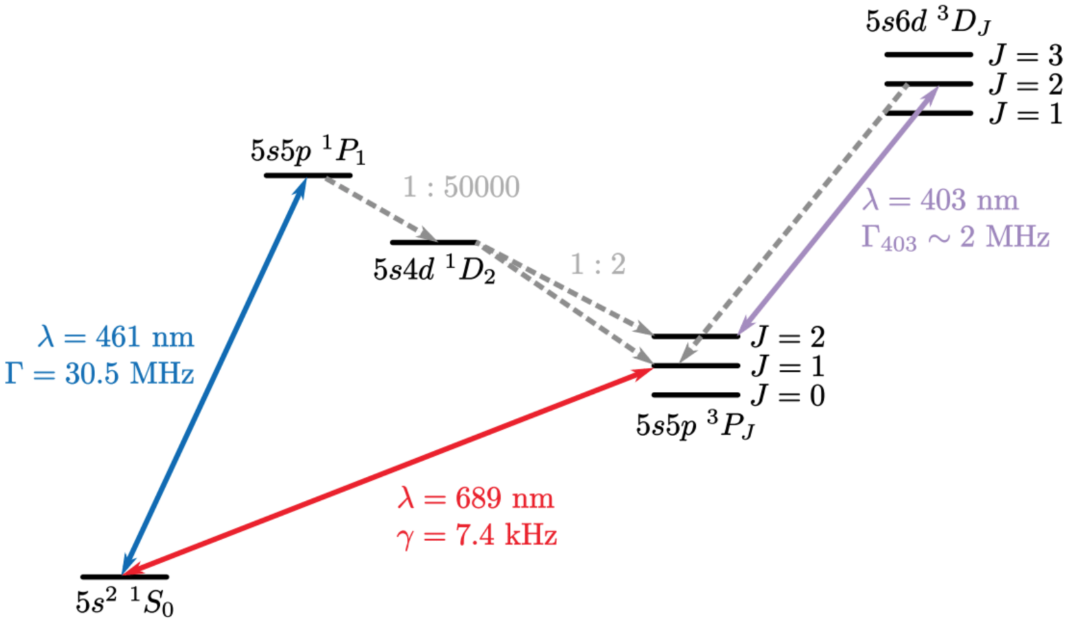


Figure 1.2: Transitions usually used on the experiment. (Blue narrow) $^1S_0 -^1 P_1$ broadband transition to cool the atoms down (Red narrow) This transition is also used to cool the atoms (narrow MOT) and manipulating spins (Purple narrow) This narrow is to repump the atoms that we lose during the process (Dotted narrows) These transitions are due to spontaneous emission

2. PROGRAMMING TO FIT DIFFRACTION PATTERNS

2.1 Band theory : about diffraction [2][3][4]

To understand the experiment of atoms diffraction by a lattice, we need a quick reminder of band theory. We are interested in the motion of quantum particles in a periodic potential. We consider the problem in 1D. The periodic potential of a laser of amplitude V_0 is expressed as

$$V_a(x) = V_0 \sin^2 kx \tag{2}$$

a being the period of the potential such as $V_a(x+a) = V_a(x)$ and $k = \pi/a$ the recoil momentum of the lattice. The hamiltonian of the system is expressed as

$$\hat{H} = \frac{p^2}{2m} + V(\hat{x}) \tag{3}$$

$p = -i\hbar\vec{\nabla}$. We can apply Bloch theorem on this hamiltonian that is invariant by translation. Its eigenstates are

$$\psi_{n,q}(x) = e^{iqx} u_{n,q}(x) \tag{4}$$

where $u_{n,q}(x)$ are periodic function of period a such as $u_q(x+a) = u_q(x)$. q is the quasi-momentum defined in the reciprocal space that defines Bloch states (equation 4). We can rewrite the Schrödinger equation according to the Bloch theorem

$$\left(\frac{(p+\hbar q)^2}{2m} + V_a(x)\right) u_{n,q}(x) = E_n(q) u_{nq}(x) \tag{5}$$

n indexes the eigenstate corresponding to a unique band and m is the mass of the atom. We can express on the plane wave basis

$$u_q(x) = \sum_{j\in\mathbb{Z}} C_j(q) e^{2ijx\frac{\pi}{a}} \tag{6}$$

$$\psi_{n,q} = \sum_{j\in\mathbb{Z}} C_j(q) e^{ix(2jk+q)} \tag{7}$$

with $p = \hbar(2jk + q)$. By combining 6 and 5, and rewritting $V_a(x)$ as $V_a(x) = \frac{V_0}{2} - \frac{V_0}{4}(e^{2ikx} + e^{-2ikx}$ we have for the left part of equation 5

$$\sum_{j\in\mathbb{Z}} E_r \left(2j + \frac{q}{k}\right)^2 C_j(q)e^{2ijx\frac{\pi}{a}} - \frac{V_0}{4}\left(\sum_{j\in\mathbb{Z}} C_j(q)e^{2ijx\frac{\pi}{a}}e^{2ixk} + \sum_{j\in\mathbb{Z}} C_j(q)e^{2ijx\frac{\pi}{a}}e^{-2ixk}\right) \tag{8}$$

$E_r = \frac{\hbar^2 k^2}{2m}$ is the recoil energy. Via changes of variable, the equation 5 is

$$\sum_{j\in\mathbb{Z}} e^{2ijx\frac{\pi}{a}} \left[\left(E_r\left(2j + \frac{q}{k}\right)^2 + \frac{V_0}{2} - E_n(q)\right) C_j(q) - \frac{V_0}{4}(C_{j+1} + C_{j-1})\right] = 0 \tag{9}$$

$$\left[E_r\left(2j + \frac{q}{k}\right)^2 + \frac{V_0}{2E_r}\right] C_j(q) - \frac{V_0}{4E_r}(C_{j+1} + C_{j-1}) = \frac{E_n(q)}{E_r} C_j \tag{10}$$

which can also be writen as

$$\begin{pmatrix} \ddots & & \cdots & & \\ & \frac{\hbar^2(q-2k)^2}{2m} & \frac{-V_0}{4} & 0 & \\ \cdots & \frac{-V_0}{4} & \frac{\hbar q^2}{2m} & \frac{-V_0}{4} & \cdots \\ & 0 & \frac{-V_0}{4} & \frac{\hbar(q^2+2k)}{2m} & \\ & & \cdots & & \ddots \end{pmatrix} \begin{pmatrix} \vdots \\ C_{-1} \\ C_0 \\ C_1 \\ \vdots \end{pmatrix} = E_n(q) \begin{pmatrix} \vdots \\ C_{-1} \\ C_0 \\ C_1 \\ \vdots \end{pmatrix} \tag{11}$$

The terms $-V_0/4$ couple momentum $2jk$ plane waves to plane waves of momentum $2(j \pm 1)k$ which discretize also the position of the atoms in real space.

In practice, we lighten the atoms by the lattice 689 nm during a short time $t = 10\mu$s and we switch it off during a chosen time of flight (ToF) that let them spread into space. By imaging their spread, we can have access to their initial velocity in the lattice and this velocity depends on the depth of the potential wells.

This experiment is exactly the same as diffracting a monochromatic wave on a diffraction grating which prints a phase on the atoms. Here, the presence of the lattice changes the atoms wavefunction of a phase

$$\psi(x,t) = e^{\frac{-iV(x)t}{\hbar}} \psi(x, t = 0) \tag{12}$$

In both cases we obtain a diffraction pattern that is caracterize by the diffracting grating in a case, the optical lattice in the other.

2.2 Programming

The diffraction images are obtained by absorption : we image on camera the shadow of the atoms illuminated by a resonant laser. [5] The untreated images of diffraction that we obtain from the camera look like in 2.1

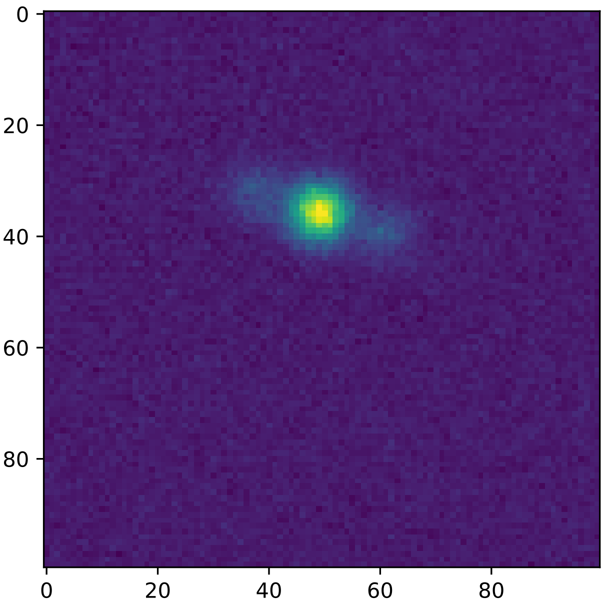


Figure 2.1: Absorption image of the diffracted atoms on a CCD camera (axes represent pixels position). This manipulation is done by illuminating quickly the atoms with an optical lattice. We see that orders -1, 0 and 1 are populated.

where we see the atoms spread on three diffraction orders. The the axes of this picture are positioning the pixels.

The optical density (OD) defined by the light absorbed by the atoms is proportionnal to the atoms number which is the quantity we are interested in.

Beer-Lambert law stipulates for a resonant light of intensity $I(x, y, z)$ that crosses an atomic cloud of density $n(x, y, z)$

$$\frac{\mathrm{d}I(x,y,z)}{I(x,y,z)} = -n(x,y,z)\sigma_{abs}\mathrm{d}z \tag{13}$$

σ_{abs} the atom-light absorption cross-section.

$$OD(x,y) = -\log\left(\frac{I(x,y)}{I_0(x,y)}\right) = \sigma_{abs}\int_{-\infty}^{\infty} n(x,y,z)\mathrm{d}z \tag{14}$$

By integrating 14

$$N = \frac{1}{\sigma_{abs}}\int_{-\infty}^{\infty} OD(x,y)\mathrm{d}x\mathrm{d}y \tag{15}$$

Thanks to the parameters determined by the scipy *curvefit* function, we can express the number of atoms in each gaussian peak as [**5**]

$$N_{fit} = A_{fit}\sqrt{2\pi}\sigma_{fit}\frac{a^2}{\sigma_{abs}} \tag{16}$$

A_{fit} the peak amplitude find by the fit, σ_{fit} proportionnal to the full width at half maximum (FWHM), a the pixel size.
I used the function *curvefit* from scipy

$$scipy.optimize.curve\_fit(f, xdata, ydata, p0 = None, sigma = None...) \tag{17}$$

The first step has been to find the more robust initial parameters to have the better fit. I did it from 1D tabs because of the constraint of the function. I summed all the rotated intensities obtained by the scipy interpolation function *rotate* in x then y axis to create **x_data_1D** and **y_data_1D**. From those tabs I determined the position, maximum amplitude and width of each gaussian peak by the scipy functions $find\_peaks$ and $peak\_widths$. I also had to find the best boundaries to each parameter to converge to the best fit. We see in figure 2.2 an exemple of a fit in x and y directions of a diffraction pattern of three to five diffraction orders. We can say that the simulation fits quite well to the datas. We have $N_{-3} = 303.7$, $N_{-2} = 2055.9$, $N_{-1} = 5400.6$, $N_0 = 7429.3$, $N_1 = 5879.6$, $N_2 = 2058.5$, $N_3 = 646.0$, so a total number of atoms $N_{tot} = 23773.7$.

