Numerical simulations of magneto-optical traps for Ytterbium atoms

Onno Sietse Martin Smit

Bachelorarbeit in Physik
angefertigt im Institut für Angewandte Physik

vorgelegt der
Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität
Bonn

May 2024

I hereby declare that this thesis was formulated by myself and that no sources or tools other than those cited were used.

Bonn, 07.05.2024 .....
Date

.....
Signature

1. Gutachter: Prof. Dr. Sebastian Hofferberth
2. Gutachter: Prof. Dr. Simon Stellmer

Acknowledgements

I would like to thank Sebastian and Xin for allowing me to work on this topic and giving me guidance throughout. I would further like to thank the rest of the YQO group, Eduardo, Ludwig, Tangi, Thilina and Wolfgang for their help with my project.

Contents

1 Introduction **1**

2 Theory **2**
2.1 Concepts of light-atom interaction and laser cooling . . . 2
2.1.1 Lasercooling and magnetic optical traps . . . 2
2.1.2 Optical Bloch equations and forces . . . 4
2.2 Different models for calculation of the force on the atoms . . . 8
2.2.1 Heuristic model . . . 8
2.2.2 Optical bloch equations . . . 8
2.2.3 Rate equation model . . . 10
2.2.4 Momentum diffusion . . . 10
2.2.5 Differences between the models . . . 11

3 Description of the experimental setup **13**

4 PyLCP: A python package for computing laser cooling physics, description and analysis **16**
4.1 PyLCP implementation . . . 16
4.2 Simplifications made by PyLCP . . . 16
4.3 Units . . . 17
4.4 Timestep comparisons . . . 17
4.5 Saturation intensity comparisons . . . 18
4.6 Model comparison . . . 20
4.7 Runtime improvement . . . 22
4.8 conclusion . . . 22

5 Implementation of the experimental setup **23**
5.1 Dispenser sampling . . . 23
5.2 2D MOT . . . 25
5.3 Pushbeam . . . 28
5.4 Tube . . . 30
5.5 3D MOT . . . 30

6 Summary and outlook **33**

A Attempts at parallelization of the heuristic equation **34**

B Additional pictures of the setup 36

C Additional plots for the simulations 37

D Additional scans 42

Bibliography 47

List of Figures 49

List of Tables 51

CHAPTER 1

Introduction

Since the proposal of the dipole blockade mechanism [1] some 20 years ago and the subsequent demonstration of nonlinearity in a magnetic optical trap [2], Rydberg atoms, with their large cross-sections and longer lifetimes, have come to advance quantum information technology and both understanding of quantum many body systems and quantum optics.

To explore rydberg physics one needs atomic clouds of very low temperature with as many atoms as possible. In order to achieve these particle clouds, we can turn to lasercooling and magnetic optical traps (MOTs).

The YQO group explores Rydberg physics using Ytterbium as their medium. Ytterbium, with its two valence electrons, not only offers a broad singlet-singlet transitions but also a narrow 182 kHz inter-combination transition, allowing for a lower Doppler temperature in the trap. It also however presents additional challenges to overcome, as it requires a more complex experimental setup to overcome the low vapour pressure of the element.

The group has build an experimental setup consisting of a 2D MOT vacuum chamber and a 3D MOT science chamber and have taken to optimizing the variables of their experimental setup.

This project will use the python library 'PyLCP: A python package for computing laser cooling physics' [3] to simulate particles in the experiment, in order to eventually be able to give theoretical predictions to be compared with the optimization measurements done by the group.
PyLCP allows for the usage of different models with differing precisions, which will be explored in this thesis. Additionally i will analyse PyLCP to be able to make changes and setup our simulations. Following this i will continue by simulating the setup as used in the group and compare my findings to experimental results as found by the group. Finally i will give an outlook for future use of the library.

CHAPTER 2

Theory

2.1 Concepts of light-atom interaction and laser cooling

To be able to understand the physical background of the simulations, laser cooling and magnetic optical traps need to be understood. Therefore this chapter will explain the basic theory of magnetic optical trapping and laser cooling. In this I will mention the concept of absorption/emission, the Dopplereffect and Zeeman splitting. For a more detailed look at these, one can read p. 161, chapter 5, chapter 7 and chapter 8 of [4].

2.1.1 Lasercooling and magnetic optical traps

Laser cooling and trapping techniques rely on forces arising from the absorption and reemission of light by the atoms.
An atom can absorb a photon, if it has an energy $\hbar\omega_l$ close to an atomic transition in the atom, that is the energy difference between two electronic levels in the atom. If the photon has the same energy as the transition, it is at resonance. The closer a photon to resonance is the more likely it is to be absorbed.
In this process, not only the energy of the photon is transferred to the atom, but also its momentum. An atom, having absorbed such a photon, is then raised into its exited state.
In addition to this process an exited atom can also fall back to its ground state by sending out a photon either spontaneously or induced by a photon with the energy of the transition.
The stimulated emission applies the momentum of the incoming photon to the atom. The spontaneous emission gives the atom a momentum kick in random direction. Thus, over many emission-absorption cycles, the momentum kicks from spontaneous emission cancel out, while the momentum gained from stimulated emission and absorption is directional, creating a force along the direction the photons travel.
These forces will be explored more indepth in the next chapter 2.1.2.

Lasers with highly coherent and monochromatic light are used to address the atomic transitions. They propagate as Gaussian beams with the intensity profile given as:

$$I(r) = I_0 \cdot e^{-2(r/w)^2} \tag{2.1}$$

$$I_0 = \frac{2P_0}{\pi w_0^2} \tag{2.2}$$

where P_0 is the power of the beam at the center of the beam at its point of focus and w_0 is the $1/e^2$ width there.

When such a laser is addressing an atomic transition, the Doppler effect can be used to create a damping force on the atom absorbing the light.
The atom, counter-propagating to the beam, sees the wave fronts of the beam in quicker succession from its perspective. This causes it to experience the light with a blue shift in frequency of $\omega_{d,blue} = \vec{k} \cdot \vec{v}$. A laser that has been red-detuned, I.e. shifted to a lower frequency than resonance, will compensate for this effect. The atom experiences light closer to resonance again, and thus experiences a force along the direction of the beam it is counter-propagating, slowing it down.
An atom co-propagating with the beam experiences a red frequency shift $\omega_{d,red} = -\vec{k} \cdot \vec{v}$ and so the red-detuned laser beam is very far from resonance and exerts very little or no damping force on the atom. However, a blue-detuned laser could be used to accelerate such co-propagating particles.
A red-detuned laser thus creates a cooling force, while a blue-detuned laser creates a heating force.

The damping force alone is not enough to trap atoms. For this magnetic fields need to included into the process.
Due to the Zeeman effect, the magnetic sublevels in an atom shift with the magnetic field as

$$\Delta E_m = \mu_B \cdot m \cdot B \tag{2.3}$$

where μ_B is the Bohrsches magneton and B is the local magnetic field at which the atom is located. The transitions Δm can only be driven by light of certain polarisation. Specifically for right and left polarized light $\sigma^{\pm}$ and linear polarized light π:

$$\Delta m = \begin{cases} +1 & \text{for } \sigma^+ \\ -1 & \text{for } \sigma^- \\ 0 & \text{for } \pi \end{cases} \tag{2.4}$$

Using a magnetic field with a linear gradient and B = 0 at the centre, and laser beams with σ^+/σ^- polarisation as in the figure 2.1, we can build a trap that pushes atoms towards the centre.
Due to the changing magnetic field the atoms on the right side of the MOT are closer to resonance with the σ^- beam than with the σ^+ beam. This means that the atoms are much more likely to absorb a photon coming from the right and will experience an overall force to the left.
On the opposite side, atoms will be more on resonance with the σ^+ beam thus creating a force to the right. This creates an overall restoring force towards the centre of the beam.
If the beams are now also red-detuned we create a damping force that cools the atoms. This combination is what creates a MOT.

So far, this explanation is for a one-dimensional MOT. A MOT can be extended by adding several of these one-dimensional setups in the spatial directions in which the atoms need to be cooled and trapped.

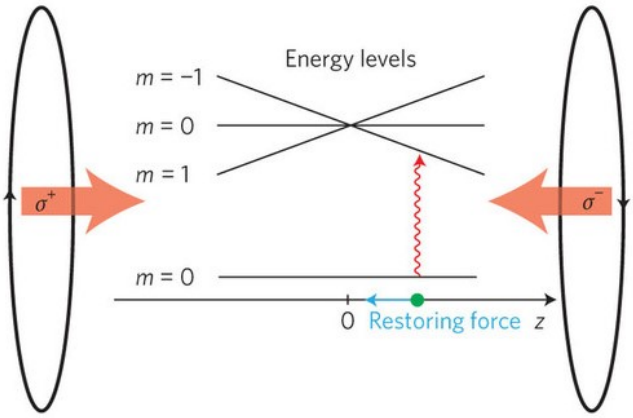


Figure 2.1: Basic schematic of the setup of a MOT along one axis with varying energies for transitions over the position z, as seen in [5]

2.1.2 Optical Bloch equations and forces

The changes of population and the resulting scattering forces on the particles can be derived from the optical Bloch equations (OBE), which evolve the density matrix of the atomic level system in time. This section will be largely based upon [6].
To understand the OBE we will first look at the density matrix itself. For this I will describe a simpler two level system which can be described visually. Note however that in a MOT there are more energy levels.
The density matrix for a two level system is given as:

$$\rho = \begin{pmatrix} \rho_{ee} & \rho_{eg} \\ \rho_{ge} & \rho_{gg} \end{pmatrix} = \begin{pmatrix} c_e c_e^* & c_e c_g^* \\ c_g c_e^* & c_g c_g^* \end{pmatrix} \tag{2.5}$$

where c_e and c_g are the probability amplitudes of the states $|e\rangle$ and $|g\rangle$. The diagonal terms therefore describe the population of the states in an ensemble of atoms or alternatively the probability to find an atom in this state, while the off diagonal states describe the coherence between the states. The following applies: $\rho_{eg} = \rho_{ge}^*$ and $\rho_{ee} + \rho_{gg} = 1$.
Coherence in this case describes the super-positions of the different states.

The density matrix for this system can be described as a vector in a Bloch sphere like figure 2.2. The projection of the vector to the z-axis describes the populations, with the north pole representing the exited state and the south pole the ground state, while the phase in the x-y plane describes coherences. The vector can be described as in [6]:

$$\vec{R} = \begin{pmatrix} c_g c_e^* + c_e c_g^* \\ i(c_g c_e^* - c_e c_g^*) \\ c_e c_e^* - c_g c_g^* \end{pmatrix} = \begin{pmatrix} \rho_{ge} + \rho_{eg} \\ i(\rho_{ge} - \rho_{eg}) \\ \rho_{ee} - \rho_{gg} \end{pmatrix} \tag{2.6}$$

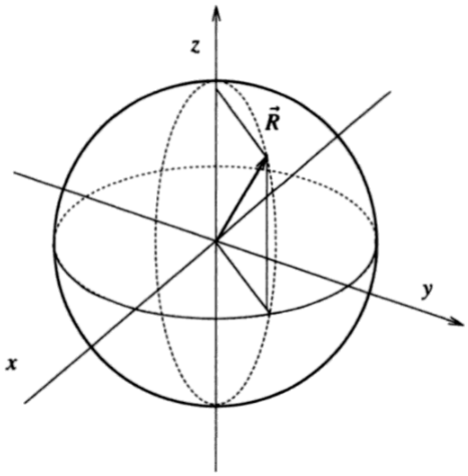


Figure 2.2: Exemplary Bloch sphere [6].

In an electromagnetic field, this vector moves in the Bloch sphere. The direction of this movement and thus the evolution of the populations of the states is influenced by the current position of the vector and thus by the current coherences.

An atom changes states at a rate which is described by the Rabi frequency

$$\Omega \equiv -\frac{dE_0}{\hbar} \tag{2.7}$$

Where d is the electrical dipole moment between the two states and E_0 the field driving the transition. The Rabi frequency will appear in the OBE again, as it describes the change of the components of the density matrix.

We can evolve ρ in time by inserting it into the time evolution of a quantum mechanical operator in the Heisenberg-picture given by:

$$i\hbar\frac{\mathrm{d}\hat{O}}{\mathrm{d}t} = [\hat{H}, \hat{O}] \tag{2.8}$$

To account for spontaneous emission the following terms need to be included:

$$\left(\frac{\mathrm{d}\rho_{eg}}{\mathrm{d}t}\right)_{\text{spon}} = -\frac{\gamma}{2}\rho_{eg} \tag{2.9}$$

$$\left(\frac{\mathrm{d}\rho_{ee}}{\mathrm{d}t}\right)_{\text{spon}} = -\gamma\rho_{eg}$$

and equivalently for ρ_{ge}.
With this we get the optical-Bloch equations as seen in [6]:

$$\begin{aligned}
\frac{d\rho_{gg}}{dt} &= +\gamma\rho_{ee} + \frac{i}{2}(\Omega^*\tilde{\rho_{eg}} - \Omega\tilde{\rho_{ge}}) \\
\frac{d\rho_{ee}}{dt} &= -\gamma\rho_{ee} + \frac{i}{2}(\Omega\tilde{\rho_{ge}} - \Omega^*\tilde{\rho_{eg}}) \\
\frac{d\tilde{\rho_{ge}}}{dt} &= -(\frac{\gamma}{2} + i\delta)\tilde{\rho_{ge}} + \frac{i}{2}\Omega^*(\rho_{ee} - \rho_{gg}) \\
\frac{d\tilde{\rho_{eg}}}{dt} &= -(\frac{\gamma}{2} - i\delta)\tilde{\rho_{eg}} + \frac{i}{2}\Omega(\rho_{gg} - \rho_{ee})
\end{aligned} \tag{2.10}$$

where $\tilde{\rho_{ij}} = \rho_{ij}e^{-i\,\delta t}$ and δ is the detuning.

Assuming a steady state solution of 2.10, that means all time derivatives are 0, we can gain some understanding about the scattering rate from the laser field and with it understand saturation.
In the steady state the Bloch-vector does not move anymore. Where an atom to be in a perfectly homogeneous light and magnetic field, then it would equilibrate into this state.
Solving the OBEs for this state, we can derive solutions for the different components of the matrix to then define saturation, where the off-resonance saturation parameter is given by:

$$s \equiv \frac{s_0}{1 + (2\delta/\gamma)^2} \tag{2.11}$$

the on-resonance saturation parameter is given by:

$$s_0 \equiv \frac{2|\Omega|^2}{\gamma^2} = \frac{I}{I_s} \tag{2.12}$$

and lastly the saturation Intensity is given by:

$$I_s \equiv \frac{\pi hc}{3\lambda^3\tau} \tag{2.13}$$

Saturation occurs because of the atoms lifetime in the exited state. When an atom is already exited, it cannot be exited into the same state again before it has decayed and therefore no absorption can occur in this time. At first glance one could think that, the more photons one sends in the higher the scattering rate is. This is however limited and instead the populations of ground and exited state converge to be equal. As can be seen by 2.13, the saturation fully depends on the combination of the wavelength of the beam used and the decay time, both of which are given by the transition we are trying to drive. Saturation is therefor unique to every transition.
Since excitation into and decay rate out of the exited state γ are equal in the steady state solution, we can get a scattering rate of

$$\gamma_p = \gamma\rho_{ee} \tag{2.14}$$

where the population of the exited state is given by

$$\rho_{ee} = \frac{s_0/2}{1 + s_0 + (2\delta/\gamma)^2} \tag{2.15}$$

Next we take a look at the force in general. The expectation value of the quantum mechanical force is defined as

$$F = \langle \hat{F} \rangle = \frac{\mathrm{d}}{\mathrm{d}t} \langle \hat{P} \rangle \tag{2.16}$$

using 2.8 we arrive at the semi-classical force

$$\vec{F} = -\nabla \langle \hat{H} \rangle \tag{2.17}$$

The force an atom experiences from absorption into spontaneous emission is then given by

$$F = \hbar k \gamma \rho_{ee} = \hbar k \gamma_p \tag{2.18}$$

The first factor of $\hbar k$ describes the momentum transferred by a single photon, the second term $\gamma \rho_{ee} = \gamma_p$ describes the rate at which photons scatter of the atom. 2.18 will be used by both the heuristic model and the rate equation model. The OBE model will expand the optical Bloch equations to include more energy levels and directly use the Hamiltonian and 2.17.