The final step of this code was to plot the evolution of the atoms number in each peak on time to compare to an evolution population model. The theoretical model is presented in 2.3. I do not have a final experimental figure to present because the curves are not behaving as expected. It comes from the fact that I still need to improve the

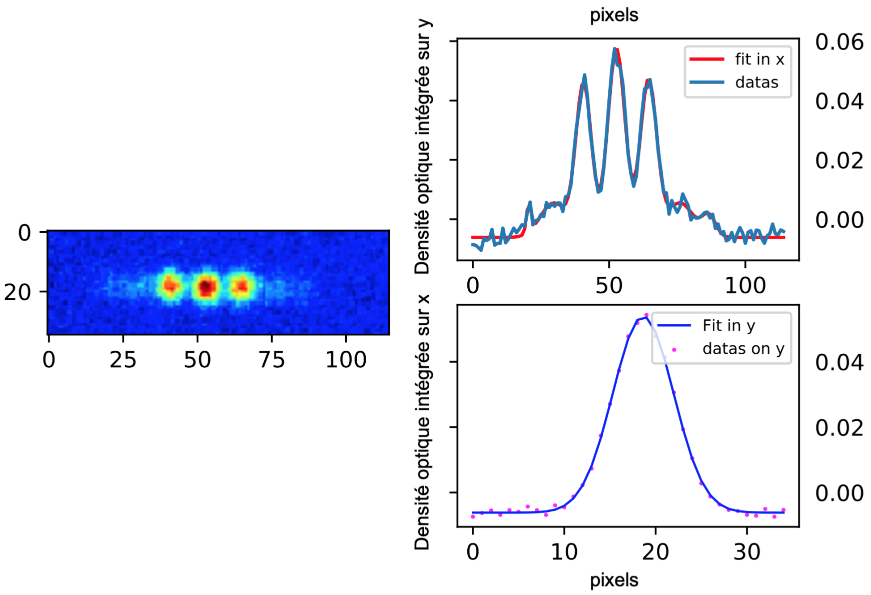


Figure 2.2: Exemple of a fit of a diffraction pattern (left) Diffraction pattern obtained on a CCD camera depending on the pixels position (right up) Plot of the gaussian beams obtained in the left image in x axis (right down) Plot of the same picture on y axis

robustness of my programm because when the higher (or lower) orders are not populated, my programm identify sometimes badly the position of the populated ones. I need to put more restriction on the position of each peak at every picture. They are supposed to stay at the same position because the distance between the orders is imposed by the band theory (cf **Lattice theory 2.1 part**).

Because the apparence of the curves in the theoretical model depends on the lattice depths that we are changing as a variable in the theoretical programm, we can deduce the lattice depth by comparing the two graphs.

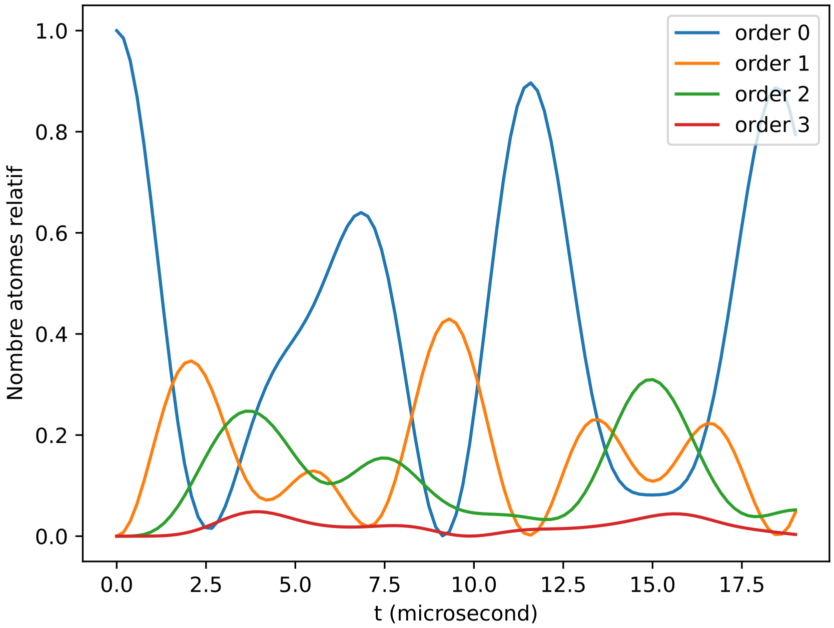


Figure 2.3: Theoretical behaviour of the population evolution of each order of diffraction on time. The appearance of this graph depends on the lattice depth. By comparing the experimental curves obtained with the programm (cf annexes), this parameter will be predictable.

The last part of this project would be to include this programm directly on the experiment to have a live fitting.

3. CAVITY

The second part of my internship consisted in building a cavity to filter a laser at 689 nm. This laser is involved in spin manipulation of strontium atoms by Raman transitions. This process is a two-photon excitation with a strong detuning that minimizes spontaneous emission. Because there is amplified spontaneous emission (ASE) phenomenon in the laser, some spectral components are resonant with the atoms which create spontaneous emission inside the atomic cloud. I have been building the cavity in order to filter this laser to keep only the more powerful component.

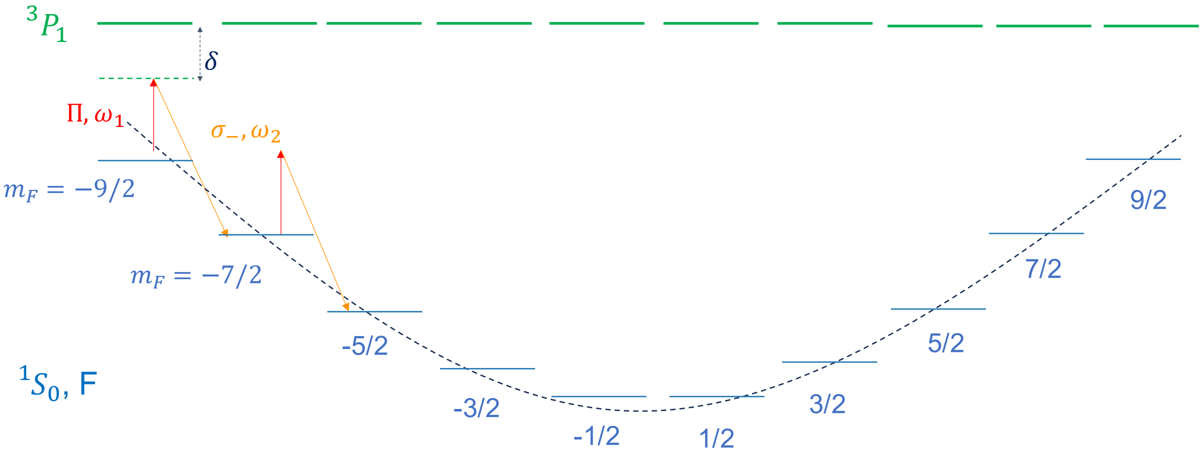


Figure 3.1: Raman transition process. With a π polarized laser, the energy levels are quadraticaly lifted. With a second σ_- polarized laser, added to the previous photon brought by the π laser, the states $|^1S_0, m_F\rangle$ and $|^1S_0, m_F \pm 1\rangle$ are coupled.

3.1 Spin manipulation by Raman spectroscopy

One objective of the experiment is to simulate special quantum states by manipulating spins. To prepare the initial state we do a two photons Raman transition. From a state $|g, F, m_F\rangle$ - that is a degenerate ground state of the total atomic angular momentum F and projection of the total atomic angular momentum m_F - we excite the atom non-resonantly (for exemple $|e, F+1, m_F\rangle$) with a *Dressing laser* in order to avoid spontaneous emission. With a second laser called *Raman laser* we de-excite this atom to $|g, F, m_{F+1}\rangle$. By this process we can choose the state of the atom.
The Raman process on the ten states of strontium is presented in figure 3.1. We do it at detuning $\delta = \omega_{laser} - \omega_{atom}$ in order to avoid spontaneous emission in case of the atom exactly at resonance.
A first Π polarized laser called *Dressing laser* leads the atoms near resonance that couples states $|^1S_0, m_F\rangle$ and $|^3P_1, m_{F'} = m_F\rangle$ only (depending on absorption or emission), according to selection rules. A second σ_- or σ_+ polarized *Raman laser* couples states $|^1S_0, m_F\rangle$ and $|^1S_0, m_F \pm 1\rangle$ with the selection rule $\delta m_F = \pm 1$. The polarization of this laser depends on which state we want to have.

The light-matter interaction is due to dipolar electric transition that makes, at resonance, the atoms oscillate between two states at Rabi frequency $\Omega = -\frac{\vec{D}.\vec{E}}{\hbar}$, $\vec{D}$ the electric dipole operator and $\vec{E}$ the electric field. Because we are off-resonant, we can describe this coupling as a second-order perturbation in case of $\sigma-$ polarization

$$\langle F, m_F \left| V_{eff} \right| F, m_F{}' \rangle = \left| E_\Pi \right| \left| E_{\sigma-} \right| \sum_{|e\rangle} \frac{\left\langle F, m_F \left| \overrightarrow{\varepsilon_\pi} \cdot \vec{D} \right| e \right\rangle \left\langle e \left| \overrightarrow{\epsilon_{\sigma_-}} \cdot \vec{D} \right| F, m_F{}' \right\rangle}{\hbar\delta} \tag{18}$$

where E_π and E_{σ_-} are the polarized electromagnetic field amplitudes, $\vec{\epsilon_\pi}, \vec{\epsilon_{\sigma_-}}$ their polar, $\hbar$ Planck constant. The *Dressing laser* second function is to create a quadratic energy displacement. It is necessary to have a degeneracy splitting besides quadratic because a combination of two lasers at frequency ω_1 and ω_2 need to be resonant only for one transition as illustrated by the second narrow pair on figure 3.1. Otherwise we could not isolate a two-level system in order to have full transfer between those states.
In other words from 18 the quadratic displacement expression is - with A and B constant -

$$V_{eff} \propto A + Bm_F^2 \tag{19}$$

The hamiltonian describing our ten states system in a rotating wave picture - where terms that rotate rapidly are neglected - is given by

$$\hat{H}(\delta) = \frac{\hbar}{2} \begin{pmatrix} -9\delta & \Omega_{-9/2}^{-7/2} & 0 & \dots & 0 & 0 & 0 \\ \Omega_{-9/2}^{-7/2*} & -7\delta & \Omega_{-7/2}^{-5/2} & \dots & 0 & 0 & 0 \\ 0 & \Omega_{-7/2}^{-5/2*} & -5\delta & \ddots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ddots & 5\delta & \Omega_{5/2}^{7/2} & 0 \\ 0 & 0 & 0 & \dots & \Omega_{5/2}^{7/2*} & 7\delta & \Omega_{7/2}^{9/2} \\ 0 & 0 & 0 & \dots & 0 & \Omega_{7/2}^{9/2*} & 9\delta \end{pmatrix} \tag{20}$$

every element is determined by 18. $\Omega_{-9/2}^{-7/2}$ is propotionnal to $\Omega' = \frac{\Omega_\pi \Omega_{\sigma_-}}{\delta}$. $\hat{H}_q$ is the spontaneous emission hamiltonian. For exemple, the coupling by the two lasers of state $|\frac{9}{2}\frac{-9}{2}\rangle$ to $|\frac{9}{2}\frac{-7}{2}\rangle$ is

$$\Omega_{-9/2}^{-7/2} = \langle \frac{9}{2}\frac{-9}{2}|1,0|\frac{9}{2}\frac{-9}{2}\rangle \langle \frac{9}{2}\frac{-9}{2}|1,1|\frac{9}{2}\frac{-7}{2}\rangle \, \Omega' = \frac{6}{11}\Omega' \tag{21}$$

$\langle \frac{9}{2}\frac{-9}{2}|1,0|\frac{9}{2}\frac{-9}{2}\rangle$and $\langle \frac{9}{2}\frac{-9}{2}|1,1|\frac{9}{2}\frac{-7}{2}\rangle$ are the Clebsh-Gordan coefficients.

3.2 Introduction to cavity

A cavity is made of two mirrors that are reflecting the light injected between them. This reflected light interfers with itself and the solutions of Maxwell equations give different amplitude spatial distributions called **Hermite-Gauss modes TEM$_{pq}$** expressed as

$$A_{pq}(r,z) = A_0 \frac{w_0}{w(z)} e^{\frac{-r^2}{w(z)^2}} e^{i\frac{kr^2}{2R(z)}} e^{i\varphi_{pq}(z)} H_p\left(\frac{\sqrt{2}x}{w(z)}\right) H_q\left(\frac{\sqrt{2}y}{w(z)}\right) \tag{22}$$

and visible in figure 3.2. ω_0 is the waist of the beam, $\omega(z)$ its radius at z position, R(z) the curvature radius, k the wavelength number, $\phi_{pq}(z)$ Gouy phase and H_p, H_q are the Hermite polynomes.

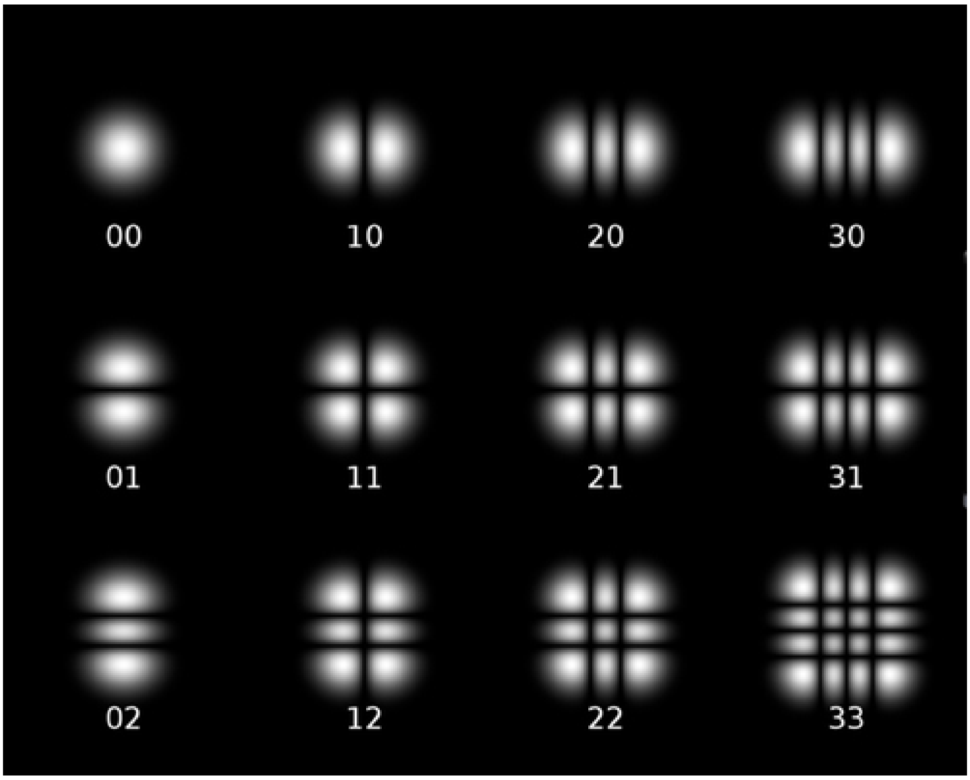


Figure 3.2: Intensity profile of Hermite-Gauss resonant modes TEM$_{pq}$

3.3 Confocal cavity

In a confocal cavity, all the odd modes are resonant at the same frequency f_{odd} and the even ones at f_{even}. I chose to build a confocal cavity to maximize the transmission of it because if the modes are not all at the same frequency, some light is reflected and we lose transmission.

This cavity is made of two curved mirrors with the same curvature R_c, and $d_0 = R_c$ appart. We want to match TEM$_{00}$ mode that is the situation where the beams are overlapping contrarely to the situation presented in figure 3.3. This symmetry argument enables to have a maximum of transmission.

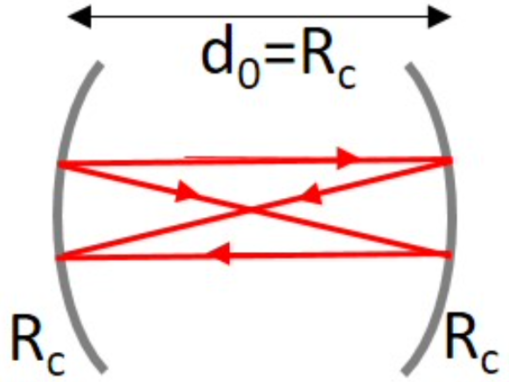


Figure 3.3: Example of how the light behaves on a confocale cavity that is made of two curved mirrors, $d_0 = R_c$ appart, with R_c their curvature.

3.4 Experiment

Before starting to build the cavity, it was important to make sure that the transmission and reflection of the mirrors, individually rates were consistent with the supplier ones. I obtained $T = 1.19 \pm 0.06\%$ where error bar corresponds to fluctuations between different measurements, while the provider gives $T = 1.28\%$. This value is not quite consistent with the expected transmission. It could come from the fact that I was not perfectly centered on the mirror when I did the measure, or the laser that is fluctuating.

The first step of the experiment was to glue the mirrors on their mounts trying to affix them the flattest as possible. It is to avoid the beam to deflect that could imply a clip of the outgoing beam but also introduce aberrations.

One of the mirror was easy to glue because I just had to glue it on a cylindrical mount. I just putted the mirror on a surface and the mount over it with some clue. The other one was more difficult because it had to remain on a piezo - useful to modify the cavity length - itself glued on the mount. As visible in 3.4, I used two affixed rods that had screw at the top. I could modify the position of the mirror by graduated plates. I adjusted the position of the rods in ordrer to be exactly at the extremum of the piezo. From this position, I exactly moved the mirror into the two direction by the distance to which the mirror was centered on the piezo. In this way, the screws on the rods pushed gently the mirror on the piezo surface. After that the mirrors were ready to be assembled with the isolating structure.

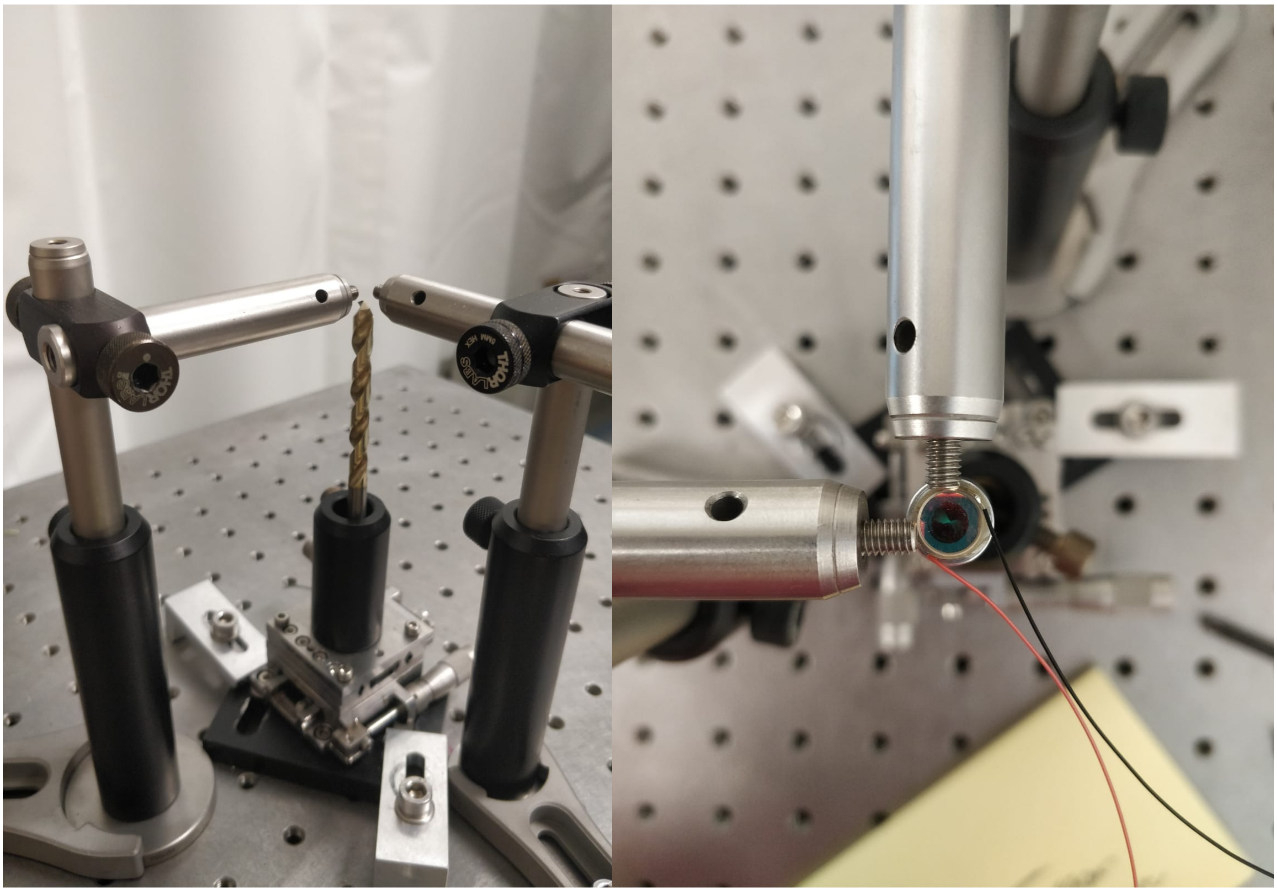


Figure 3.4: Experimental set-up to glue a mirror. The mirror glued to a piezo and a mount is removable by graduated plates. By moving it against the two screws (at the top of affixed rows), we can adjust precisely the position of the mirror on the piezo than we glue then.