2.2 Different models for calculation of the force on the atoms

For our simulations we will calculate the forces on the atoms using PyLCP. PyLCP gives us three models in order to simulate these forces. The heuristic equation, the rate equations and the full optical Bloch equations (OBEs). This section describes these models and the differences between them.

2.2.1 Heuristic model

The heuristic model is a fast but fairly adhoc way of calculating the force. Continuing with the steady state solution we can calculate the forces the atoms experience broadly.
Combining 2.18 and 2.12 and accounting for a further frequency shift from the Doppler effect $\omega_d = -\vec{k} \cdot \vec{v}$ we arrive at

$$F = \hbar \vec{k} \gamma \frac{s_0/2}{1 + s_0 + 4(\delta - \vec{k} \cdot \vec{v})^2/\gamma^2} \tag{2.19}$$

as the force per laser beam, see [6] p. 74-75.
We also need to account for the frequency shift due to the Zeeman effect and the polarisation of the beams. The former is done by adding in the term $q \cdot \mu_b \cdot B$, where $q = -1, 0, 1$ represent σ^-, π and σ^+, and summing over these three options. This can be done for a F = 0 to F = 1 transition where the dipole transition strength is equal for all three transitions and the Lande factor is 1.
Finally we need to take a closer look at the polarisation. To make the calculation easier we want to rotate the polarisation onto the magnetic field. Technically, the magnetic moment of the Zeeman split states rotates around the local magnetic field axis with the Lamor frequency. Moving into a frame of reference following the B field at every point means we lose information about the rotation between the two points. However, because the Lamor frequency is so high compared to the time it takes to move to the next point in space we can neglect this effect and use the absolute value of the magnetic field, while handling the polarisation as part of the intensity rotated onto an axis defined by the local magnetic field.
Combining the previous concept and adding up the force of all beams leads to the acceleration of a particle as calculated in PyLCP [3] with particle mass M, laser index l and constant acceleration a (e.g. gravity):

$$\ddot{\vec{r}} = \frac{\hbar \vec{k}_l \gamma}{2M} \sum_{l,q} \frac{s_{0,l} (\epsilon'_{l,q})^2}{1 + \sum_j s_{0,j} + 4(\delta - \vec{k} \cdot \vec{v} - q\mu_b |B|)^2/\gamma^2} + \boldsymbol{a} \tag{2.20}$$

$(\epsilon'_{l,q})^2$ is the polarisation of the laserbeam l rotated onto the magnetic field vector.
Also note that the total saturation of all beams was approximated as a sum over all saturation parameters. This is an approximation, as depending on the lasers the reality lays somewhere inbetween this aproximation and an aproximation where we look at the saturation of each beam separately. The latter might be more applicable to our usecase, as we will see later on in this thesis.

2.2.2 Optical bloch equations

The OBE (optical bloch equations) model does not assume the steady state anymore. Instead the goal is to evolve the density matrix in order to be able to calculate a force at every point from the populations and coherences that the density matrix describes. Equations in this section are taken from

[3] unless specified otherwise, though one could compare the equation to [7] and [8], since this is where PyLCP bases its model upon.

Starting from the full Hamiltonian

$$\hat{H} = \hat{H}_{\text{atom}} + \hat{H}_{\text{field}} - \hat{\boldsymbol{d}} \cdot \hat{\boldsymbol{E}} - \hat{\boldsymbol{\mu}} \cdot \hat{\boldsymbol{B}} \tag{2.21}$$

Where $\hat{\boldsymbol{d}}$ and $\hat{\boldsymbol{\mu}}$ are the dipole operators, the field hamiltonian is

$$\hat{H}_{\text{field}} = \int \left(\frac{\epsilon_0 \hat{E}}{2} + \frac{\hat{B}}{2\mu_0} \right) \mathrm{d}V \tag{2.22}$$

and the atomic Hamiltonian is

$$\hat{H}_{atom} = \frac{P^2}{2M} + \hat{H}_{int} \tag{2.23}$$

$\hat{H}_{int}$ is the Hamiltonian describing the atom's internal Hamiltonian, while the first term describes its kinetic energy.
Using 2.8 we can get a time evolution of ρ for the given Hamiltonian at any given point. The derivation of the evolution will give similar results to 2.10, though some changes will be applied to create a evolution that is efficiently solve able in code and to derive the solution including decay term.

In order to include a decay part into the equation a radiation reaction approximation is applied. This approximation adds the electric field created by photon emission to the external electric field. The gradient of this field is 0 at the position of the atom and will thus not affect the force in any way. The approximation can be described as (see [8]):

$$\hat{E}(\vec{x}) = \hat{E}_0(\vec{x}) + E_{RR} \tag{2.24}$$

$$E_{RR} = \frac{1}{6\pi\epsilon_0 c^3} \frac{\mathrm{d}^3 \hat{\boldsymbol{d}}}{\mathrm{d}t^3}$$

The time evolution now includes a real component describing the decay. PyLCP writes this equation in matrix form $\dot{\boldsymbol{\rho}} = (1/\hbar)\boldsymbol{\Gamma} \cdot \boldsymbol{\rho}$ where ρ gets turned into a flattened vector in order to later be used with a numeric solver.
The imaginary components of $\hat{\boldsymbol{d}} \cdot \hat{\boldsymbol{E}}$ describes the coherence between the states here. This part has to be included in the evolution as well. So similarly PyLCP turns the rest of the evolution into matrix form too. In total this yield for the evolution of ρ:

$$\dot{\boldsymbol{\rho}} = \frac{1}{\hbar}[\boldsymbol{\Gamma} - i(\boldsymbol{H_{int}} - \sum_{q,n,m} (-1)^q [\boldsymbol{D_q^{nm}} E^*_{0,n\to m,-q} - \boldsymbol{D_q^{*nm}} E_{0,n\to m,-q} - \boldsymbol{M}_{n,q} B_{-q}])] \cdot \boldsymbol{\rho} \tag{2.25}$$

The shape and factors of the matrices then describe the OBEs. The shape of these matrices combined with 2.7 gives us back 2.10 with an additional magnetic field part. The only missing part is the detuning, which is either contained in $\boldsymbol{H_{int}}$ or the electrical field depending on computational speed. With the evolution of the density matrix done, we can now move to the force. From 2.17 we get the

semi classical acceleration of the particle:

$$\ddot{\vec{r}} = \frac{1}{M}[\nabla(\hat{\boldsymbol{d}} \cdot \hat{\boldsymbol{E}}_{\mathbf{0}} + \hat{\boldsymbol{\mu}} \cdot \hat{\boldsymbol{B}})] + \boldsymbol{a} \tag{2.26}$$

in this, $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{d}}$ are given as expectation values using the density matrix via the calculation for any operator $\hat{O}$

$$\langle \hat{O} \rangle = Tr(\hat{O} \cdot \rho) \tag{2.27}$$

2.2.3 Rate equation model

With the rate equation model we do away with coherences in the optical Bloch equations and purely focus on the population of the states. Instead of evolving the density matrix we now evolve the population vector of a given state and manifold $|i, n\rangle$, which is given from the Einstein rate equation [9] like

$$\dot{N}_i^n \pm \sum_{m,j,l} R_{i,j,l}^{n\to m}(N_i^n - N_j^m) + \sum_{m>n} \gamma_{ij}^{m\to n} N_j^m - \sum_{m<n} \gamma^{m\to n} N_i^n \tag{2.28}$$

where the first term describes the optical pumping, the second term describes the decay into the state from higher states and the last term describes the decay out of the state.
For the force we will want to use 2.18. For this the excitation rate will need to be calculated. As can be seen in [10], similarly to the heuristic equation it is given with laser index l

$$R_{i,j,l}^{n\to m} = \frac{\gamma}{2} \frac{f_{i,j,l} s_0}{1 + 4(\delta_l - \vec{k}_l \cdot \vec{v} - \Delta\omega)/\gamma^2} \tag{2.29}$$

with the fractional strength of the driven transition $f_{I,j,l}$ and the energy difference between the states between which the transition is driven $\Delta\omega$. The fractional strength is given in [10] with polarisation ϵ_l

$$f_{I,j,l} = \frac{|\langle i| \boldsymbol{d} \cdot \epsilon_l |j\rangle|^2}{\sum_k |\langle k| \boldsymbol{d} |u\rangle|^2} \tag{2.30}$$

With this the force can be calculated using

$$\ddot{\vec{r}} = \frac{\hbar \vec{k} \gamma}{2M} \sum_l R_{i,j,l}^{n\to m}(N_j^m - N_i^n) + \boldsymbol{a} \tag{2.31}$$

where $\sum_l R_{i,j,l}^{n\to m}(N_j^m - N_i^n)$ describes the upper level population according to Einstein's rate equations [9].

2.2.4 Momentum diffusion

Previous calculations give us an expectation value of the force at any given point in our evolution of the particles trajectory. Yet it does not contain the fluctuation of said force as given by the random spontaneous emission events. For this we turn to the so called momentum diffusion constant.

It is given from [7] as

$$2D_p = \left(\frac{d}{dt}\right)\left(\langle \vec{P}\cdot\vec{P}\rangle - \langle\vec{P}\rangle\cdot\langle\vec{P}\rangle\right) = 2Re\int_{-\infty}^{0} dt[\langle\vec{f}(t)\cdot\vec{f}(0)\rangle - \langle\vec{f}(t)\rangle\cdot\langle\vec{f}(0)\rangle] \tag{2.32}$$

Following through with this calculation for a two level system one eventually gets

$$2D_p \approx \hbar^2\gamma\rho_{ee}(k^2+\alpha^2+\beta^2) \tag{2.33}$$

where β describes absorption, α describes stimulated emission and the wave vector k part describes the spontaneous emission. PyLCP now neglects the latter two terms and generalizes the results for more states. This gives

$$2D_{ii} = k^2_{n\to m}\sum_{n<m}\gamma_i^{n\to m}\rho_{ii}^{mm} \tag{2.34}$$

2.2.5 Differences between the models

Knowing the models a little better I will now describe what physical concept each model would capture compared to the previous model.

For this we will want to consider a situation where the atom moves around in a perfectly homogeneous magnetic and light field. As mentioned before the particle will eventually move into the steady state solution, but at the start its population will fluctuate a little as the Bloch vector finds its equilibrium. Figure 2.3 shows this process.
The particle would find a new equilibrium like this depending on the local fields in a real MOT.
The heuristic model neglects the first part of this graph and assumes the particle immediately moves into the constant population observed towards the end.
These fluctuations, described by the current Rabbi-frequency, happen on a timescale of $1/\gamma$.
If an atom moves the distance of the wavelength of light in that time, then the phase of the coherence relative to the light field matters. If γ is low and therefore the fluctuations are long, and if the atom is moving slowly at a speed at which the coherences matter then the OBEs must be used. However, if only the fluctuations play a role, then the rate equation, which capture the Rabbi-frequency are sufficient. If however, γ is high compared to the time of flight $1/t_{flight}$ for the atom to travel one λ, then the heuristic model becomes a good approximation because the relative phase does not play a role and the fluctuations take too little time to matter.

As we have seen the heuristic model depends on the assumption that there is no change in population and ignores coherences. This means that this model will give more of an averaged force over longer time spans. By adding in the current population of the states we can include fluctuation of the populations and therefore the Rabbi-frequency. Finally with the OBE we can add in coherences to capture the full internal dynamics of the atoms.

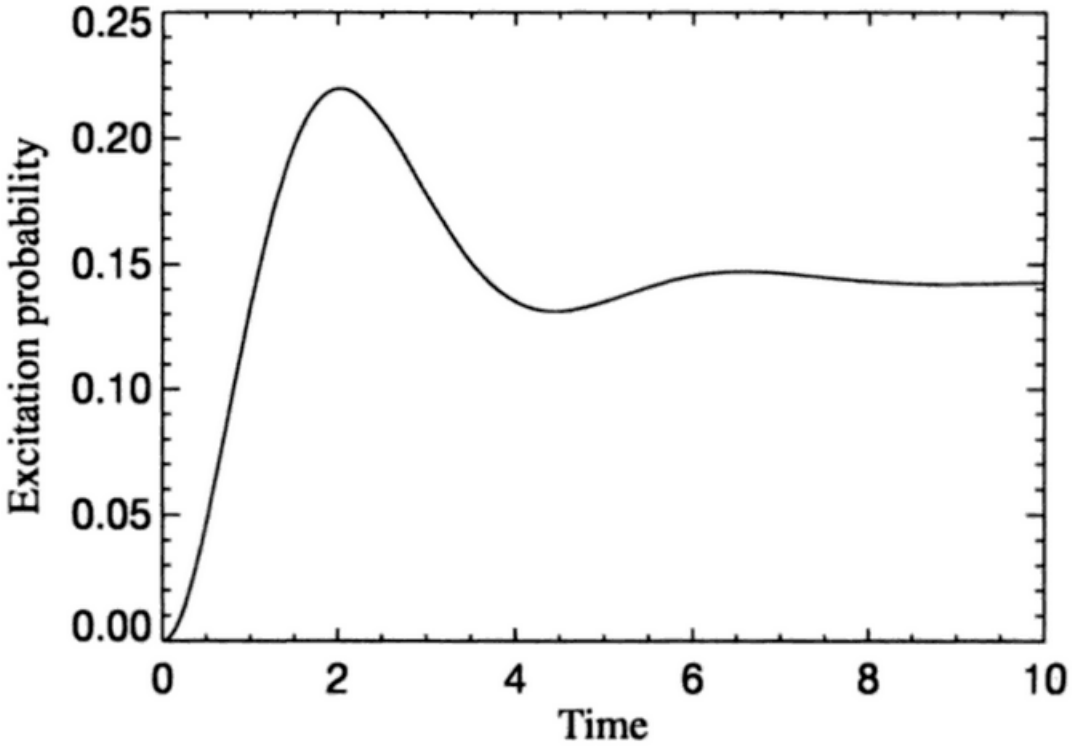


Figure 2.3: Probability $|c_e(t)|^2$ for the atom to be in the exited state for $\Omega = \gamma$ and $\delta = \gamma$ with time in units of $\frac{1}{\gamma}$, calculated numerically from the OBEs by [6].

CHAPTER 3

Description of the experimental setup

This chapter describes the experiment and shows its layout, so that the implementation of the setup later in this thesis can be understood.