The second part of this project was to align the 689 beam with the cavity.

Rough alignement

To be able to visualize light going through the cavity, we need to have the mirrors distant by $d_0 = n\frac{\lambda}{2}$, n an integer and λ the light wavelength. As a start I sweep the cavity length via a piezo that we modulate.

Then I could align the beam by the "walking" technique that consists in adjusting the position of the beam by two mirrors that possess horizontal and vertical knobs. At this stage, I just tried to maximize the transmitted intensity.
One consideration to make was to add a converging lens before the cavity to have a mode-matching closer to the TEM_{00}. This is to account for an imperfect confocal cavity: we want to restrict our beam overlap to a small set of TEM_{pq}. The required focal length f is determined by the waist of the cavity TEM_{00} mode that we want to attempt. This waist is defined by the Rayleigh distance z_R

$$z_R = \frac{\pi w_{TEM_{00}}^2}{\lambda} = \frac{d_0}{2} \tag{23}$$

This last equality is proper to a confocal cavity. From 23 we can express $w_{TEM_{00}}$ as

$$w_{TEM_{00}} = \sqrt{\frac{\lambda d_0}{2\pi}} \tag{24}$$

that gives a theoretical waist size $w_{TEM_{00}} = 74\mu\text{m}$.
First, I do not consider the mirror that is affixed to a diverging lens. In gaussian optics, the relation of the waist w' after a lens of focal lens f and the waist w_0 before the lens [**1**]

$$w' = \frac{f\lambda}{\pi w_0} \tag{25}$$

Because we want to have $w' = w_{TEM_{00}}$ at the center of the cavity, the needed focal lens is

$$f = \frac{w_{TEM_{00}} w_0 \pi}{\lambda} \tag{26}$$

With $w_0 = 500\mu\text{m}$, $f = d_0/2 = 2.5$ cm and $\lambda = 689$ nm, to have the best mode-matching, we need a focal lens $f = 16.87$ cm.
Now if we consider the divergence of the mirror that has a focal lens $f' = \frac{R_c}{1-n}$ with n the glass optical index equals to 1.5, $f' = -10$cm. To estimate it, I have been simulating it in *Gaussianbeam* software [**10**]. For a configuration of a lens of focal lens $f = 15$cm, succeded by a diverging lens with $f' = -9$ cm, the position of the waist changes of less than a millimeter, and the waist changes from 65μm to 87μm. It confirms that we would not affect that much the mode-matching by neglecting the diverging lens.

Overlap the two beams

I can now observe two beams that are reflected and then transmitted by the cavity and I try to end up with a unique beam by overlapping those two.

Adjusting the confocality and transmission

We can now see on an oscilloscope the resonant peaks of the light modes. The cavity is confocal when all the peaks converge in two big peaks that are at f_{odd} and f_{even} frequencies, which is the case in 3.5 while we modulate the size of the piezo by a tension ramp. From that we try to maximize one of the two and verify at the camera this is the TEM$_{00}$.
We are finally maximized on the confocality and the mode-matching : the final step is to lock the cavity on the laser.

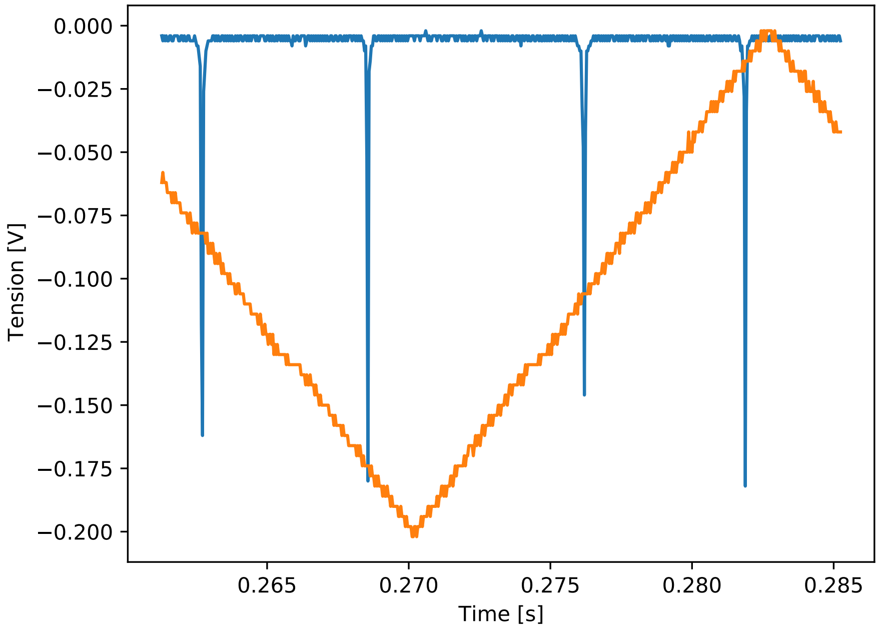


Figure 3.5: (Blue curve) Resonant odd en even peaks observed on an oscilloscope at the output of the cavity. From this configuration we can maximize by a walking one of them. Those two peaks are proper to a confocale cavity (Orange curve) Tension ramp applied to the piezo that sweeps the length of the cavity

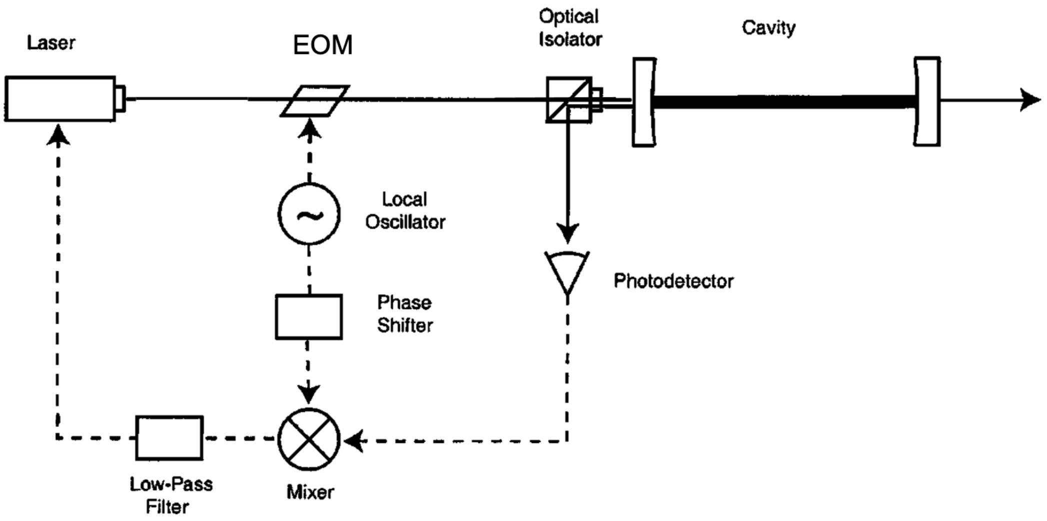


Figure 3.6: Pound-Drever-Hall technique to spectraly purify a laser (in our case). A laser is first modulated by an EOM in order to have sidebands. This modulated signal passes then through a cavity and is reflected and led to a photodector. The signal is then mixed with a RF signal delivered by the same source (local oscillator) as the one for the EOM : we have an error signal that gives the informations to lock the cavity on the laser. The waveplate added to the polarizing cube are there to avoid the signal coming back to the laser.

Pound-Drever-Hall technique for the lock

To lock the cavity resonance on the laser, I used the Pound-Drever-Hall technique that is presented in figure 3.6 [6]. From a laser beam, we first modulate it to have sidebands that give informations about the phase of the beam. Then the modulated beam passes through the cavity where it is reflected. At the entrance the beam is deflected to a photodiode via a waveplate added to a beamsplitter cube, and the signal is mixed with a radio-frequence signal (RF) in order to have a lock-in detection. After that we can tend to the best error signal by changing the RF frequency. Finally we can lock the cavity to the laser thanks to the pid controller that enables a live feedback.

As said in the previous paragraph we need the phase of the reflected beam to be able to adjust its frequency. To do so we first modulate the beam that is transformed from $E = E_0 \exp i\omega t$ to $E = E_0 \exp i(\omega t + \beta sin(\Omega t)$ by an electro-optic modulator (EOM) at

$\Omega = 40MHz$ frequency. We can express this electric field by the Bessel functions J_i

$$E_{\text{inc}} \approx \; E_0 \left[J_0(\beta)e^{i\omega t} + J_1(\beta)e^{i(\omega+\Omega)t} - J_1(\beta)e^{i(\omega-\Omega)t} \right] + ... \tag{27}$$

We see by equation 27 that we have three different beams at different frequencies : one at ω - which is the carrier - and two others at $\omega \pm \Omega$ that are the sidebands. These beams appears as in 3.7 where we see the carrier and the sidebands that are at $\Omega_{exp} = 40$MHz.

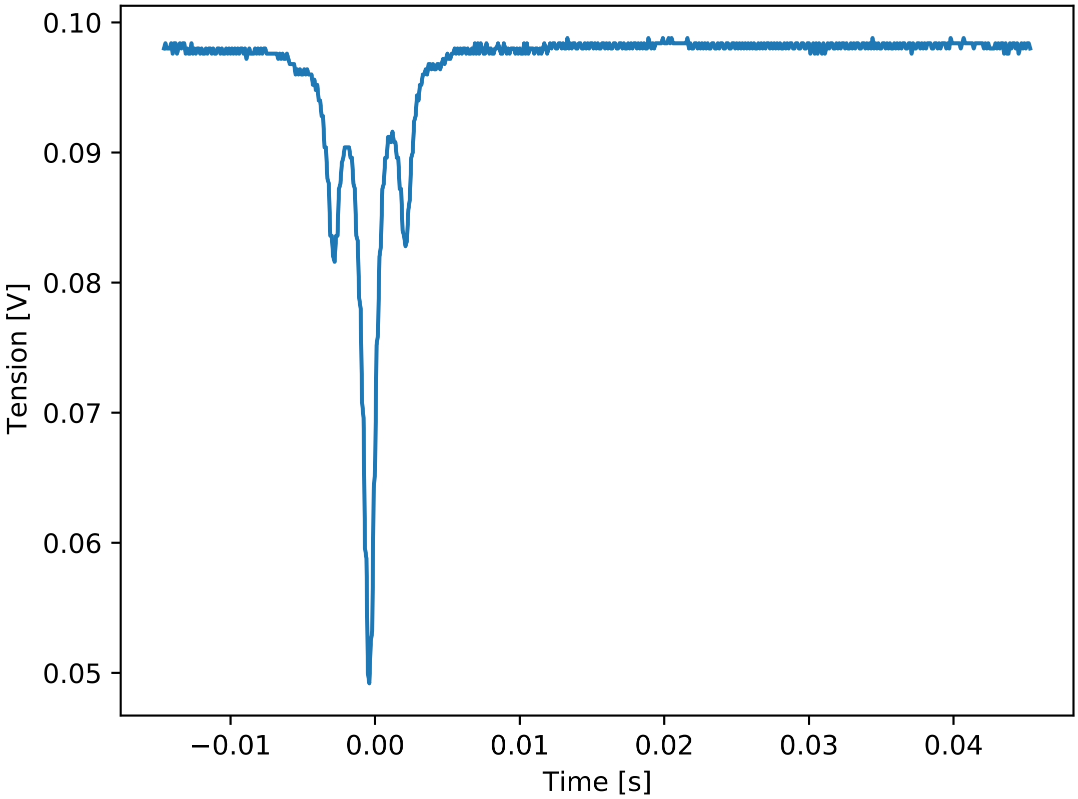


Figure 3.7: Carrier and its sidebands obtained by the modulation of the signal with the EOM. The sidebands are 40MHz apart from the carrier.

Then we measure the error signal that is obtained by a lock-in amplifier. I visualize this signal on a pid controller (figure 3.8) that enables me to control the cavity length continuously. We see on the figure exactly the error signal of the peaks that we see on 3.7. I can lock the cavity resonance modifying the RF frequency and its amplitude by observing this error signal. When I have a flat signal, I know that the cavity is locked.

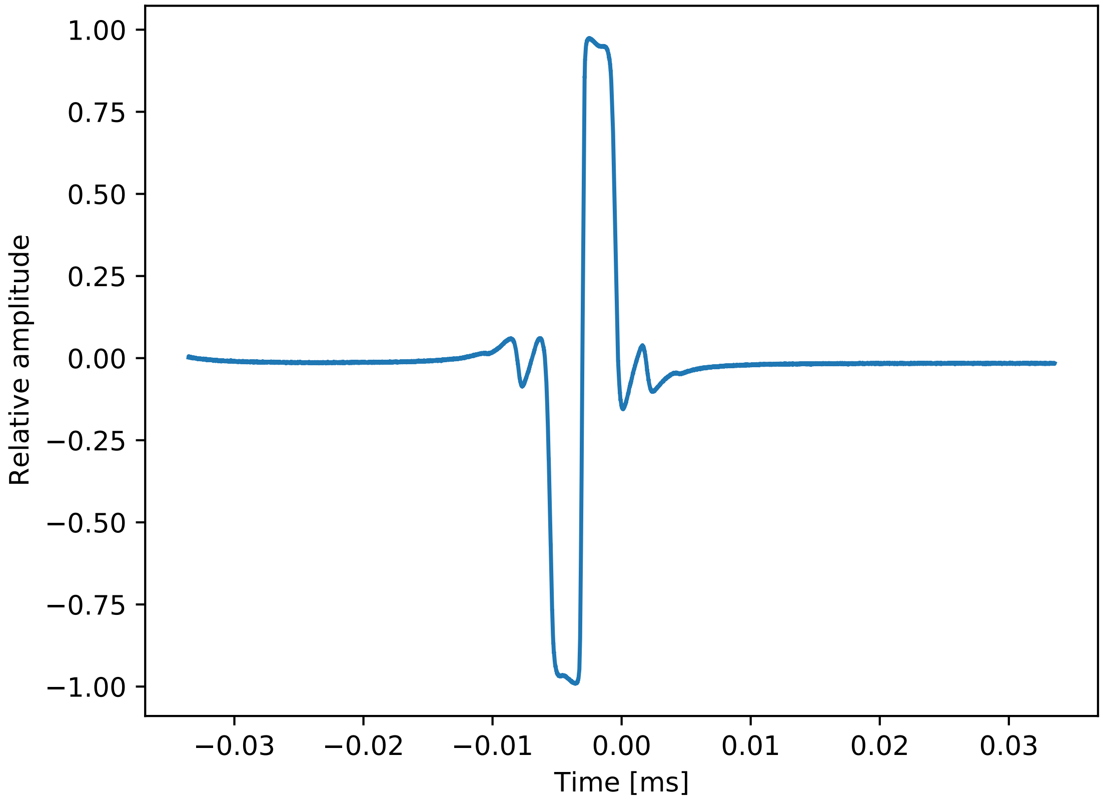


Figure 3.8: Error signal obtained from the lock-in amplifier. From this graph, we can minimize the signal in order to be exactly at the frequency of the laser. We use then a pid controller to stabilize the cavity.

Analysis

Figure 3.7 shows the sidebands that are 40MHz appart from the carrier. The attempted $FSR = \frac{c}{2d_0}$, c the speed of light, is $FSR = 3$GHz.

I measured the transmission of the cavity that is the ratio of the output power on the input one $T = \frac{P_{out}}{P_{in}}$. It gave me $T_{exp}^{cavity} = 34.7\%$ which is surprisingly low ratio. I know that the major origin of it is the lens because I did a bad estimation of its focal length. Because of it the mode-matching is wrong. Also I know that I could improve the injection of the beam in the cavity.

About the finess F of the cavity, we expect from the mirrors with reflection factor $R = 99\%$

so $T = 1\%$ (that we can trust considering the introduction of **Experiment**)

$$F = \frac{\pi\sqrt{R}}{1-R} \tag{28}$$

which is equals to $F = 312$. I measured

FSR	134 ms	134 ms	134 ms
FWHM	640μs	880μs	760μs
F	209 ± 10.45	152.7 ± 7.65	176.3 ± 8.8

the FWHM is the width of the resonant peak. The FSR value is actually the corresponding time. I admit that I have an FSR between two resonant peaks that I visualize in an oscilloscope, while the piezo changes the cavity length during a modulation ramp. It gives me $F = 179 \pm 9$. We can also calculate the reflectivity of the mirrors by the experimental finess that I measured :

$$(28) \iff R^{1/2}(1-R)^{-1} = 2\left(\frac{F}{\pi}\right)^2 \tag{29}$$

We fix $c = F/\pi$. The solution of this equation is

$$R = \frac{2c^2+1}{2c^2} - \frac{1}{2}\sqrt{\frac{4c^2+1}{c^4}} \tag{30}$$

which gives $R = 98.26 \pm 0.03\%$.

Finally with this number and the measured transmission of the mirrors that I introduced in this part, I can estimate the experimental absorption of the mirrors. For $T = 0.0119 \pm 0.006$ and $R = 0.983 \pm 0.0003$, I have $T + R = 0.995 \pm 0.036$. Because we are supposed to have $R + T = 1$, we can say that absorption rate $A = 1 - (R + T) = 0.005 \pm 0.006$.

4. CONCLUSION

The aim of this internship has been to improve the actual experiment in two levels

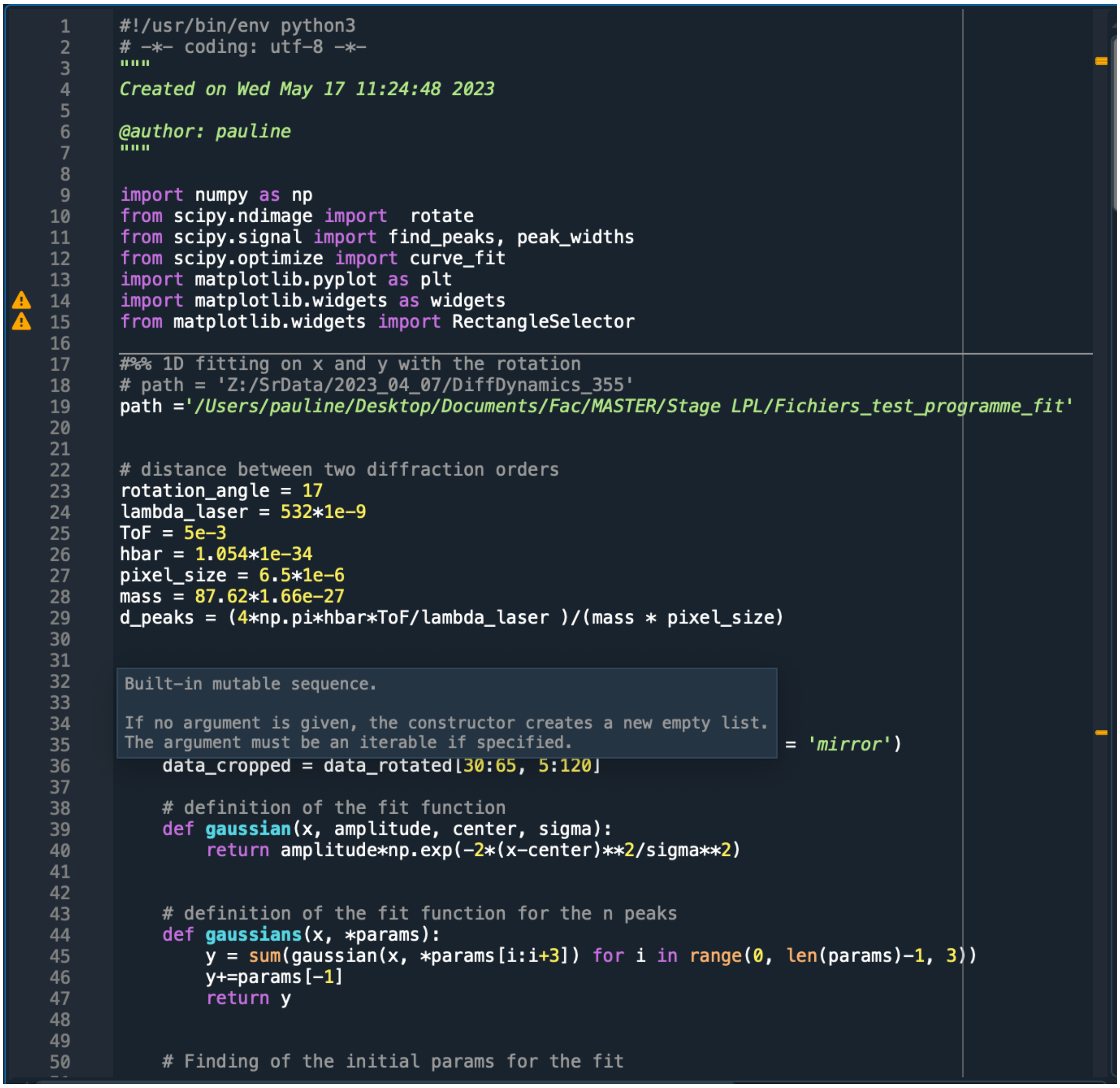
1. I have been building a code that plots the population evolution of each diffraction order of an atomic cloud of strontium 87. By comparing the experimental curves to the theoretical one that has been made by my collegues I will be able to calibrate the lattice depth

2. The other project that I led was to construct a confocal cavity in order to spectraly purify a laser involved in quadratic light-shift of the energy levels of the atoms and at the same time on Raman transitions.