The experiment consists of two chambers. A vacuum chamber with a two dimensional magnetic optical trap, which creates an atomic beam, and a science chamber with a 3D MOT which traps and cools the incoming beam in order to conduct experiments on. The atomic beam created in the 2D MOT chamber ensures that we have a high enough atom density going into the 3D MOT. This is necessary for Ytterbium since dispensing particles directly into the 3D MOT would not trap enough particles, as the vapor pressure from this element is too low.

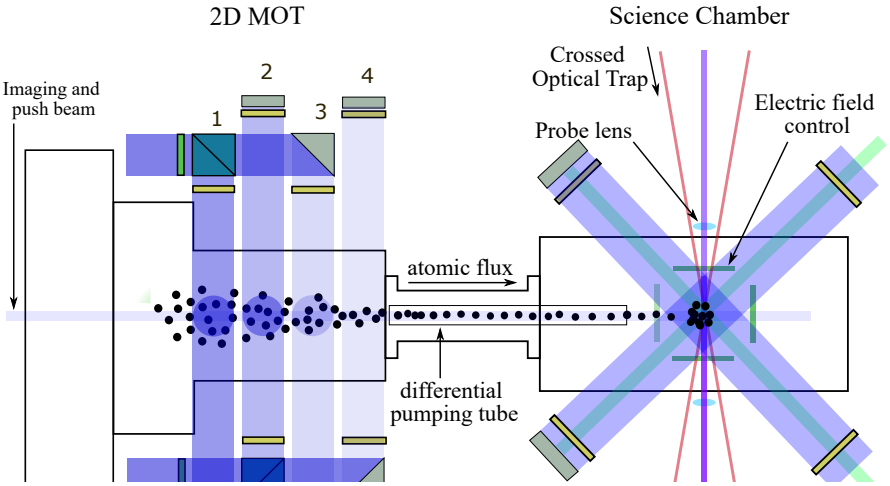


Figure 3.1: Schematic of the setup as used in the YQO group. Picture created by Thilina Senaviratne, who is a part of the group.

Figure 3.1 shows the setup of the experiment. Note that the z-axis is defined towards the science chamber, while the y-axis is defined upwards from the ground. The distances shown are the ones used in my code.
Atoms are heated in dispensers at a temperature of $T = 500\,°C$. These dispensers are located at the diagonals of the chamber about 4 mm back from the center of the first beams. They shoot out the atoms at an angle of 45° towards the center of the chamber.
An atom coming out of the dispenser first enters the vacuum chamber where four two dimensional MOTs collimate the atoms into a beam. The atoms however come out in a cone, which can be described by an angle in the x-z and the x-y planes. Figure 3.2 shows the cone in the x-y plane.
Having measured the angle from this picture I come to an opening angle of $\beta = 23.5°$.

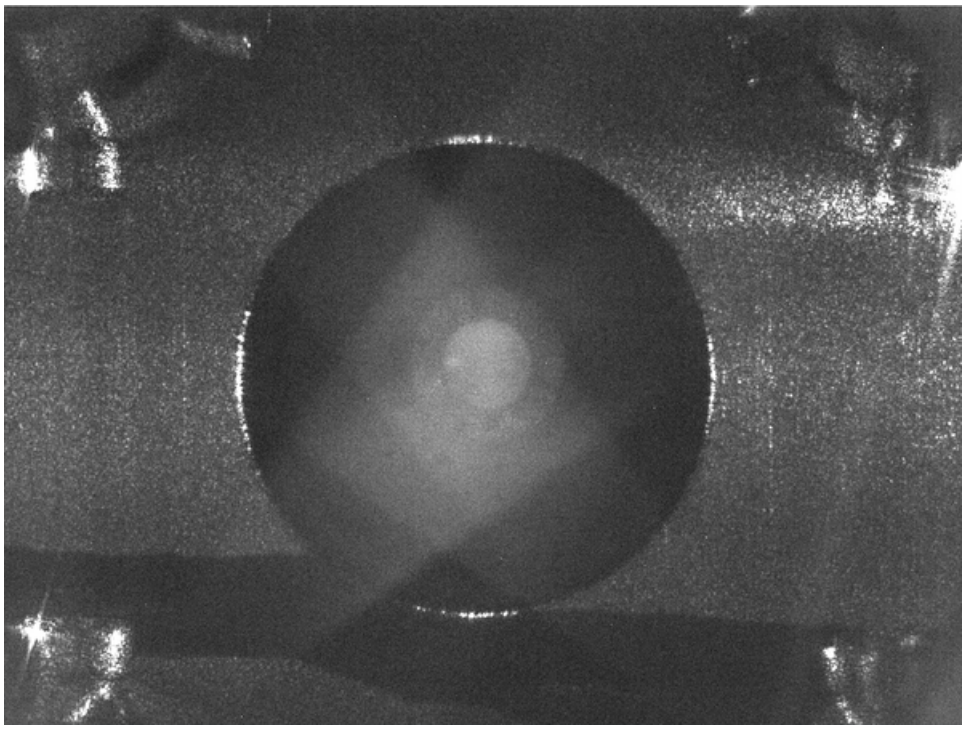


Figure 3.2: Picture of the 2D MOT with the tube in the back as atoms come out of the dispenser. The cones are lit up slightly in the picture.

The x-z angle is harder to determine as no pictures along this axis can be taken. Therefore I refer to an older thesis from the group [11] for an angle of $\alpha = 23°$.

From the dispenser the particle enters into the laser fields. These lasers are retro-reflected to use their power once again in the opposite direction.
The total power of the laserbeams is distributed as listed in table 4.1, measured for a total power after the 2D MOT fiber outcoupler of 100.1 mW. Such a distribution is chosen because the first of the four MOTs overlaps with the dispenser. This means that the atoms are faster here and thus need more power to be trapped.

Table 3.1: Laser power distribution of the 2D MOT

MOT number	Horizontal beams [mW]	Vertical beams [mW]	Total [%]
1	33.5	24.8	59%
2	6.75	3.13	11%
3	6.26	10.86	17%
4	0.4	0.29	1%

A quadrupole magnetic field with a gradient calculated to $\alpha_{2D} = 54 \frac{G}{\text{cm}}$ and a very small z component is present here. A pushbeam has been added to push particles through the tube in order to increase the capture number in the 3D MOT. It has a detuning of $\delta = 17.5\,\text{MHz}$ a power of $P_{push} = 41\,\mu\text{W}$ and a waist of $w_{push} = 0.9\,\text{mm}$. Its detuning is positive to ensure that it mainly pushes the particles already moving towards the science chamber.

Moving along towards the science chamber the atoms enter a differential pumping tube, which ensures the pressure in the chambers stay low. The length for this tube can be seen in figure 3.1. It is situated 9.6 cm away from the center of the first beam and its diameter is 3 mm.

From this tube the particle then enters into the science chamber where it is trapped using a full 3D MOT.
The lasers beams in the x-z plane are angled at 45°. The laser beams divide their total power of $P_{3D} = 40\,\text{mW}$ as 2:1:2, where the two horizontal pairs of beams get double the power the vertical gets.

The atoms go through three stages in the 3D MOT. In the first stage, the particles are trapped with blue light. The second stage consists of a handover process to the green light. The YQO group has optimized this transition through the use of frequency broadening. In the last stage, the atoms are then cooled in a pure green MOT.
Both the handover and the green cooling process take comparatively long to the trapping in the 2D and 3D blue MOTs. This part will therefore not be simulated. Instead, only the trapping in the blue 3D MOT will be looked at.

CHAPTER 4

PyLCP: A python package for computing laser cooling physics, description and analysis

PyLCP implements the three models described above [3]. This allows one to choose which model to use based on the accuracy needed for the current problem at hand.
In this chapter, I will explain the implementation of PyLCP a bit more in detail and analyse some changes I made that are useful for our simulations. In addition I will compare the models in order to see what level of accuracy is needed for our simulations.

4.1 PyLCP implementation

PyLCP works by letting the user specify the laser fields, magnetic field and Hamiltonian for the problem at hand. These are inserted into the selected model. The module then provides functions to run the the simulations.

These function take the given fields and Hamiltonian and use them to calculate the forces the particles experience at their position using the the equation given in 2.2. For the rate equation the time derivative of the populations of the states also get calculated. For the optical Bloch equation the $\dot{\rho}$ is calculated.
These, together with the current state of the particle, are then inserted into a Runge-Kutta solver [12, 13]. Such a solver finds numerical solution for differential equations.

4.2 Simplifications made by PyLCP

Since the implementation can only have a finite complexity, some simplifications are made by PyLCP.

The first of these is found in the laser beams. The only positional argument the beams take are the waist for Gaussian beams. However, the beam does not get assigned an origin, nor does the beam width vary with distance. This means that the Gaussian beams are infinitely long Gaussian distributions. This suffices for our setup, but one could imagine a setup where the distances are very large, making this relevant.

The second simplification is made in order to shorter computational times. The momentum diffusion is simulated using a Monte Carlo method. Such a method relies on repeated generation of random values in order to approximate complex analytical problems numerically. Specifically, PyLCP simulates the diffusion by applying random 'momentum kicks' to the particle at random moments determined by the decay rate. This is called 'random recoil' by PyLCP in the code and can be turned on and off using a Boolean.
During every time step Δt, PyLCP calculates the probability of a decay as [3]

$$P = \Delta t \gamma_i^{m \to n} \rho_{ii}^{mm} . \tag{4.1}$$

A random number p is generated, and if p < P, two random $\hbar k$ kicks are applied to the particle.
The timestep Δt is shortened after this process in order to make sure no multiple decays happen in future steps of the solver. This is done based of the amount of scatters that PyLCP calculates to have happened. This means that the runtime will increase significantly if one turns this method on.

A final limitation of PyLCP is that we can only specify a single γ. This means that for the heuristic model we cannot really simulate a multiple transition setup like in Rubidium. For the other models there is a workaround as described in [14], but it is rather complicated.

4.3 Units

By default, PyLCP sets γ and k to 1 and $\hbar = 1$. In principle however one can specify any unit system as long as $\hbar = 1$ is conserved.
I chose to keep γ and k as 1, which means that we rescale our timescale and space coordinates to $x_0 = \frac{1}{k}$ and $t_0 = \frac{1}{\gamma}$. Because I give k in cm this means that $[x_0] = \mathrm{cm}^{-1}$ and $[t_0] = \mathrm{s}^{-1}$.
All velocities, times, positions and gravity can be rescaled accordingly. We however want to pay special attention to the magnetic field and the mass, as we want to make these unitless as well.
We can do this using the following rescalings:

$$M^{'} = M \cdot M_u \cdot \left(x_0 \cdot 0.01 \frac{\mathrm{m}}{\mathrm{cm}} \right)^2 \frac{1}{\hbar \cdot t_0} \tag{4.2}$$

where M_u is the atomic mass constant and M the mass of the particle in atomic units. For the magnetic field gradient we use

$$\alpha^{'} = \mu_B \cdot 10^{-4} \frac{T}{G} \cdot \alpha \cdot \left(2\pi t_0 \right) \tag{4.3}$$

where α is the magnetic field gradient in $\frac{G}{\mathrm{cm}}$, μ_B is the Bohr magneton and $2\pi t_0$ is the decay time associated with γ.

4.4 Timestep comparisons

The Runge-Kutta solver has an option to determine the upper boundary of the timesteps Δt between calculations by the solver. Ultimately, the solver chooses the step size based on an estimated error that would accumulate per timestep. However the user can specify the maximum Δt. A default limit is

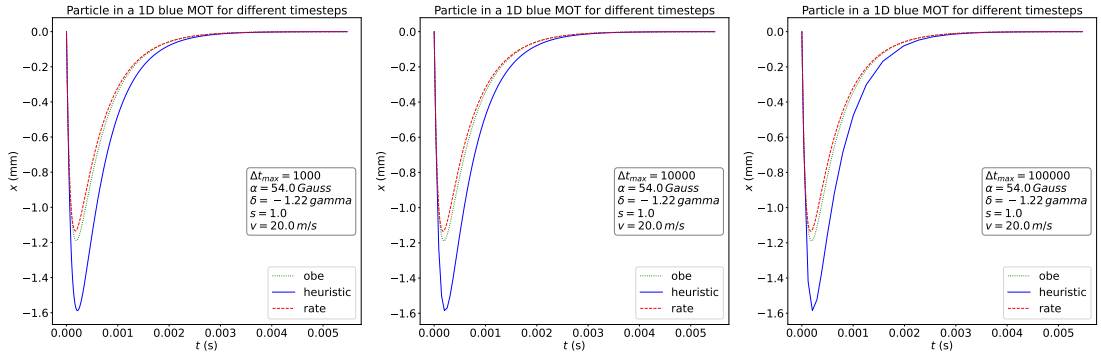


Figure 4.1: Particle simulation for different allowed maximum timesteps.

implemented, but can be changed using an appropriate variable.

The heuristic model can use larger timesteps because it purely calculates the average of the force the atoms experience. This changes with the local fields the particle experiences, who change more slowly in position then the coherences and population of the other models. The optical Bloch equations and rate equations meanwhile will choose smaller timesteps to accuratly simulate the faster changing internal state of the atoms.
However, if we allow the timesteps to be to large, as we can see in figure 4.1, the solver, seeing very little change in the light and magnetic fields, will take steps that are so large the trajectory stops being smooth for the heuristic equation. In order to prevent this I set the timestep to $\Delta t = 1000\,\frac{1}{\gamma}$ for the blue MOT.
For other MOTs this value needs to be changed based of γ, but for the purposed of our simulation it works fine.

4.5 Saturation intensity comparisons

Equation 2.20 from chapter 2.2 shows that per default, PyLCP handles the saturation in the heuristic model by summing over the saturation of all laser beams. This would model an atom that gets saturated by all beams simultaneously.
However, since we will be using large detunings and magnetic fields, this approximation might not be applicable to us. Instead the atom will interact only with one laserbeam along an axis because of the concepts described in chapter 2.1.1. Therefore, I tested whether or not we could get more accurate results for the heuristic equation if we instead just use the saturation of the laserbeams individually, therefore replacing the sum with just the saturation parameter of a single laser.
In figure 4.2 we see that, for parameters like the ones used in the experiment, we find that applying the saturation of the beams individually yields results practically equal to the more accurate models. Our suspicions therefore turn out to be correct.