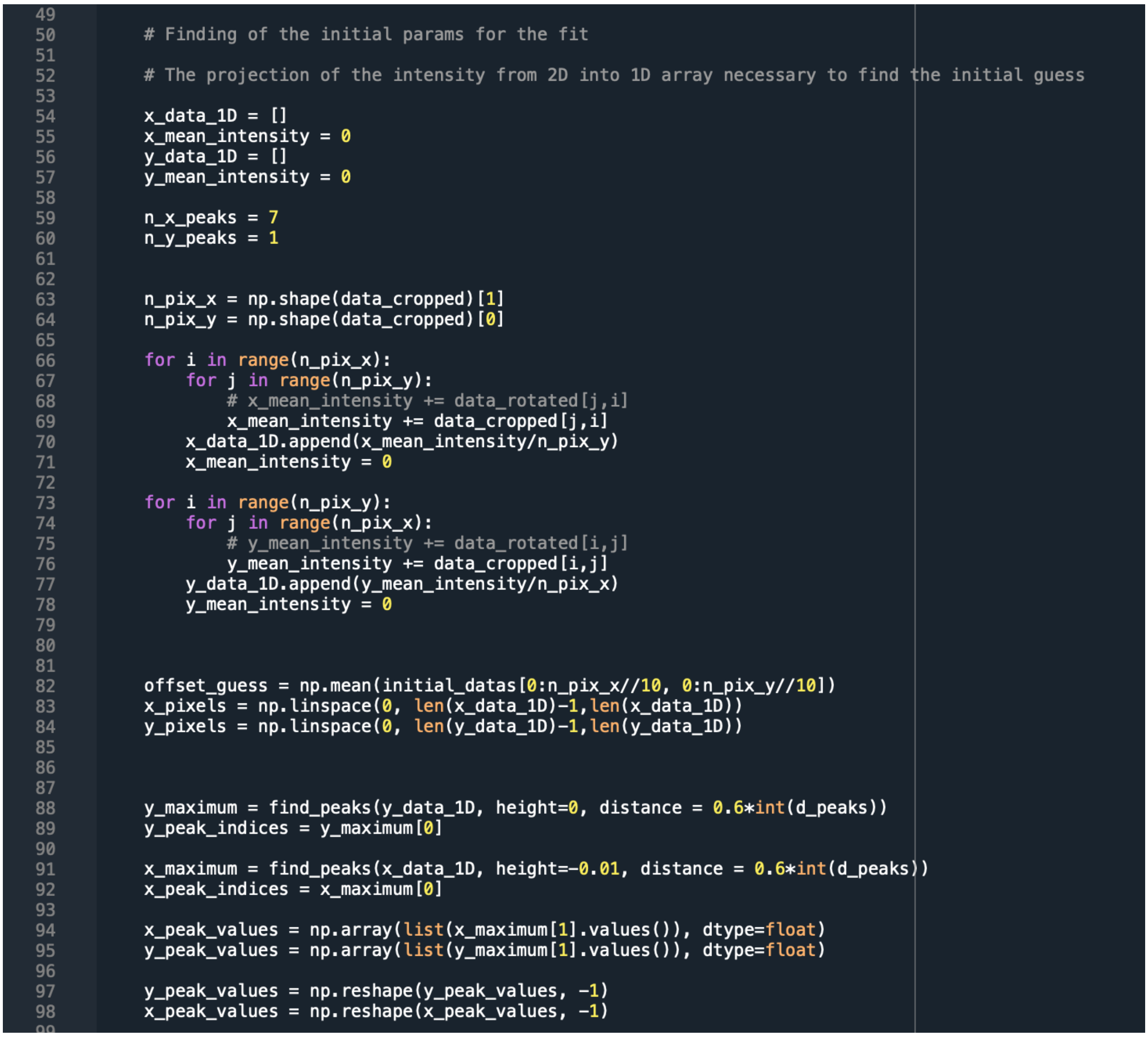
These two works are not finished yet. About the programm, I still have to make it more robust because it does not always identify correctly the position of the populated orders. I will try to take an image as reference where all the orders are populated and from that, affix a position to look at in each simulation. I might improve in that way the accuracy of the code.

For the cavity, I should change the actual lens to a suitable lens in order to improve the mode-match so then the transmission rate. After that I will be able to put this cavity into the experiment table. It should improve the spin manipulation and the actual spontaneous emission that comes with the laser could be the source of an anormal decoherence on the Rabi oscillation between two states that we are now observing.

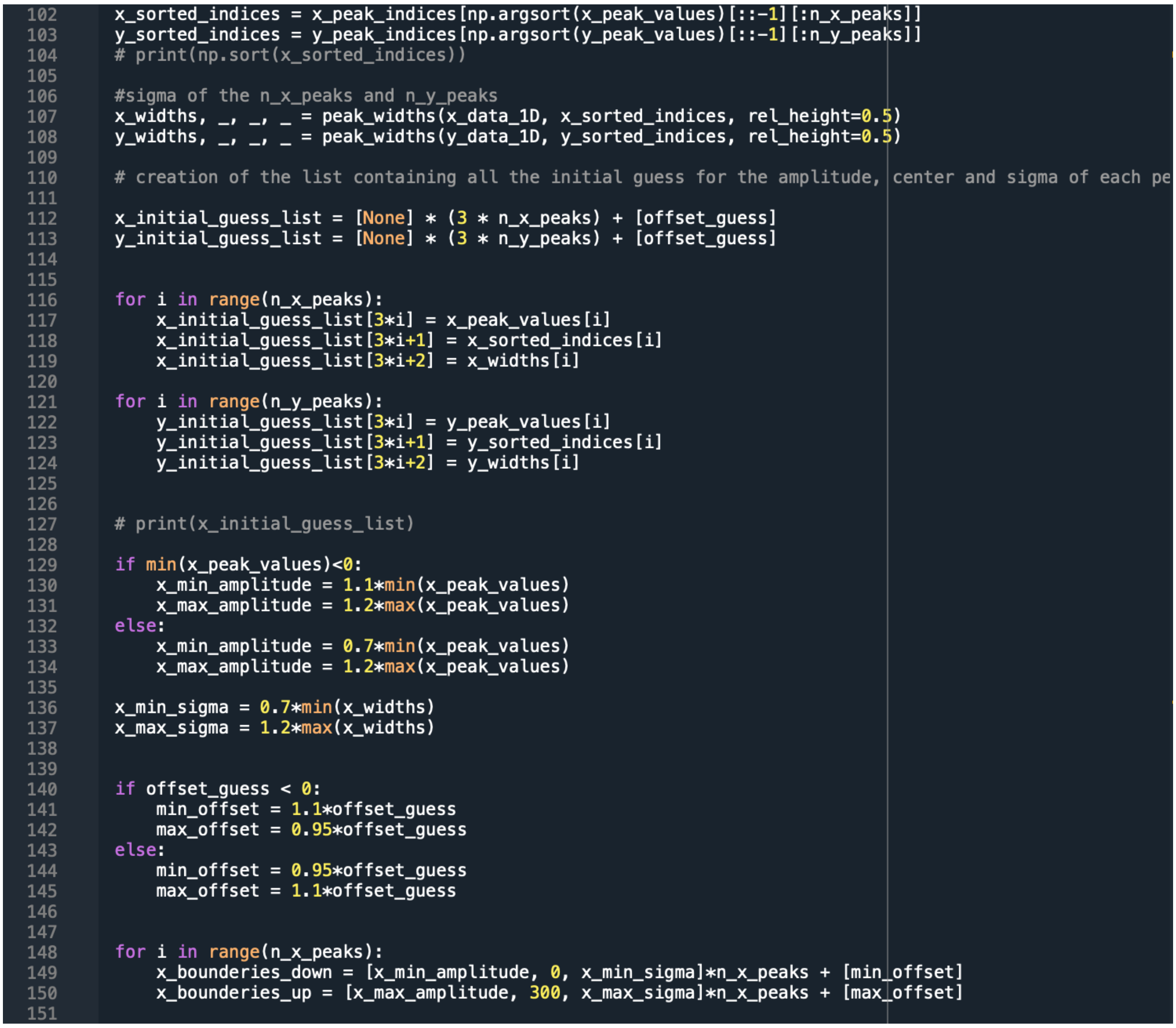
5. SUPPLEMENT



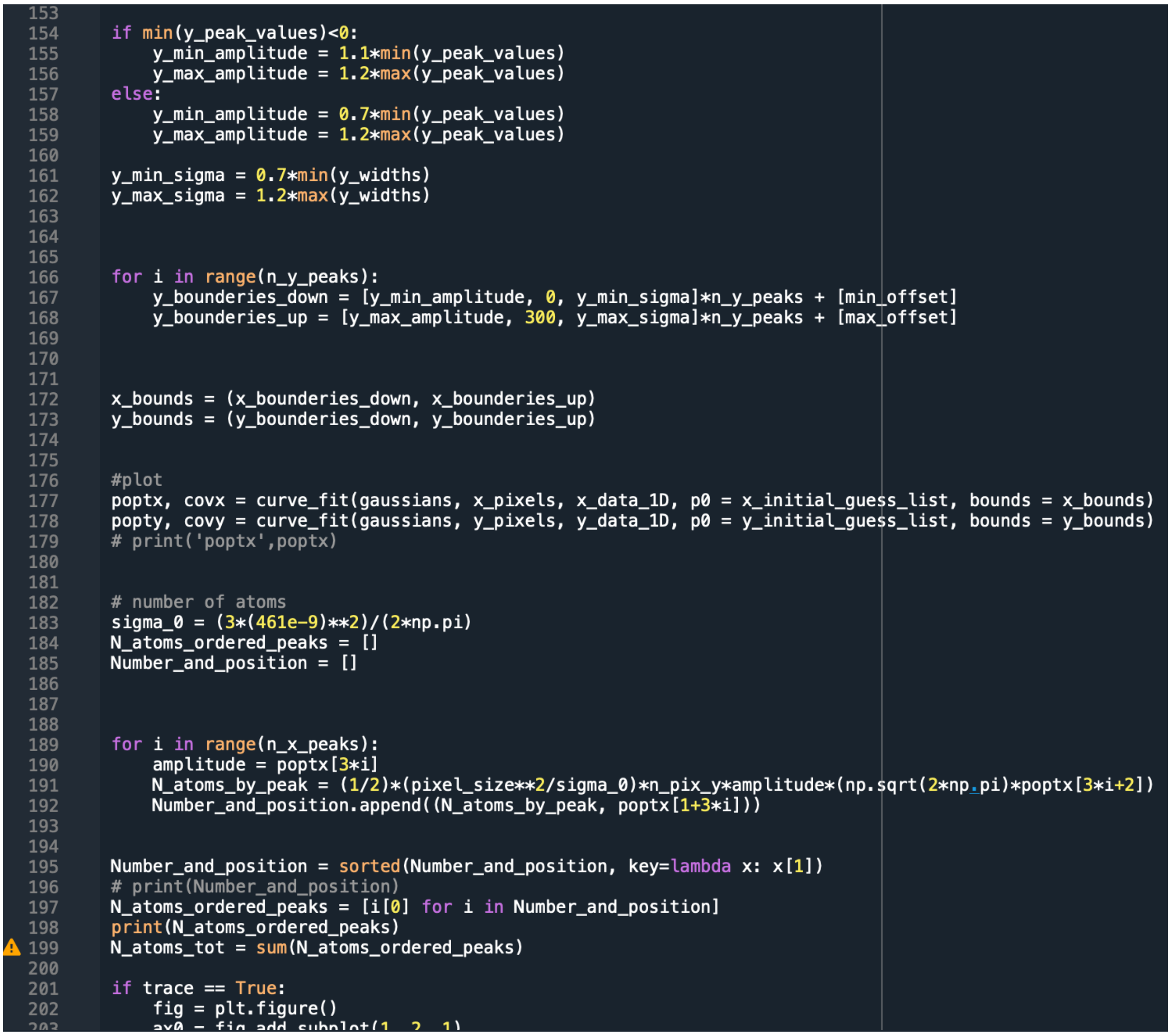
```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed May 17 11:24:48 2023

@author: pauline
"""

import numpy as np
from scipy.ndimage import  rotate
from scipy.signal import find_peaks, peak_widths
from scipy.optimize import curve_fit
import matplotlib.pyplot as plt
import matplotlib.widgets as widgets
from matplotlib.widgets import RectangleSelector

#%% 1D fitting on x and y with the rotation
# path = 'Z:/SrData/2023_04_07/DiffDynamics_355'
path ='/Users/pauline/Desktop/Documents/Fac/MASTER/Stage LPL/Fichiers_test_programme_fit'


# distance between two diffraction orders
rotation_angle = 17
lambda_laser = 532*1e-9
ToF = 5e-3
hbar = 1.054*1e-34
pixel_size = 6.5*1e-6
mass = 87.62*1.66e-27
d_peaks = (4*np.pi*hbar*ToF/lambda_laser )/(mass * pixel_size)


Built-in mutable sequence.

If no argument is given, the constructor creates a new empty list.
The argument must be an iterable if specified.                       = 'mirror')
    data_cropped = data_rotated[30:65, 5:120]

    # definition of the fit function
    def gaussian(x, amplitude, center, sigma):
        return amplitude*np.exp(-2*(x-center)**2/sigma**2)


    # definition of the fit function for the n peaks
    def gaussians(x, *params):
        y = sum(gaussian(x, *params[i:i+3]) for i in range(0, len(params)-1, 3))
        y+=params[-1]
        return y


    # Finding of the initial params for the fit
```



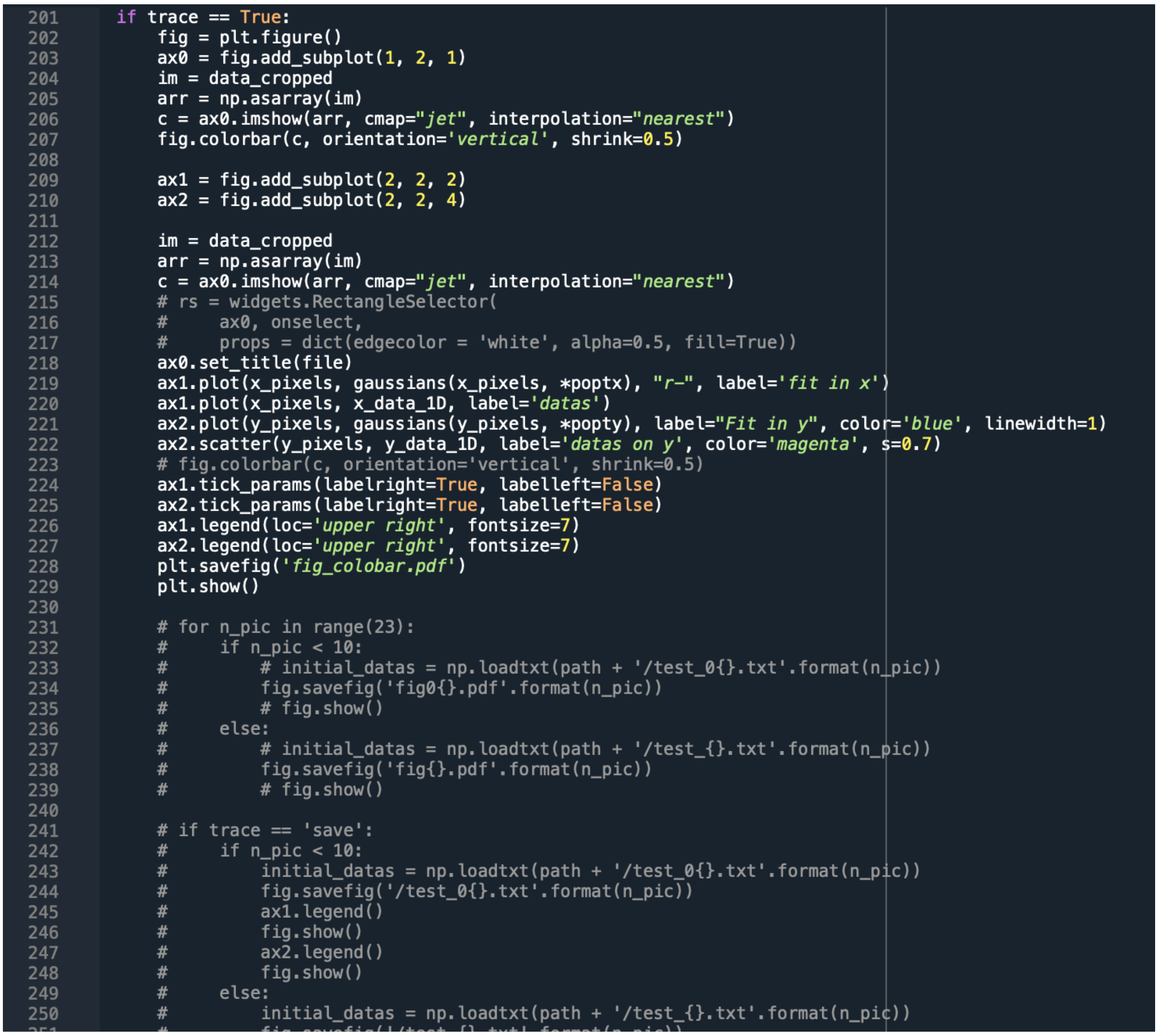
```
    # Finding of the initial params for the fit

    # The projection of the intensity from 2D into 1D array necessary to find the initial guess

    x_data_1D = []
    x_mean_intensity = 0
    y_data_1D = []
    y_mean_intensity = 0

    n_x_peaks = 7
    n_y_peaks = 1


    n_pix_x = np.shape(data_cropped)[1]
    n_pix_y = np.shape(data_cropped)[0]

    for i in range(n_pix_x):
        for j in range(n_pix_y):
            # x_mean_intensity += data_rotated[j,i]
            x_mean_intensity += data_cropped[j,i]
        x_data_1D.append(x_mean_intensity/n_pix_y)
        x_mean_intensity = 0

    for i in range(n_pix_y):
        for j in range(n_pix_x):
            # y_mean_intensity += data_rotated[i,j]
            y_mean_intensity += data_cropped[i,j]
        y_data_1D.append(y_mean_intensity/n_pix_x)
        y_mean_intensity = 0



    offset_guess = np.mean(initial_datas[0:n_pix_x//10, 0:n_pix_y//10])
    x_pixels = np.linspace(0, len(x_data_1D)-1,len(x_data_1D))
    y_pixels = np.linspace(0, len(y_data_1D)-1,len(y_data_1D))



    y_maximum = find_peaks(y_data_1D, height=0, distance = 0.6*int(d_peaks))
    y_peak_indices = y_maximum[0]

    x_maximum = find_peaks(x_data_1D, height=-0.01, distance = 0.6*int(d_peaks))
    x_peak_indices = x_maximum[0]

    x_peak_values = np.array(list(x_maximum[1].values()), dtype=float)
    y_peak_values = np.array(list(y_maximum[1].values()), dtype=float)

    y_peak_values = np.reshape(y_peak_values, -1)
    x_peak_values = np.reshape(x_peak_values, -1)
```