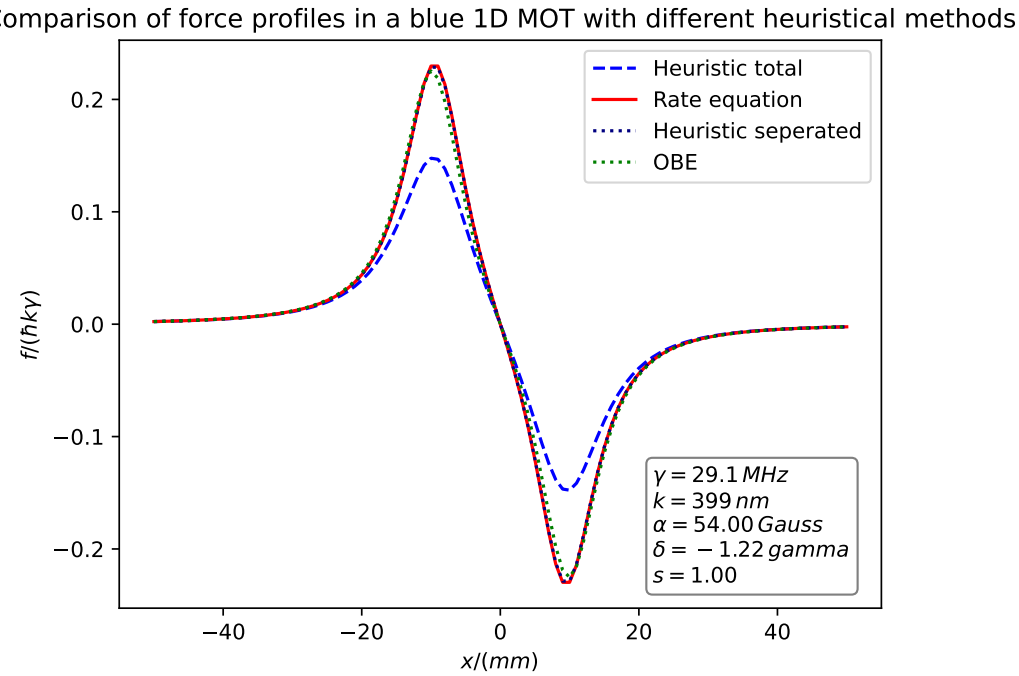


Figure 4.2: Comparison of heuristic methods for a blue 1D MOT. Heuristic total describes the default equation PyLCP uses, while Heuristic separated describes the saturation of each beam being applied individually.

I also tried to find a case where the opposite is true, therefore where the atom interacts with both beams as equally as possible. For this I consider to the most extreme case, a optical molasses (therefore no B-Field) with extremely small detuning.
Because there is no magnetic field, that could change with position, in the optical molasses, we have no positional component to the force from there. Additionally, because the particle moves in a one dimensional space, in which PyLCP makes no changes to the light field, we also do not gain any positional arguments from the laser fields. It therefore makes more sense to look at the force as an argument of the particles velocity. Figure 4.3 shows how both approximations now deviate from the more accurate OBEs. The beams still have to be seen somewhat separated as the dopplershift causes the atom to be more on resonance with one of the two beams, but not entirely as the magnetical fields are removed and the detuning is low.

From this I gather that separating the intensities is the correct way forward.

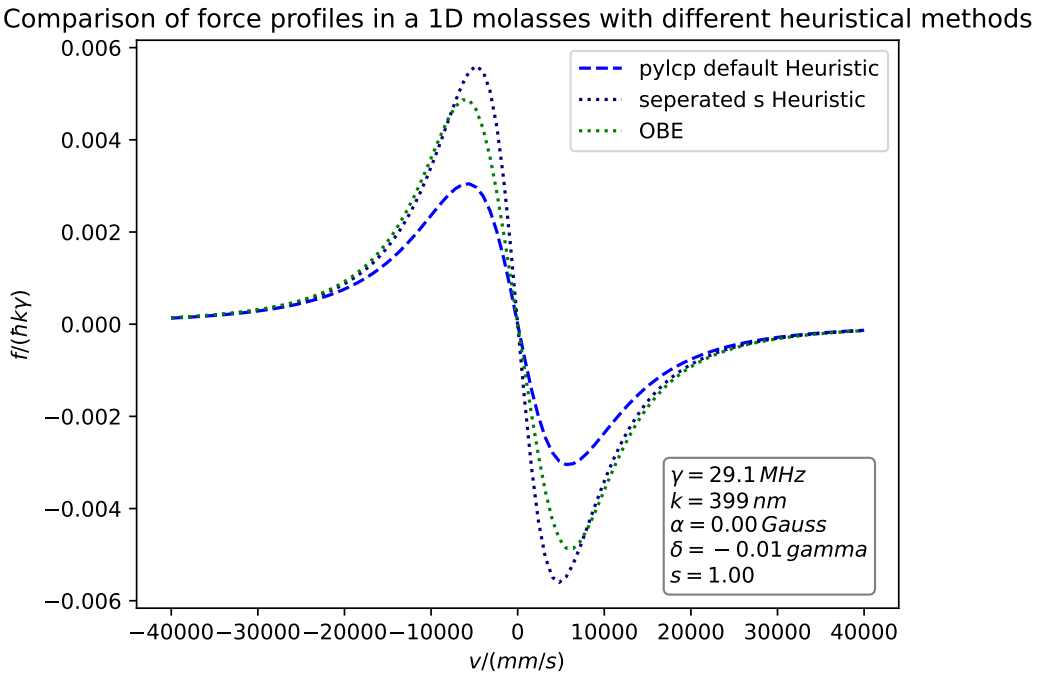


Figure 4.3: Comparison of heuristic methods for a blue 1D MOT. Again default is the default used by PyLCP, while separated mean applying the saturation individually per beam.

4.6 Model comparison

In order to know what model we want to use later on for the simulations, I compared the different models by sending out particles in one dimensional MOTs with variables as used in the experiment. I did this for the later described blue and green MOT with and without random recoil. The blue MOT has a broad linewidth of $\gamma = 29.1\,\text{MHz}$, while the green MOT has a narrow linewidth of $\gamma = 182\,\text{kHz}$.

Figure 4.4 shows a particle with a speed realistic to the experiment as it moves in a one dimensional blue MOT for all three different models. The particle starts in the MOT center and moves away from it, to then be trapped by the MOT as it is again pushed to the center of the MOT. The figure shows this for the case with momentum diffusion and without.
Comparing the models we can see that the OBE differs slightly from the other two. A combination of a typical speed and the blue γ therefore create a small difference for this MOT.
The blue light is mainly used to trap the atoms when they are still fairly fast and traveling in larger spaces. The heuristic model is then totally sufficient here, since the OBE only differs less then 0.1 mm from the heuristic model, while the 2D MOT chamber is several centimeter in width.
Adding on the momentum diffusion in figure 4.4 adds flucutations to the particles position. These are most notable when the particle is slowed towards the center. Here these random recoils create fluctuations in the range of $\Delta x = 0.2\,\text{mm}$. This again is insignificant for the blue MOTs.

To compare the models for the green MOT it is more useful to look at the velocity, as the green MOT

in the experiment is used to cool the atoms to very low temperatures, which correlates with the velocity of the particles.
Figure 4.5 shows the velocity in time. Again this simulation was done for a particle velocity realistic to the experiment. We conclude that there is even less of a difference between the OBE and the other two. This shows that even here γ is high enough compared to $1/t_{flight}$ for the heuristic model to be accurate.
However when we now turn on random recoil, as seen in figure 4.5, we can see that diffusion start to matter here significantly.
This is because one random kick by a photon causes a much larger dopplershift relative to the linewidth, which causes the particle to be strongly red detuned compared to before the kick.

From this I conclude that for the blue MOTs the heuristic model without momentum diffu-

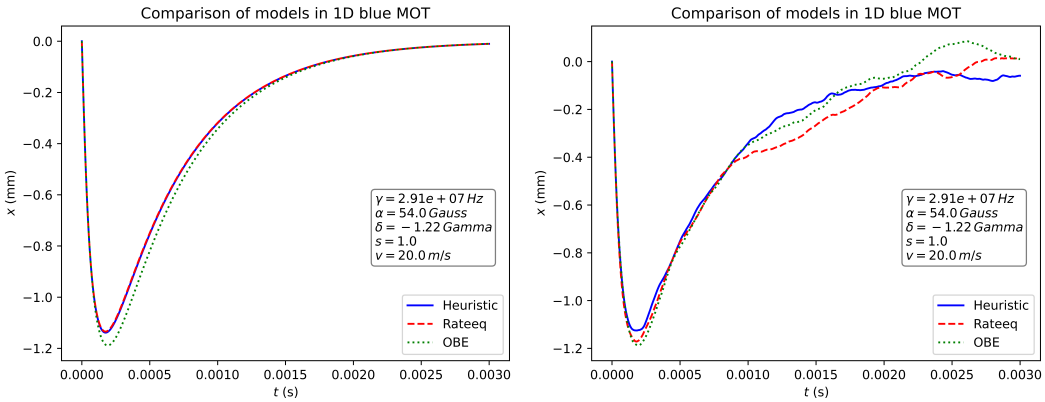


Figure 4.4: Comparison of the 3 models for parameters like the blue MOTs used in the experiment without random recoil (left) and with random recoil (right).

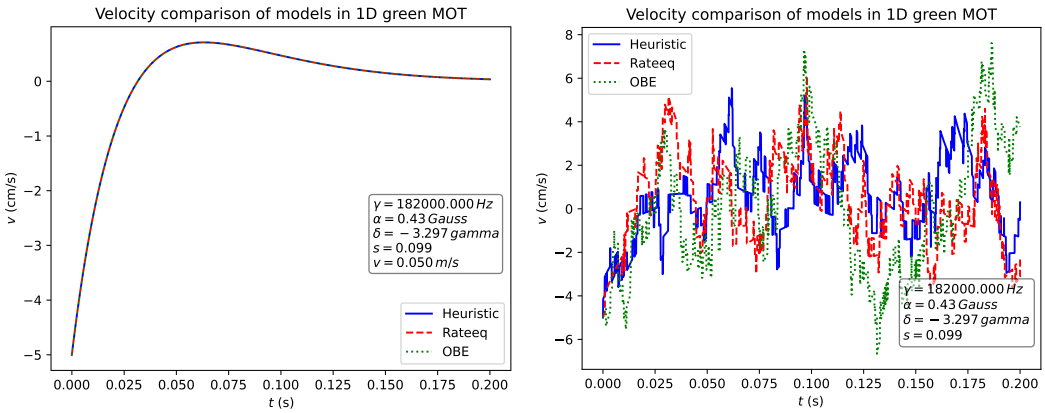


Figure 4.5: Comparison of velocities for parameters like the green MOT used in the experiment without random recoil (left) and with random recoil (right).

sion will suffice. Even for the green MOT the heuristic model will suffice, however we need to consider

momentum diffusion here as the fluctuation in velocity means significant fluctuations in temperature for a ensemble of atoms.

4.7 Runtime improvement

Computational times for the simulations can get long, with a simulation of one atom taking several seconds. Since we want to simulate many atoms many times I spends time trying to speed up the code. This is described in more detail in appendix A. I ended up using multiprocessing, which allows me to use multiple cores of the computer to simulate multiple particles at the same time.

4.8 conclusion

With my findings of this chapter I can now setup the simulations of the experimental setup. I found that the heuristic model is significantly more accurate when the saturation of every beam is applied separately. I furthermore found that for our simulations the heuristic model suffices. I lastly implemented changes to decrease computational times. This is done by choosing larger timesteps for the solver and by implementing multiprocessing.

CHAPTER 5

Implementation of the experimental setup

In order to implement the setup I have divided the simulation into three parts. A dispenser sampling part, where I sample velocity vectors for the particles that get simulated. A 2D MOT simulation, where the trapping in the 2D MOT and the movement along the z axis through the tube are simulated. And a 3D MOT simulation, where the 3D MOT trapping is simulated. Dividing the 2D and 3D simulation has the benefit that one can simulate the parts separately, which could pay a lot in computational time in case one of the two simulations isn't needed, makes debugging of single components easier and allows for more both components to be simulated separately.

In order to be able to make predictions about good parameters for the different components of the experiment I want to be able to scan the parameters. I do this by selecting different sets of parameters with which i simulate 1000 particles each time. I then count the number of particles trapped.
For this to run smoothly both the defining the variables of the experiment and the simulations are handled in a class. This class then has functions with which I can initialize the setup for both 2D and 3D separately and functions that allow me to set certain variables and reset certain objects like the lasers or magnetic fields accordingly.

5.1 Dispenser sampling

In order to get usable data from my simulations, we need to add some sort of realistic random sampling of atoms. This will ensure that we can get realistic data without having to simulate too many particles, because the particles we simulate will be representative of the particles that would go through the experiment.

For this I generate velocity vectors representing the starting velocities of the particles.
This is done by first generating an absolute velocity and then assigning generated angles to create a carthesian velocity vector.

The absolute value of the velocity of a particle is taken from a Maxwell-Boltzmann distribution with

probability density:

$$f(v) = \sqrt{\frac{2}{\pi}} v^2 \frac{m}{k_B \cdot T} e^{-v^2 \frac{m}{k_B \cdot T}/2} \tag{5.1}$$

The dispenser is supposed to operate at about 500°C. Using this we generate velocities as can be seen in figure 5.1. I have build in a cutoff point for the velocity, above which no velocity is actually simulated, because simulating particles with speeds to high to be trapped is just a waste of time. By trying out the simulation for a few different such velocities, I came to the conclusion that 50 m/s works well. When going to higher powers of the MOT this velocity could be increased as well.

Next I generate two angles. One for the x-z plane and one for the x-y plane.
The distribution for these angles is based on PhD thesis [15]. From here I deduced that the probability to find a particle with opening angle α must follow a cosinus. Therefor the following distribution was chosen:

$$f(\alpha) = \frac{1}{2\pi}(1 + \cos\alpha) \tag{5.2}$$

Which gives us a a normalized and raised distribution based on a cosinus. The constraint to the angles are given in chapter 3. In order to account for these I reroll any angle until it fits within the constraints. One angle constraint that still has not been attended to, is the angle towards the back of the chamber, away from the science chamber. In the experiment the particles are mostly launched towards the science chamber. Similarly I chose a, quite arbitrary, angle of 5° for the maximum angle here because most particles who would fly out towards the back don't get trapped anyways and this way we save on computational time by not simulating those.

These angles and the absolute velocity now make spherical coordinates which are then converted to

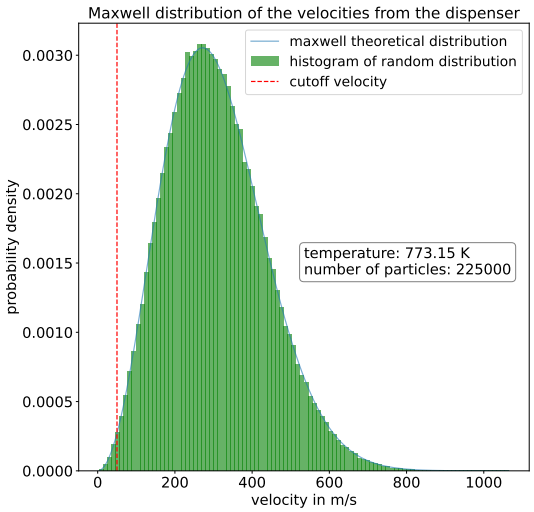


Figure 5.1: Exemplary velocity distribution

carthesian coordinates using:

$$\begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} = \begin{pmatrix} v \cdot \cos\beta \cdot \cos\alpha \\ v \cdot \sin\beta \cdot \cos\alpha \\ v \cdot \sin\alpha \end{pmatrix} \tag{5.3}$$

This then gives us the initial velocity vectors which describe our particles.