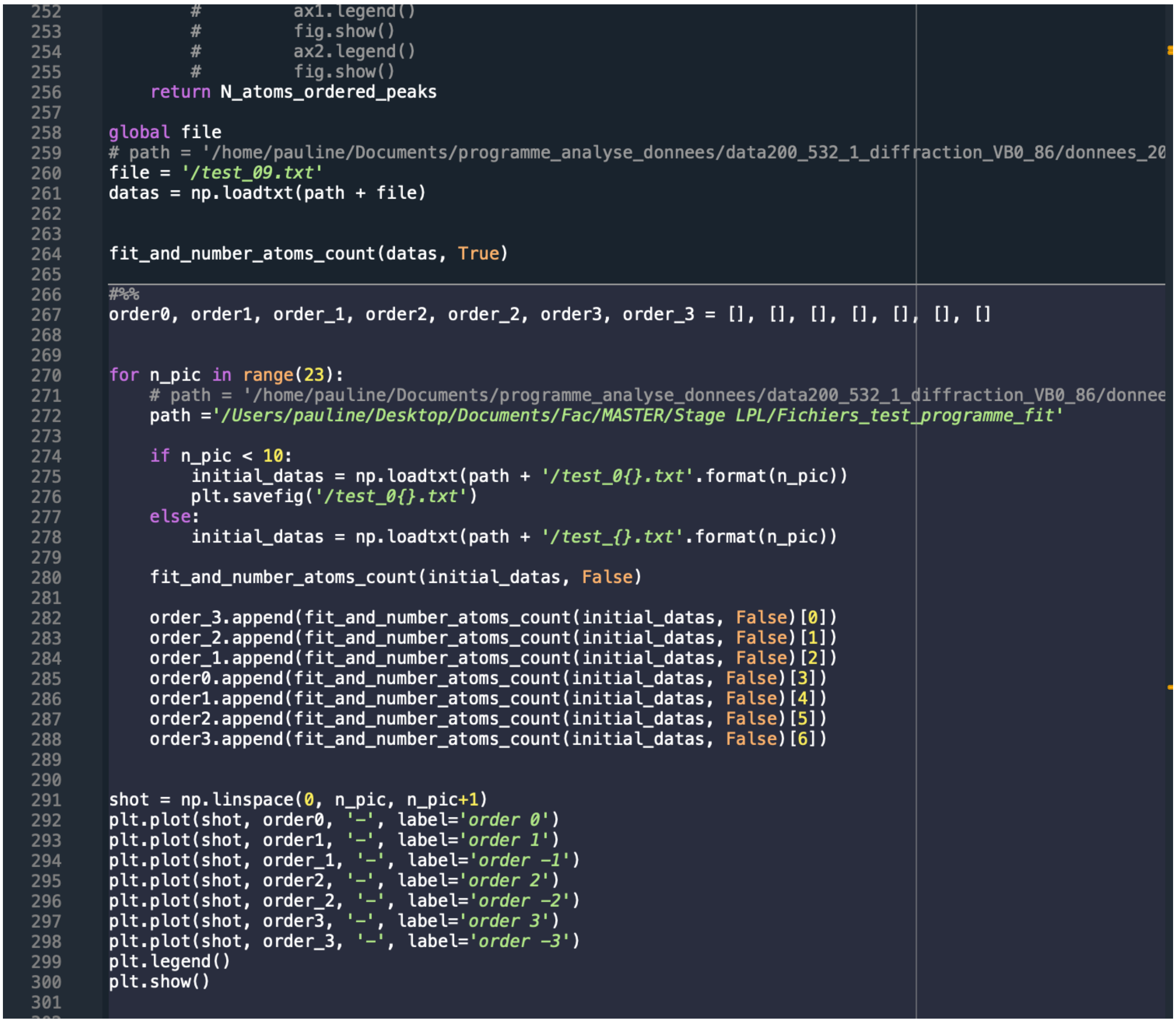
```
x_sorted_indices = x_peak_indices[np.argsort(x_peak_values)[::-1][:n_x_peaks]]
y_sorted_indices = y_peak_indices[np.argsort(y_peak_values)[::-1][:n_y_peaks]]
# print(np.sort(x_sorted_indices))

#sigma of the n_x_peaks and n_y_peaks
x_widths, _, _, _ = peak_widths(x_data_1D, x_sorted_indices, rel_height=0.5)
y_widths, _, _, _ = peak_widths(y_data_1D, y_sorted_indices, rel_height=0.5)

# creation of the list containing all the initial guess for the amplitude, center and sigma of each pe

x_initial_guess_list = [None] * (3 * n_x_peaks) + [offset_guess]
y_initial_guess_list = [None] * (3 * n_y_peaks) + [offset_guess]


for i in range(n_x_peaks):
    x_initial_guess_list[3*i] = x_peak_values[i]
    x_initial_guess_list[3*i+1] = x_sorted_indices[i]
    x_initial_guess_list[3*i+2] = x_widths[i]

for i in range(n_y_peaks):
    y_initial_guess_list[3*i] = y_peak_values[i]
    y_initial_guess_list[3*i+1] = y_sorted_indices[i]
    y_initial_guess_list[3*i+2] = y_widths[i]


# print(x_initial_guess_list)

if min(x_peak_values)<0:
    x_min_amplitude = 1.1*min(x_peak_values)
    x_max_amplitude = 1.2*max(x_peak_values)
else:
    x_min_amplitude = 0.7*min(x_peak_values)
    x_max_amplitude = 1.2*max(x_peak_values)

x_min_sigma = 0.7*min(x_widths)
x_max_sigma = 1.2*max(x_widths)


if offset_guess < 0:
    min_offset = 1.1*offset_guess
    max_offset = 0.95*offset_guess
else:
    min_offset = 0.95*offset_guess
    max_offset = 1.1*offset_guess


for i in range(n_x_peaks):
    x_bounderies_down = [x_min_amplitude, 0, x_min_sigma]*n_x_peaks + [min_offset]
    x_bounderies_up = [x_max_amplitude, 300, x_max_sigma]*n_x_peaks + [max_offset]

```



```

if min(y_peak_values)<0:
    y_min_amplitude = 1.1*min(y_peak_values)
    y_max_amplitude = 1.2*max(y_peak_values)
else:
    y_min_amplitude = 0.7*min(y_peak_values)
    y_max_amplitude = 1.2*max(y_peak_values)

y_min_sigma = 0.7*min(y_widths)
y_max_sigma = 1.2*max(y_widths)



for i in range(n_y_peaks):
    y_bounderies_down = [y_min_amplitude, 0, y_min_sigma]*n_y_peaks + [min_offset]
    y_bounderies_up = [y_max_amplitude, 300, y_max_sigma]*n_y_peaks + [max_offset]



x_bounds = (x_bounderies_down, x_bounderies_up)
y_bounds = (y_bounderies_down, y_bounderies_up)


#plot
poptx, covx = curve_fit(gaussians, x_pixels, x_data_1D, p0 = x_initial_guess_list, bounds = x_bounds)
popty, covy = curve_fit(gaussians, y_pixels, y_data_1D, p0 = y_initial_guess_list, bounds = y_bounds)
# print('poptx',poptx)


# number of atoms
sigma_0 = (3*(461e-9)**2)/(2*np.pi)
N_atoms_ordered_peaks = []
Number_and_position = []



for i in range(n_x_peaks):
    amplitude = poptx[3*i]
    N_atoms_by_peak = (1/2)*(pixel_size**2/sigma_0)*n_pix_y*amplitude*(np.sqrt(2*np.pi)*poptx[3*i+2])
    Number_and_position.append((N_atoms_by_peak, poptx[1+3*i]))


Number_and_position = sorted(Number_and_position, key=lambda x: x[1])
# print(Number_and_position)
N_atoms_ordered_peaks = [i[0] for i in Number_and_position]
print(N_atoms_ordered_peaks)
N_atoms_tot = sum(N_atoms_ordered_peaks)

if trace == True:
    fig = plt.figure()
    ax0 = fig.add_subplot(1, 2, 1)
```



```
    if trace == True:
        fig = plt.figure()
        ax0 = fig.add_subplot(1, 2, 1)
        im = data_cropped
        arr = np.asarray(im)
        c = ax0.imshow(arr, cmap="jet", interpolation="nearest")
        fig.colorbar(c, orientation='vertical', shrink=0.5)

        ax1 = fig.add_subplot(2, 2, 2)
        ax2 = fig.add_subplot(2, 2, 4)

        im = data_cropped
        arr = np.asarray(im)
        c = ax0.imshow(arr, cmap="jet", interpolation="nearest")
        # rs = widgets.RectangleSelector(
        #     ax0, onselect,
        #     props = dict(edgecolor = 'white', alpha=0.5, fill=True))
        ax0.set_title(file)
        ax1.plot(x_pixels, gaussians(x_pixels, *poptx), "r-", label='fit in x')
        ax1.plot(x_pixels, x_data_1D, label='datas')
        ax2.plot(y_pixels, gaussians(y_pixels, *popty), label="Fit in y", color='blue', linewidth=1)
        ax2.scatter(y_pixels, y_data_1D, label='datas on y', color='magenta', s=0.7)
        # fig.colorbar(c, orientation='vertical', shrink=0.5)
        ax1.tick_params(labelright=True, labelleft=False)
        ax2.tick_params(labelright=True, labelleft=False)
        ax1.legend(loc='upper right', fontsize=7)
        ax2.legend(loc='upper right', fontsize=7)
        plt.savefig('fig_colobar.pdf')
        plt.show()

        # for n_pic in range(23):
        #     if n_pic < 10:
        #         # initial_datas = np.loadtxt(path + '/test_0{}.txt'.format(n_pic))
        #         fig.savefig('fig0{}.pdf'.format(n_pic))
        #         # fig.show()
        #     else:
        #         # initial_datas = np.loadtxt(path + '/test_{}.txt'.format(n_pic))
        #         fig.savefig('fig{}.pdf'.format(n_pic))
        #         # fig.show()

        # if trace == 'save':
        #     if n_pic < 10:
        #         initial_datas = np.loadtxt(path + '/test_0{}.txt'.format(n_pic))
        #         fig.savefig('/test_0{}.txt'.format(n_pic))
        #         ax1.legend()
        #         fig.show()
        #         ax2.legend()
        #         fig.show()
        #     else:
        #         initial_datas = np.loadtxt(path + '/test_{}.txt'.format(n_pic))
        #         fig.savefig('/test_{}.txt'.format(n_pic))
```



```
        #         ax1.legend()
        #         fig.show()
        #         ax2.legend()
        #         fig.show()
    return N_atoms_ordered_peaks

global file
# path = '/home/pauline/Documents/programme_analyse_donnees/data200_532_1_diffraction_VB0_86/donnees_20
file = '/test_09.txt'
datas = np.loadtxt(path + file)


fit_and_number_atoms_count(datas, True)

#%%
order0, order1, order_1, order2, order_2, order3, order_3 = [], [], [], [], [], [], []


for n_pic in range(23):
    # path = '/home/pauline/Documents/programme_analyse_donnees/data200_532_1_diffraction_VB0_86/donnee
    path ='/Users/pauline/Desktop/Documents/Fac/MASTER/Stage LPL/Fichiers_test_programme_fit'

    if n_pic < 10:
        initial_datas = np.loadtxt(path + '/test_0{}.txt'.format(n_pic))
        plt.savefig('/test_0{}.txt')
    else:
        initial_datas = np.loadtxt(path + '/test_{}.txt'.format(n_pic))

    fit_and_number_atoms_count(initial_datas, False)

    order_3.append(fit_and_number_atoms_count(initial_datas, False)[0])
    order_2.append(fit_and_number_atoms_count(initial_datas, False)[1])
    order_1.append(fit_and_number_atoms_count(initial_datas, False)[2])
    order0.append(fit_and_number_atoms_count(initial_datas, False)[3])
    order1.append(fit_and_number_atoms_count(initial_datas, False)[4])
    order2.append(fit_and_number_atoms_count(initial_datas, False)[5])
    order3.append(fit_and_number_atoms_count(initial_datas, False)[6])


shot = np.linspace(0, n_pic, n_pic+1)
plt.plot(shot, order0, '-', label='order 0')
plt.plot(shot, order1, '-', label='order 1')
plt.plot(shot, order_1, '-', label='order -1')
plt.plot(shot, order2, '-', label='order 2')
plt.plot(shot, order_2, '-', label='order -2')
plt.plot(shot, order3, '-', label='order 3')
plt.plot(shot, order_3, '-', label='order -3')
plt.legend()
plt.show()
```

REFERENCES

1. *PierreBataille-PhD.pdf* fr. https://www.dropbox.com/s/v1ha7exoi8ng1oz/PierreBataille-PhD.pdf?dl=0 (2023).

2. Litvinov, A. Manipulation of the nuclear spin states of 87Sr in degenerate SU(N)-symmetric Fermi gases. en.

3. *Réseaux optiques : les principes de base | Collège de France* fr. Aug. 2012. https://www.college-de-france.fr/fr/agenda/cours/des-cages-de-lumiere-pour-les-atomes-la-physique-des-pieges-et-des-reseaux-optiques/reseaux-optiques-les-principes-de-base (2023).

4. Gerbier, F. Quantum gases in optical lattices. en.

5. Pyragius, T. *Developing and building an absorption imaging system for Ultracold Atoms* arXiv:1209.3408 [physics]. Sept. 2012. http://arxiv.org/abs/1209.3408 (2023).

6. Black, E. D. An introduction to Pound–Drever–Hall laser frequency stabilization. en. *American Journal of Physics* **69,** 79–87. ISSN: 0002-9505, 1943-2909. https://pubs.aip.org/aapt/ajp/article/69/1/79-87/1055569 (2023) (Jan. 2001).