5.2 2D MOT

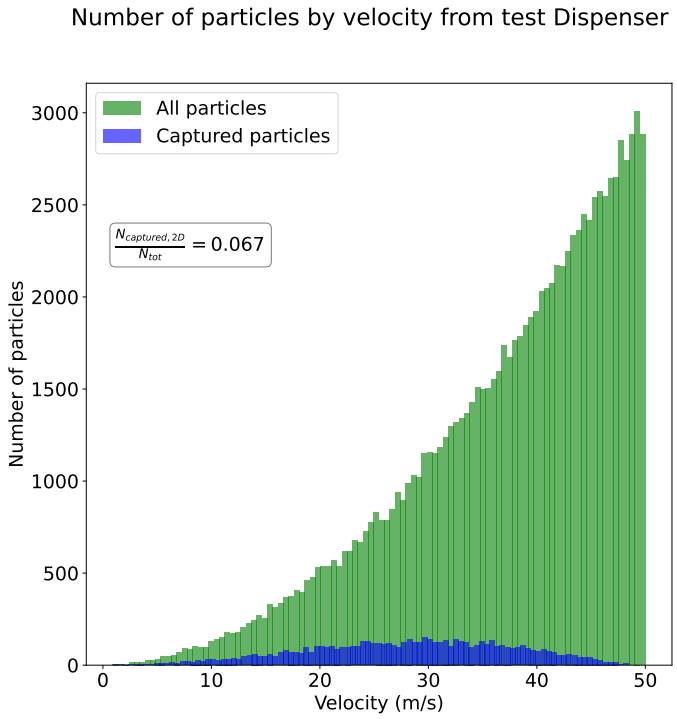
For the MOTs we can use edited versions of functions in PyLCP. PyLCP gives us quadrupole magnetic fields of form $B = \alpha(-\frac{x\hat{e}_x}{2} - \frac{y\hat{e}_y}{2} + z\hat{e}_z)$. In the 2D MOT however the z component is kept very small. For the purposes of my simulations it is set to zero.
Therefor a magnetic field of shape $B = \alpha(-\frac{x\hat{e}_x}{2} - \frac{y\hat{e}_y}{2})$ was created. Similarly an edited Gaussian laser beam function, that allows for a offset, was created, so that one could add in multiple lasers in a spatial arrangement.

The zero of the coordinate system is set at the center of the first of the four 2D MOTs. I simulate particles coming from one of the dispensers at the bottom. No other dispenser needs to be added since the setup is almost fully symmetric, except for gravity, which plays very little role in the initial simulation of the 2D MOT, as the particles move fairly fast compared to the gravitational acceleration they experience.

To save on computational time and to be able to distinguish between captured and lost particles later, I added events for the solver to track. For the 2D MOT, these events terminate the simulation and then set a flag to indicate whether the particle is lost. This happens if the particle either flies out towards the back, towards the sides or does not make it into the tube. The first two events save a lot of time, as they quickly stop any particles that are to fast to be captured or that fly in the wrong direction.

This simulation also includes the tube and the pushbeam, which will be explained in the next sections, and runs until the particles are either lost somewhere or until they reach four centimetres away from the 3D MOT. This distance is chosen to ensure that the pushbeam is simulated throughout the tube, while making sure that any particles exiting at the sides of the tube have not yet entered the beams. These particles at the sides will reach the beams faster because of the 45° angle of the 3D MOT. Because of this a larger distance must be set.

I count particles as trapped by the 2D MOT if they make it to the endpoint of the simulation without getting lost. With this I can support my cutoff velocity chosen in the previous section 5.1 for the variables as discussed in the chapter 3. Figure 5.2 shows the number of particles trapped by the 2D MOT declines to zero towards this point.



Starting of particles trapped by the 2D MOT for a total power of 100.1 mW and magnetic field gradient of $\alpha = 54\frac{\mathrm{G}}{\mathrm{cm}}$.

Figure 5.2: simulation of 2D MOT with 100k particles

For the 2D MOT I did scans to support the findings from the experiment. I run my simulations for multiple parameter sets and count the particles trapped. Particles are seen as trapped here if they are trapped in the 3D MOT as described in section 5.5.
Figure 5.3 shows this scan for a magnetic field of $\alpha = 40\frac{\mathrm{G}}{\mathrm{cm}}$. It is a heat-map of the percentage of particles captured to particles simulated. In general the figure shows how for higher power more particles are trapped. It further implies a maximum for a detuning in the area of $-1.0 \cdot \Gamma_{399}$ to $-1.1 \cdot \Gamma_{399}$.

Figure 5.4 shows the experimental findings for this same scan. As can be seen I find a very similar results with the a similar shape. Key differences are that my scan is slightly shifted towards lower detunings and that in my scan the 2D MOT efficiency drops faster for lower power.
The differences for the power can be explained if either the beam width used in my code is not perfectly accurate, as this is squared in the calculation of the intensity, or if the power distribution used in my code is different from the one in the experiment.
My scan however was done for a very different magnetic field gradient α. The gradient given from the experiment at $\alpha = 54\frac{\mathrm{G}}{\mathrm{cm}}$ is determined through some calculation the group did. My scan suggest the gradient to be much smaller though. This would be consistent with some scans shown in appendix D. Here the maximum moves towards higher detuning for smaller B fields. As a smaller B-field means a smaller Zeeman shift in the energy levels. Higher detuning would then compensate for this.
My simulations therefore validate the findings of the group but suggest that the magnetic field gradient

is significantly smaller then previously thought. Smaller then $\alpha = 40\frac{\mathrm{G}}{\mathrm{cm}}$ even, as I still find a shift towards lower detuning in my scan.

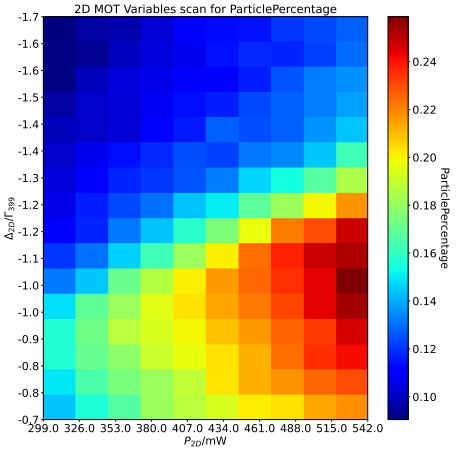
Default Parameters Table

Variable	Value
blue k-vector [1/cm]	1.575e+05
blue gamma [Hz]	1.828e+08
2D MOT detuning [Hz]	-3.55e+07
3D MOT detuning [Hz]	-4.00e+07
pushbeam detuning [Hz]	1.75e+07
2D MOT waist [cm]	0.90
3D MOT waist [cm]	0.90
pushbeam waist [cm]	0.09
tube position [cm]	9.61
tube length [cm]	16.50
tube radius [cm]	0.15
2D MOT B-field gradient [G/cm]	40.0
3D MOT B-field gradient [G/cm]	12.0

Default Powers Table

Beam	P [mW]
2D MOT total:	472.55
2D MOT 1 X	184.1
2D MOT 1 Y	136.3
2D MOT 2 X	37.09
2D MOT 2 Y	17.2
2D MOT 3 X	34.4
2D MOT 3 Y	59.67
2D MOT 4 X	2.198
2D MOT 4 Y	1.593
3D MOT total:	40.0
3D MOT horizontal	16.0
3D MOT vertical	8.0
pushbeam:	0.041

The default parameters used to initialize the simulations. The 550 mW power of the 2D MOT is already adjusted for losses here.



Percentage of particles trapped in the 3D MOT for different 2D MOT parameters.

Figure 5.3: Variable scan of the 2D MOT

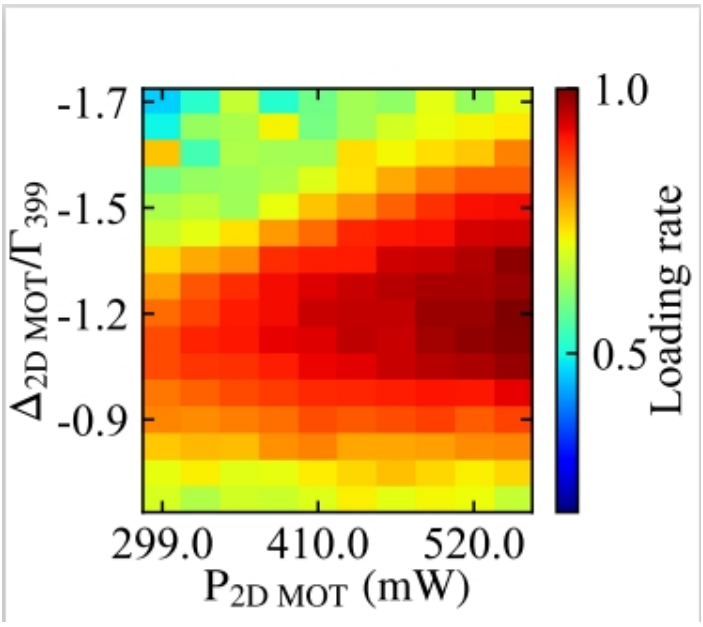


Figure 5.4: Variable scan of the 2D MOT from the experiment.

5.3 Pushbeam

I added the pushbeam used in the experiment to push the particles forward towards the science chamber. This beam is just a blue laser along the z-axis with a positive detuning and low power. I added the option to turn this beam on or off in the simulations.

Figure 5.5 shows the velocity of atoms two centimeters behind the tube with and without pushbeam. One can see how the velocities are higher when the pushbeam is turned on. Furthermore, more particles made it through the tube when the pushbeam was on. Lastly, the atoms seem to accumulate at certain velocity range around $v = 35 \frac{\mathrm{m}}{\mathrm{s}}$.
From the first two observations, I conclude that the idea of the pushbeam to accelerate atoms towards the science chamber, in order to prevent them from falling down through gravity before they enter the 3D MOT, is indeed confirmed.
The third observation can be explained because the atoms, accelerated by the pushbeam, eventually reach velocities high enough to where they are far away from resonance. This lowers the rate at which the atoms absorb new photons and thus the acceleration. The velocity at which this happens is the same for all atoms, because the transition in the atom is the same. This causes the peak around the before mentioned 35 $\frac{\mathrm{m}}{\mathrm{s}}$.

We scan power and detuning of the pushbeam as well. This will also be compared to the ex-

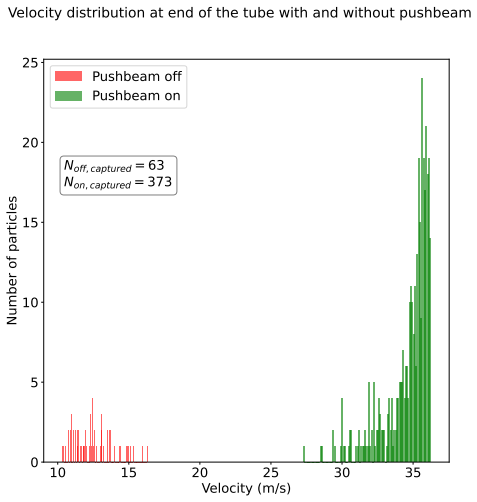
Default Parameters Table

Variable	Value
blue k-vector [1/cm]	1.575e+05
blue gamma [Hz]	1.828e+08
2D MOT detuning [Hz]	-3.55e+07
3D MOT detuning [Hz]	-4.00e+07
pushbeam detuning [Hz]	1.75e+07
2D MOT waist [cm]	0.90
3D MOT waist [cm]	0.90
pushbeam waist [cm]	0.09
tube position [cm]	9.61
tube length [cm]	16.50
tube radius [cm]	0.15
2D MOT B-field gradient [G/cm]	54.0
3D MOT B-field gradient [G/cm]	12.0

Default Powers Table

Beam	P [mW]
2D MOT total:	472.55
2D MOT 1 X	184.1
2D MOT 1 Y	136.3
2D MOT 2 X	37.09
2D MOT 2 Y	17.2
2D MOT 3 X	34.4
2D MOT 3 Y	59.67
2D MOT 4 X	2.198
2D MOT 4 Y	1.593
3D MOT total:	40.0
3D MOT horizontal	16.0
3D MOT vertical	8.0
pushbeam:	0.041

Default parameters used to initialize the simulation. The 550 mW power of the 2D MOT already is adjusted for losses here.



Comparison of velocity distribution at the end of the 2D chamber simulation with and without pushbeam

Figure 5.5: effect of the pushbeam on particle velocities

periment.

Figure 5.6 shows this scan. From high power and low detuning onwards we see a diagonal area, where particles are trapped more often. This is very similar to the scan done by the group as

seen in figure 5.7. The range I scanned was done for a slightly lower power range but the same trend is very much visible.
My scan is less smooth than the experimental results. The difference in the number of particles captured in my simulation is also not very large, as the difference in the number of particles captured is only a few tens of particles. From this I can conclude that in the future more particles need to be simulated for a pushbeam scan.

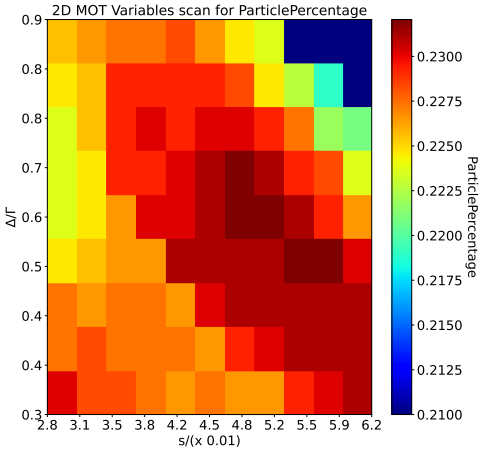
Default Parameters Table

Variable	Value
blue k-vector [1/cm]	1.575e+05
blue gamma [Hz]	1.828e+08
2D MOT detuning [Hz]	-3.55e+07
3D MOT detuning [Hz]	-4.00e+07
pushbeam detuning [Hz]	1.75e+07
2D MOT waist [cm]	0.90
3D MOT waist [cm]	0.90
pushbeam waist [cm]	0.09
tube position [cm]	9.61
tube length [cm]	16.50
tube radius [cm]	0.15
2D MOT B-field gradient [G/cm]	54.0
3D MOT B-field gradient [G/cm]	12.0

Default Powers Table

Beam	P [mW]
2D MOT total:	343.65
2D MOT 1 X	133.9
2D MOT 1 Y	99.1
2D MOT 2 X	26.97
2D MOT 2 Y	12.51
2D MOT 3 X	25.01
2D MOT 3 Y	43.4
2D MOT 4 X	1.598
2D MOT 4 Y	1.159
3D MOT total:	40.0
3D MOT horizontal	16.0
3D MOT vertical	8.0
pushbeam:	0.041

The default parameters used to initialize the simulations. The 400 mW power of the 2D MOT already is adjusted for losses here.



Percentage of particles trapped in the 3D MOT for different push beam parameters.
The colorbar is rescaled. The 4 lower values in the top right are infact lower then shown here, being around 0.15 percent of particles trapped. This however caused the rest of the plot to be too equalized to clearly see differences.

Figure 5.6: Variable scan of the pushbeam

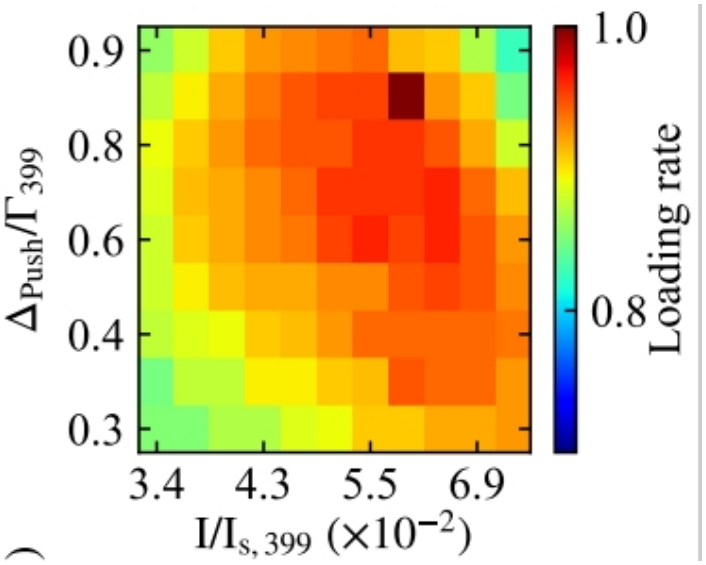


Figure 5.7: The experimental findings of the loading rate in the 3D MOT for the same parameters as figure 5.6

5.4 Tube

The tube is defined purely through events. Once a particle enters the tube it gets assigned a flag to signify this.
A particle moving in the tube is then simulated up until it either makes it out of the tube on the other side, or the simulation is stopped because a particle exits the area that is designated to be the tube.

After the 2D simulation with tube, every particle gets assigned two flags. One tells us whether the particle made it into the science chamber is therefore successfully captured by the 2D MOTs and one tells us if the particle was lost in the tube.

One has to be careful with parameter scans of the tube. After all the size and length of the tube do not just determine what particles go through but moreover it keeps the pressure low. This might be affected by such changes so a scan can not be directly translated to real changes in the experiment.

5.5 3D MOT

As described above I start a 3D MOT simulation after the 2D MOT simulation separately. This also allows me to extract data from the 2D MOT simulation before starting this simulation.

For the 3D simulation I shift into a frame where the laser beams define the axes and where the MOT center is the zero of the coordinate system. This saves me from having to write a function of a rotated and shifted magnetic field.
The starting positions and velocities of the particles are shifted into this frame using a rotational

matrix. After the simulation is done they are shifted back into the lab frame again.

To determine whether a particle is eventually captured or not, one event tracks whether particles are lost because they fly out too fast, and one looks at whether a particle has slowed down enough and is close enough to the centre to be considered captured. Again, I separate these particles using flags.

Figure 5.8 shows the starting velocities of particles that get captured by both MOTs. It further shows the starting velocities of all particles simulated. One can see that the the particles captured by the 2D and 3D MOT are the same. From repetitions of this simulation I found that for the default 3D MOT parameters all particles that make it through the tube get trapped. Particles that would have a z velocity component too high to be trapped by the 3D MOT enter the lower power beams in the 2D MOT to quickly to be trapped anyway. The combination of the beam power distribution and the tube therefore functions as a good filter for the 3D MOT.

With the 3D MOT I can now simulate the entire setup. Examples are shown in figure 5.9 and figure 5.10.

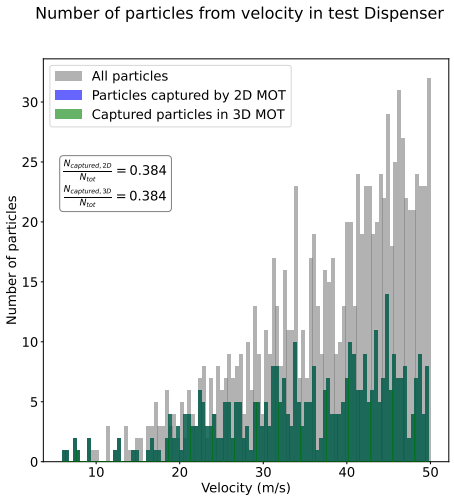
Default Parameters Table

Variable	Value
blue k-vector [1/cm]	1.575e+05
blue gamma [Hz]	1.828e+08
2D MOT detuning [Hz]	-3.55e+07
3D MOT detuning [Hz]	-4.00e+07
pushbeam detuning [Hz]	1.75e+07
2D MOT waist [cm]	0.90
3D MOT waist [cm]	0.90
pushbeam waist [cm]	0.09
tube position [cm]	9.61
tube length [cm]	16.50
tube radius [cm]	0.15
2D MOT B-field gradient [G/cm]	54.0
3D MOT B-field gradient [G/cm]	12.0

Default Powers Table

Beam	P [mW]
2D MOT total:	472.55
2D MOT 1 X	184.1
2D MOT 1 Y	136.3
2D MOT 2 X	37.09
2D MOT 2 Y	17.2
2D MOT 3 X	34.4
2D MOT 3 Y	59.67
2D MOT 4 X	2.198
2D MOT 4 Y	1.593
3D MOT total:	40.0
3D MOT horizontal	16.0
3D MOT vertical	8.0
pushbeam:	0.041

Default parameters used in the simulation.



Starting velocities compared. Velocities captured by the 2D and 3D MOT perfectly overlap and thus are the same.

Figure 5.8: Starting velocities off all particles compared to the ones captured by 2D and 3D MOT

Code to scan the variables of the 3D MOT has been added. I did not however find time to run these scans yet.

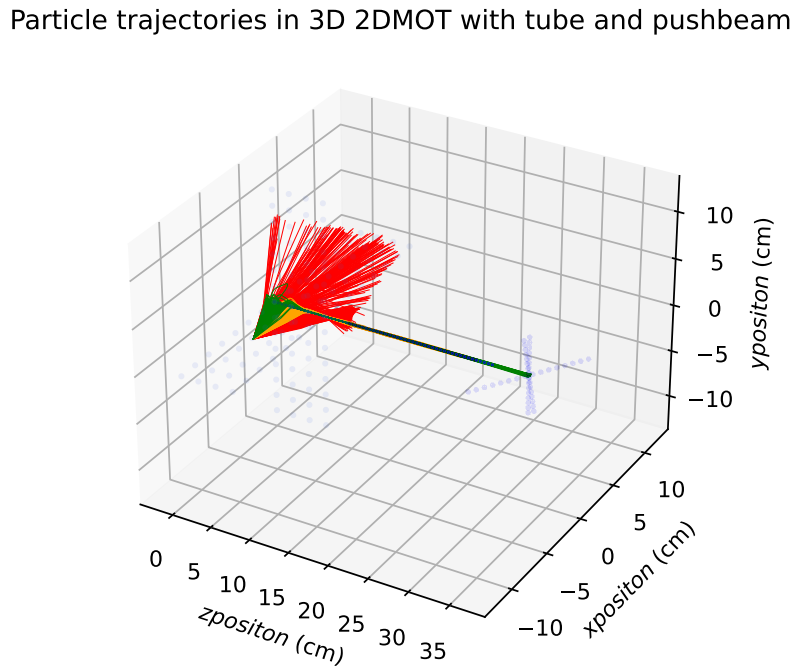


Figure 5.9: Trajectories of the particles as they move through the setup. Again the blue dots represent the lasers. Red trajectories are lost, orange are lost in the tube and green are captured.
The simulation uses the variables from figure 5.8

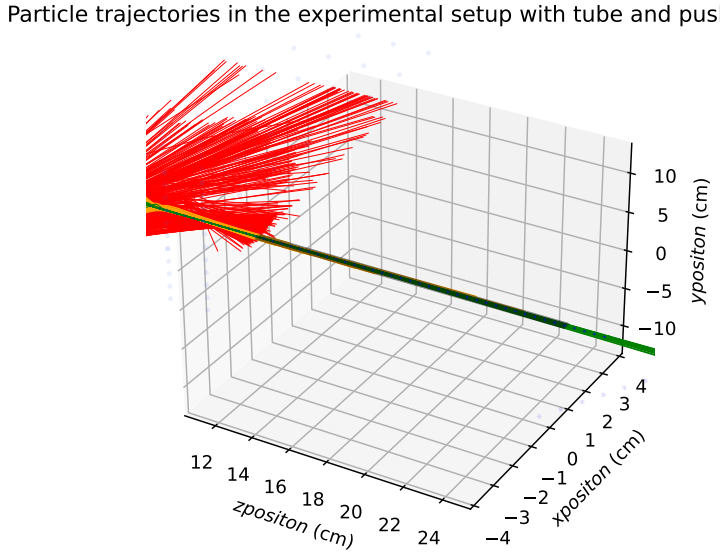


Figure 5.10: Another simulation with the same parameters zoomed in on the tube.

CHAPTER 6

Summary and outlook

In this thesis I have created a program build on the PyLCP library that allows one to simulate particles as they move through the first phases of the YQO group's experimental setup. It allows one to compare parameters to maximise the number of atoms captured.
I made changes to the heuristic model to handle saturation better. I also concluded that for the purposes of the YQO setup, the heuristic model was sufficient to simulate accurately.

With this knowledge, I set about simulating the experimental setup and was able to compare my results with the experiment. From this I was able to conclude that the general behaviour of the atomic number for different parameters found in the experiment seemed to match that of my simulations, but that the magnetic field as calculated by the group might not be accurate.
I could also see changes in the atom trajectories and velocities directly for changes in the setup.

Picking up from this point, one could imagine implementing the green 3D MOT. PyLCP allows time-dependent detunings and intensities, which can be used to mimic the frequency broadening in the experiment. From this, the velocity distribution at the end of the process could be used to calculate the temperature of the atomic ensemble. One would have to move to a computer cluster and use its many cores for the simulations in order to achieve usable computing times.

Automatic implementation of other setups via an input can also be implemented. Here one would have to overcome the problem that simulations for multiple transitions are not supported in the heuristic model and implement arbitrarily rotatable magnetic fields and a comprehensive user interface.
In addition, some automation could be implemented that uses previous results from simulations to narrow scan ranges to find parameters that are as accurate as possible.

Finally, one could still ask whether this library is worth using in all cases. For the YQO setup, only the heuristic equation was needed. This could be implemented in a programming language that is faster than Python. Alternatives already exist in C++ as used by for example [15].

APPENDIX A

Attempts at parallelization of the heuristic equation

In order to gain computational speed, one has to somehow parallelise the simulation process, because the simulation of a single particle already takes fairly long.
For this I have tried two different methods. In one I tried to change PyLCP in order to work with a larger vector in which every three entries represent the position/ velocity of one particle, in the other we use multiprocessing as an option in python.

The first option meant going through the PyLCP source code and both adding the options for larger vectors everywhere and changing certain calculations, which would now break seeing as how they were written for vectors of length 3.
After doing this for the heuristic model though, I noticed that there was not a very large increase in time gained. I tested it on a simulation of the 2D MOT chamber setup and the time per particle went down from 8.8s to 6s. The problem with this is that this only occurs after the vector reaches sizes of around 10 particles or larger. This means that we only gain about 33% in computational speed compared to solving the particles one after another. Because we later on want to be solving about 1000 particles per simulation we would still need 6000 s or over an hour with this implementation for one single simulation with the heuristic equation.
The reason for this relative lack in gain of speed is that the calculations can not be done purely in parallel, as several vectors containing various information about the beams need to be normalised and rotated around certain axes. This cannot be done for one large vector while keeping the solutions independent from one another.
This problem will also exist for both the OBE and rate equation models, so I have not bothered implementing this for those.

The second option is multiprocessing. For this we can use the pathos.multiprocessing library, which allows us to split the calculation among the cores of our computer. This is obviously limited by the amount of cores the computer we use has. It also slows down the calculations per particle slightly:

Table A.1: computational time while multiprocessing

$N_{particles}$	$t_{computational}$ [s]
1	8.84
4	9.67
8	10.93
16	14.47

The time gained however is very large seeing as how this is truly parallel, meaning we can reduce the runtime for one simulation to about 10 minutes.

Overall the first attempt at using a different representation of the particles is not worth the effort in terms of how much time is gained. However using multiprocessing is definitely worth it since it is both easier to implement and gives us much better efficiency.

APPENDIX B

Additional pictures of the setup

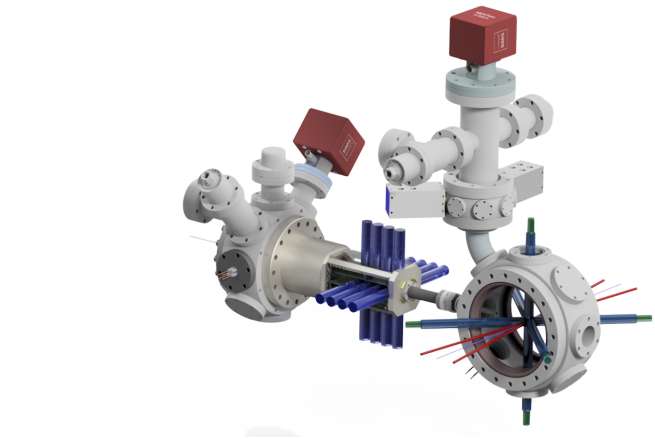


Figure B.1: Three dimensional render of the experimental setup with laser beams drawn in. Created by Thilina Senaviratne.

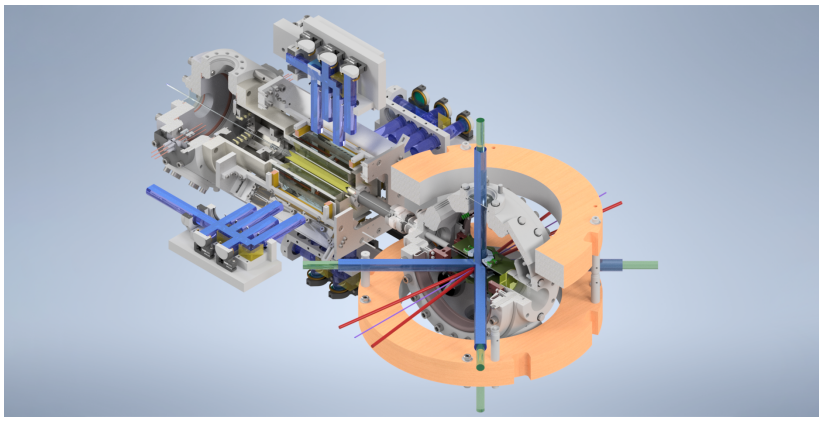


Figure B.2: Closer look at an opened up view of the setup. Includes magnetic coils. Created by Thilina Senaviratne.

APPENDIX C

Additional plots for the simulations

Following plots are for 10802 particles simulated with $P_{2D} = 100.1\,\text{mW}$:

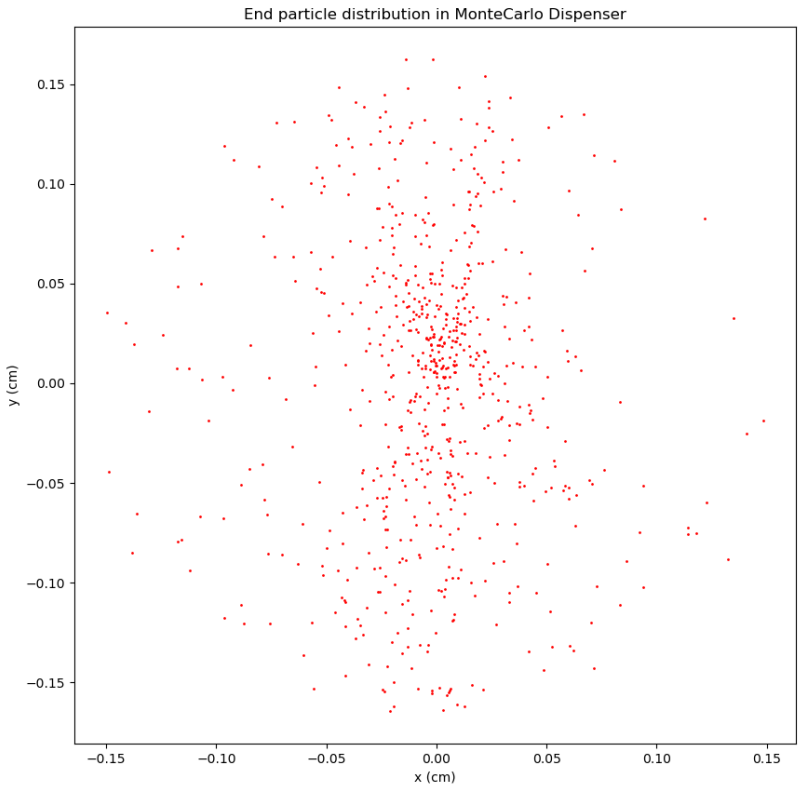


Figure C.1: Cross-section of the particle trajectories 2 cm after the tube for . The particles are distributed wider along the y-axis where the power of the 2D MOT beams are lower.

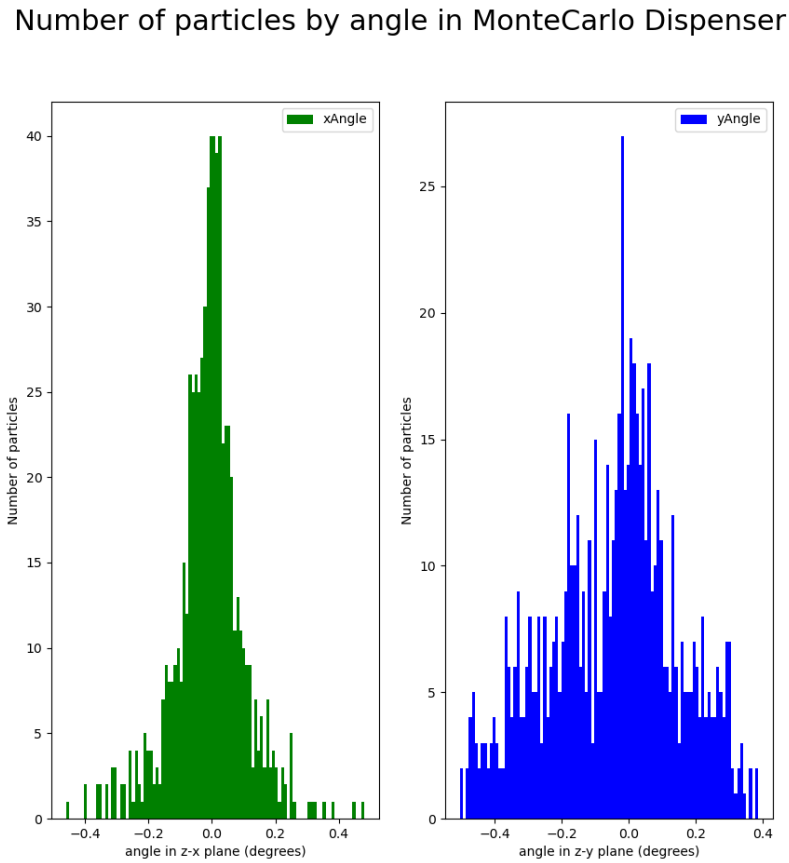


Figure C.2: Angles at which the particles leave the tube. They are very small and thus the particles leave the tube more or less parallel to it.

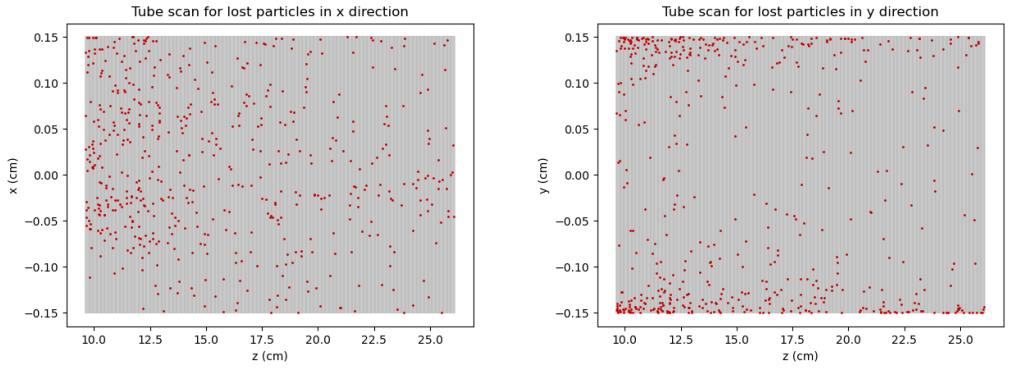


Figure C.3: Plots that show where the particles get lost in the tube along the x and y axis. Particle tend to get lost at the start of the tube more often. Furthermore particles tend to get lost at the top or bottom of the tube and tend to not get lost at the start of the tube on the side facing the dispenser.

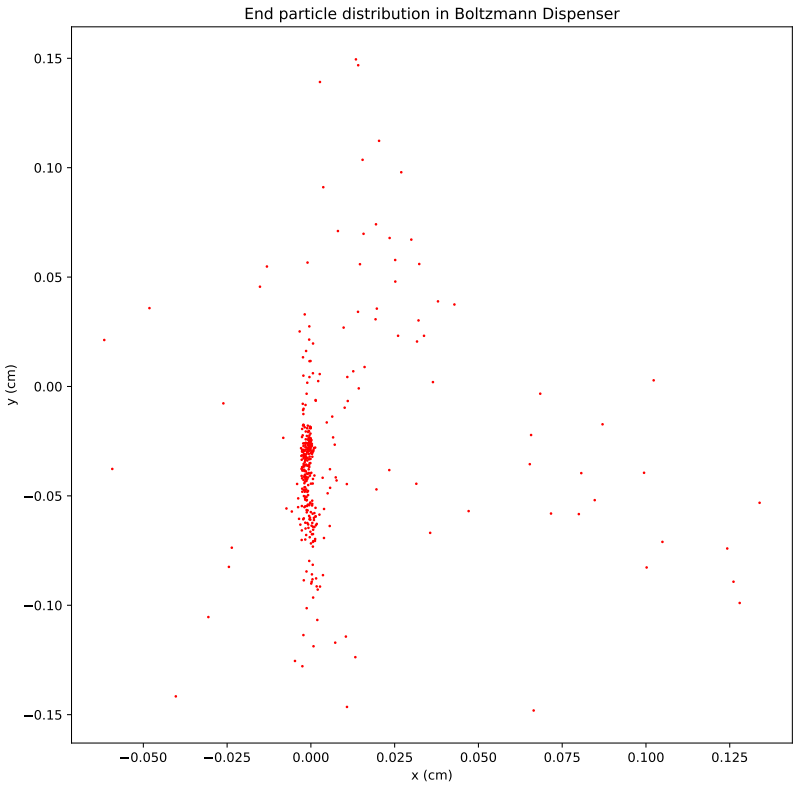


Figure C.4: Particle distribution for a larger 2D MOT power of $P_{2D} = 550\,\mathrm{mW}$. One can see how the particles are more squished together now. One can also see the effect of gravity better as the particles have fallen down a bit.

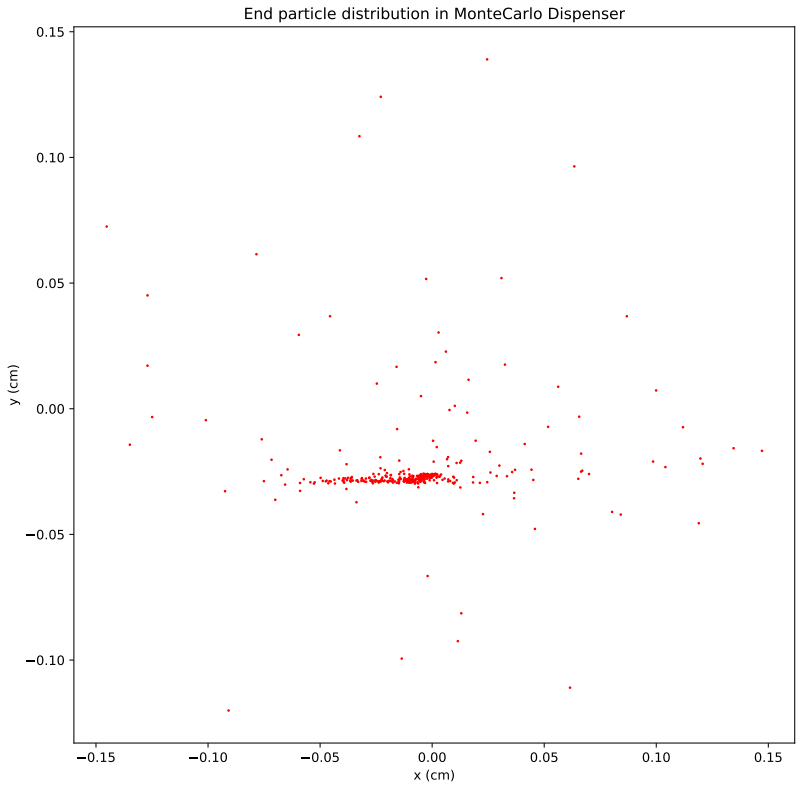


Figure C.5: Here I swapped the intensities of the y and x axis in the 2D MOT. We can see how the distribution shifts to be more spread among the lower intensity axis accordingly. We can also see that the particles are moved to one side a little. This depends on what side we simulate the particles from as we see in C.6

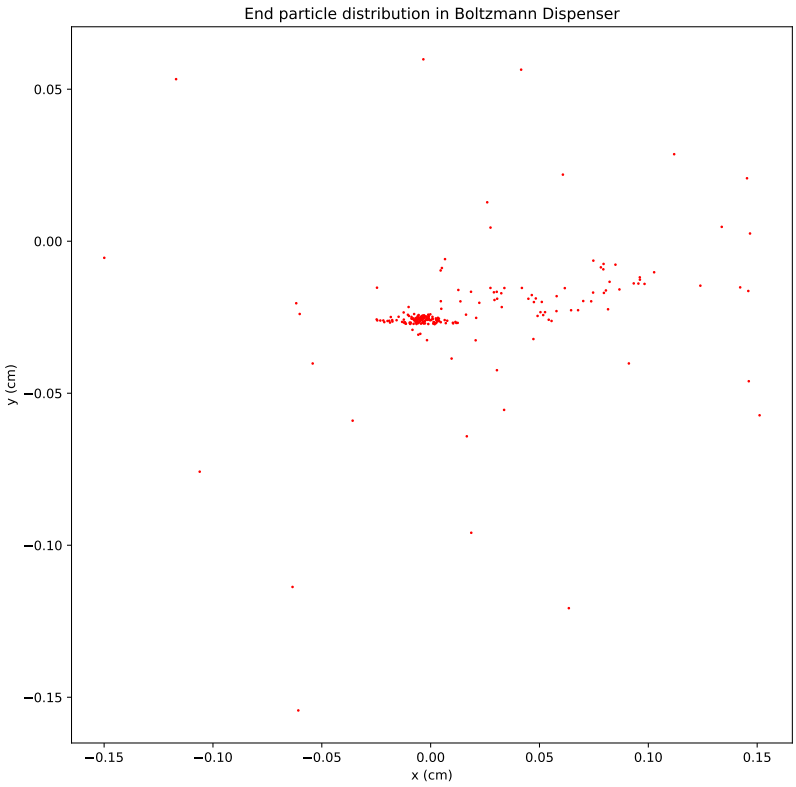


Figure C.6: Here I swapped the dispenser location to the other side of the setup. We get a mirrored distribution of C.5

APPENDIX D

Additional scans

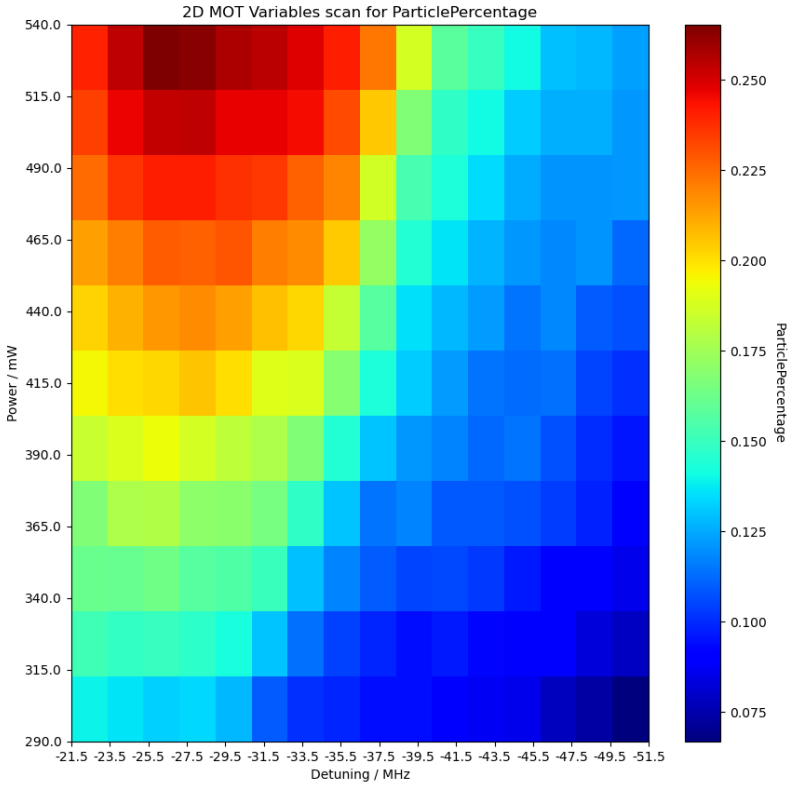


Figure D.1: A scan of the 2D MOT for $\alpha = 54 \frac{\mathrm{G}}{\mathrm{cm}}$

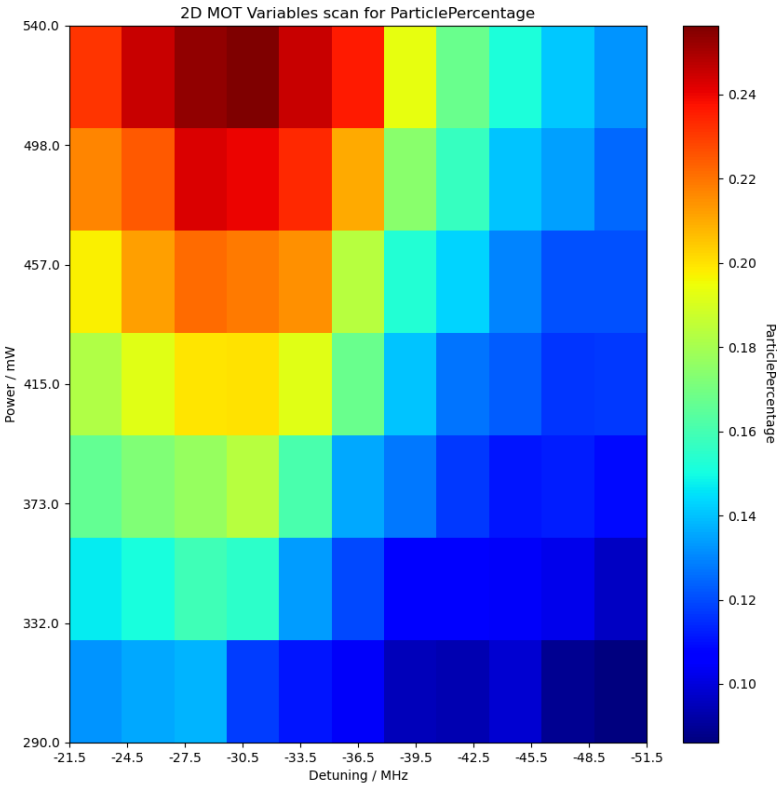


Figure D.2: A scan of the 2D MOT for $\alpha = 50\frac{\mathrm{G}}{\mathrm{cm}}$ in a smaller range. One can see how the maximum moves towards higher detunings

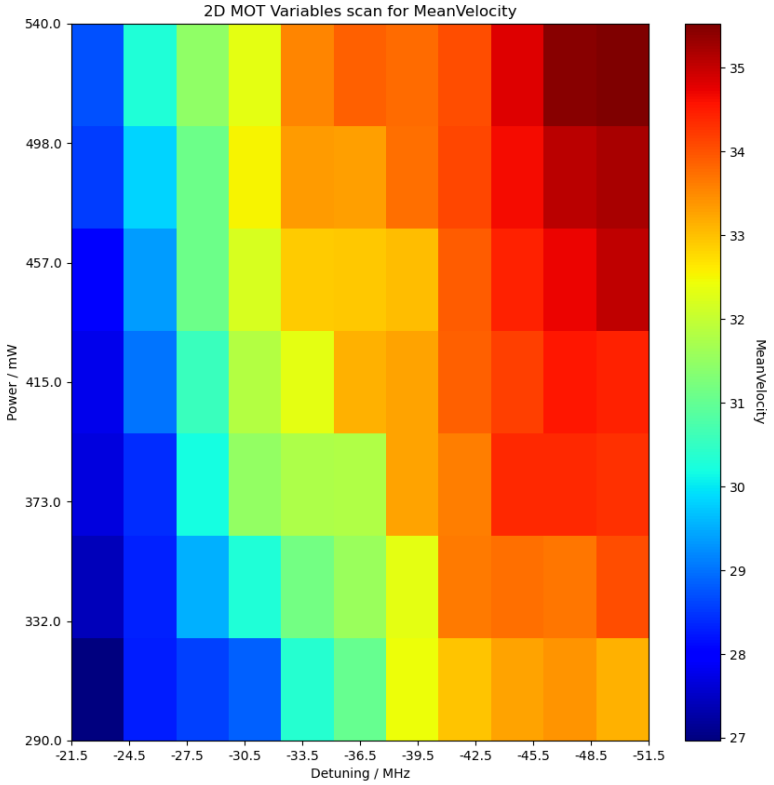


Figure D.3: The corresponding mean starting velocities of the particles of the scan above. Higher power traps all particles better, while higher detuning means particles of higher velocity are closer to resonance and are thus more likely to scatter of the beams, meaning that they get trapped more often.

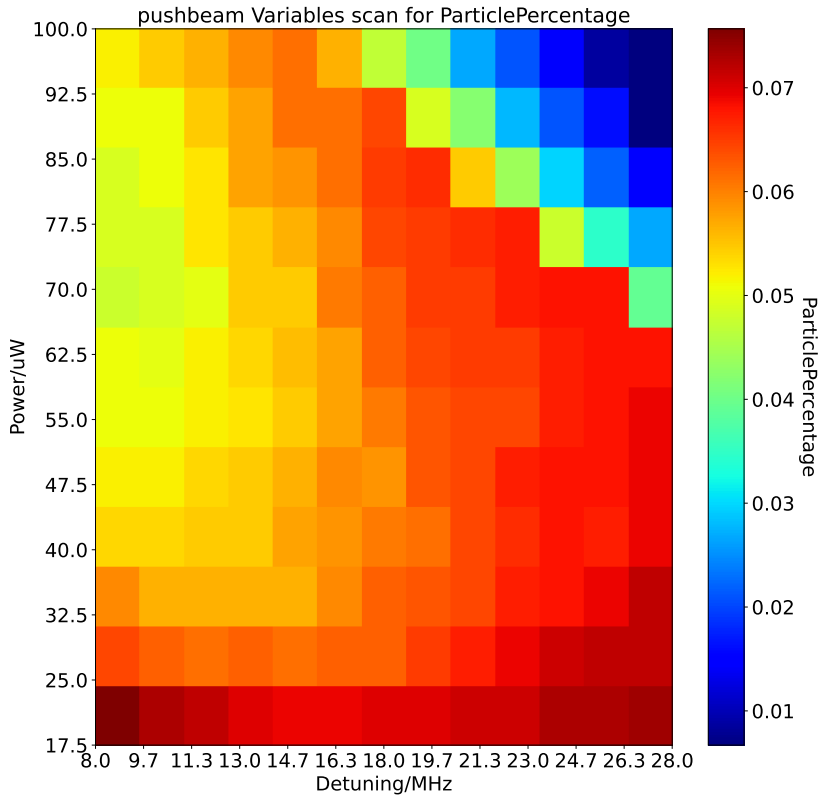


Figure D.4: Larger scan of the pushbeam for a waist of $w_b = 0.1$ cm and 2D MOT power of $P_{2D} = 100.1$ mW, lower power seems to be more efficient in this case.
We do also see a diagonal band where the pushbeam is more effective. This corresponds well with the findings of the group.

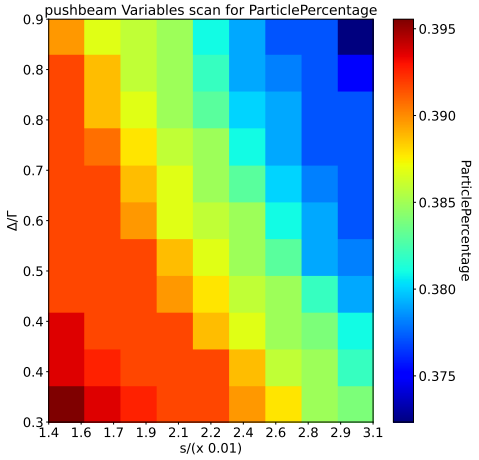
Default Parameters Table

Variable	Value
blue k-vector [1/cm]	1.575e+05
blue gamma [Hz]	1.828e+08
2D MOT detuning [Hz]	-3.55e+07
3D MOT detuning [Hz]	-4.00e+07
pushbeam detuning [Hz]	1.75e+07
2D MOT waist [cm]	0.90
3D MOT waist [cm]	0.90
pushbeam waist [cm]	0.15
tube position [cm]	9.61
tube length [cm]	16.50
tube radius [cm]	0.15
2D MOT B-field gradient [G/cm]	54.0
3D MOT B-field gradient [G/cm]	12.0

Default Powers Table

Beam	P [mW]
2D MOT total:	472.55
2D MOT 1 X	184.1
2D MOT 1 Y	136.3
2D MOT 2 X	37.09
2D MOT 2 Y	17.2
2D MOT 3 X	34.4
2D MOT 3 Y	59.67
2D MOT 4 X	2.198
2D MOT 4 Y	1.593
3D MOT total:	40.0
3D MOT horizontal	16.0
3D MOT vertical	8.0
pushbeam:	0.041

The default parameters used to initialize the simulations. The 550 mW power of the 2D MOT already is adjusted for losses here.



Percentage of particles trapped in the 3D MOT for different tube beam parameters

Figure D.5: Variable scan of the pushbeam for a beamwidth of $w_B = 0.15$ mm.

Bibliography

[1] M. D. Lukin et al.,
Dipole Blockade and Quantum Information Processing in Mesoscopic Atomic Ensembles,
Phys. Rev. Lett. **87** (3 2001) 037901,
URL: https://link.aps.org/doi/10.1103/PhysRevLett.87.037901 (cit. on p. 1).

[2] J. D. Pritchard et al., *Cooperative Atom-Light Interaction in a Blockaded Rydberg Ensemble*,
Phys. Rev. Lett. **105** (19 2010) 193603,
URL: https://link.aps.org/doi/10.1103/PhysRevLett.105.193603 (cit. on p. 1).

[3] S. Eckel, D. S. Barker, E. B. Norrgard and J. Scherschligt,
PyLCP: A python package for computing laser cooling physics, 2020,
arXiv: 2011.07979 [physics.atom-ph] (cit. on pp. 1, 8, 9, 16, 17).

[4] W. Demtroder, *Experimentalphysik 3: Atome, Molekule Und Festkorper*, eng,
4., uberarb. Aufl. 2010 edition., Springer-lehrbuch, Springer, 2010, ISBN: 3642039103
(cit. on p. 2).

[5] J. Estève, *Trapped by nanostructures*, Nature Nanotech **8** (2013) 317,
URL: https://www.nature.com/articles/nnano.2013.80#citeas (cit. on p. 4).

[6] H. J. Metcalf and P. van der Straten, *Laser Cooling and Trapping*, 4th ed., Springer, 1999
(cit. on pp. 4–6, 8, 12).

[7] J. P. Gordon and A. Ashkin, *Motion of atoms in a radiation trap*,
Phys. Rev. A **21** (5 1980) 1606,
URL: https://link.aps.org/doi/10.1103/PhysRevA.21.1606 (cit. on pp. 9, 11).

[8] P. J. Ungar, D. S. Weiss, E. Riis and S. Chu, *Optical molasses and multilevel atoms: theory*,
J. Opt. Soc. Am. B **6** (1989) 2058,
URL: https://opg.optica.org/josab/abstract.cfm?URI=josab-6-11-2058
(cit. on p. 9).

[9] A. Einstein, *Strahlungs-Emission und Absorption nach der Quantentheorie*,
Deutsche Physikalische Gesellschaft **18** (1916) 318 (cit. on p. 10).

[10] M. R. Tarbutt,
Magneto-optical trapping forces for atoms and molecules with complex level structures,
New Journal of Physics **17** (2015) 015007,
URL: https://dx.doi.org/10.1088/1367-2630/17/1/015007 (cit. on p. 10).

[11] P. Lunt, *Design and construction of a new ultracold ytterbium experiment for rydberg physics*, 2019 26 (cit. on p. 14).

[12] C. Runge, *Ueber die numerische Aufloesung von Differentialgleichungen*, 2018, URL: https://doi.org/10.1007/bf01446807 (cit. on p. 16).

[13] O. Schlömilch et al., *Zeitschrift für Mathematik und Physik*, Bd. 46, B. G. Teubner., 1901 167, URL: https://books.google.de/books?id=QL0KAAAAIAAJ (cit. on p. 16).

[14] S. Eckel, D. S. Barker, E. B. Norrgard and J. Scherschligt, *Real atoms in a MOT*, URL: https://python-laser-cooling-physics.readthedocs.io/en/latest/examples/MOTs/06_real_atoms_3D_MOT.html (cit. on p. 17).

[15] S. E. Doerscher, *Creation of ytterbium quantum gases with a compact 2D-/3D-MOT setup*, 2013 119 (cit. on pp. 24, 33).

List of Figures

2.1 Basic schematic of the setup of a MOT along one axis with varying energies for transitions over the position z, as seen in [5] . . . 4
2.2 Exemplary Bloch sphere [6]. . . . 5
2.3 Probability $|c_e(t)|^2$ for the atom to be in the exited state for $\Omega = \gamma$ and $\delta = \gamma$ with time in units of $\frac{1}{\gamma}$, calculated numerically from the OBEs by [6]. . . . 12

3.1 Schematic of the setup as used in the YQO group. Picture created by Thilina Senaviratne, who is a part of the group. . . . 13
3.2 Picture of the 2D MOT with the tube in the back as atoms come out of the dispenser. The cones are lit up slightly in the picture. . . . 14

4.1 Particle simulation for different allowed maximum timesteps. . . . 18
4.2 Comparison of heuristic methods for a blue 1D MOT. Heuristic total describes the default equation PyLCP uses, while Heuristic separated describes the saturation of each beam being applied individually. . . . 19
4.3 Comparison of heuristic methods for a blue 1D MOT. Again default is the default used by PyLCP, while separated mean applying the saturation individually per beam. 20
4.4 Comparison of the 3 models for parameters like the blue MOTs used in the experiment without random recoil (left) and with random recoil (right). . . . 21
4.5 Comparison of velocities for parameters like the green MOT used in the experiment without random recoil (left) and with random recoil (right). . . . 21

5.1 Exemplary velocity distribution . . . 24
5.2 simulation of 2D MOT with 100k particles . . . 26
5.3 Variable scan of the 2D MOT . . . 27
5.4 Variable scan of the 2D MOT from the experiment. . . . 27
5.5 effect of the pushbeam on particle velocities . . . 28
5.6 Variable scan of the pushbeam . . . 29
5.7 The experimental findings of the loading rate in the 3D MOT for the same parameters as figure 5.6 . . . 30
5.8 Starting velocities off all particles compared to the ones captured by 2D and 3D MOT 31
5.9 Trajectories of the particles as they move through the setup. Again the blue dots represent the lasers. Red trajectories are lost, orange are lost in the tube and green are captured. The simulation uses the variables from figure 5.8 . . . 32
5.10 Another simulation with the same parameters zoomed in on the tube. . . . 32

B.1 Three dimensional render of the experimental setup with laser beams drawn in. Created by Thilina Senaviratne. 36
B.2 Closer look at an opened up view of the setup. Includes magnetic coils. Created by Thilina Senaviratne. 36

C.1 Cross-section of the particle trajectories 2 cm after the tube for . The particles are distributed wider along the y-axis where the power of the 2D MOT beams are lower. 37
C.2 Angles at which the particles leave the tube. They are very small and thus the particles leave the tube more or less parallel to it. 38
C.3 Plots that show where the particles get lost in the tube along the x and y axis. Particle tend to get lost at the start of the tube more often. Furthermore particles tend to get lost at the top or bottom of the tube and tend to not get lost at the start of the tube on the side facing the dispenser. 38
C.4 Particle distribution for a larger 2D MOT power of $P_{2D} = 550\,\mathrm{mW}$. One can see how the particles are more squished together now. One can also see the effect of gravity better as the particles have fallen down a bit. 39
C.5 Here I swapped the intensities of the y and x axis in the 2D MOT. We can see how the distribution shifts to be more spread among the lower intensity axis accordingly. We can also see that the particles are moved to one side a little. This depends on what side we simulate the particles from as we see in C.6 40
C.6 Here I swapped the dispenser location to the other side of the setup. We get a mirrored distribution of C.5 41

D.1 A scan of the 2D MOT for $\alpha = 54\frac{\mathrm{G}}{\mathrm{cm}}$ 42
D.2 A scan of the 2D MOT for $\alpha = 50\frac{\mathrm{G}}{\mathrm{cm}}$ in a smaller range. One can see how the maximum moves towards higher detunings 43
D.3 The corresponding mean starting velocities of the particles of the scan above. Higher power traps all particles better, while higher detuning means particles of higher velocity are closer to resonance and are thus more likely to scatter of the beams, meaning that they get trapped more often. 44
D.4 Larger scan of the pushbeam for a waist of $w_b = 0.1\,\mathrm{cm}$ and 2D MOT power of $P_{2D} = 100.1\,\mathrm{mW}$, lower power seems to be more efficient in this case. We do also see a diagonal band where the pushbeam is more effective. This corresponds well with the findings of the group. 45
D.5 Variable scan of the pushbeam for a beamwidth of $w_B = 0.15\,\mathrm{mm}$. 46

List of Tables

3.1 Laser power distribution of the 2D MOT 15

A.1 computational time while multiprocessing 35