Progress towards the Study of Li+Rb Ultra-cold Collisions

by

Tao Kong

B.Sc., Shandong University, 2000
M.Sc., Shandong University, 2003

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in

The Faculty of Graduate Studies

(Physics)

The University of British Columbia

April, 2006

Abstract

The work described in this thesis is related to two projects I worked on towards the ultracold collisions experiment between fermionic lithium and bosonic rubidium atoms in our lab.

The first part is about the simulation of sympathetic cooling of Bose-Fermi mixture inside the magnetic trap. An original single species code, which was specially designed for simulating the forced radio-frequency evaporation of ^{133}Cs atoms in the magnetic trap (within harmonic approximation), is developed so that it can simulate the dynamics of a multi-species trap including inter-atomic collisions and non-harmonic trapping potentials, and thus aid in the design of our future degenerate gas experiments. To validate the consistence between our multi-species code and the original single-species code, a comparison was tried using the parameters from Clauz Zimmermann's sympathetic cooling of fermionic Lithium using Rubidium atoms. The result shows a very good consistence between the multi-species and single species codes. Based on the same parameters from Zimmermann's experiment, another comparison is performed to check the influence of different macro-atom distributions on the simulation results when the total number of real atoms is constant, since the number of macro-atoms determines how fast the simulation can run. The result shows that using less number of macro-atoms (2^{11} vs 2^{13}) won't effect the simulation too much and we can use less number of macro-atoms to speed up the simulation.

The second part of this thesis is concerned with the design of the experimental apparatus to study hetero-nuclear Feshbach resonances between Lithium and Rubidium. The strategy for this study was discussed in the experimental setup section, in which the required experimental subsystems for the Feshbach resonance studies were discussed, such as the vacuum system, the atomic sources for both species. This work also involved some experiments to calibrate the alkali metal dispensers to be used in this apparatus. A model was created to describe the Rubidium partial pressure in room temperature, parameters required by this model was calibrated by an absorption experiment, in which the Rubidium resonant laser was sent through a vapor cell.

Contents

List of Figures

Acknowledgements

I would like to take this chance to thank the people I met during my Master life in UBC Quantum Degenerate Gas laboratory.

First of all, I want to thank my thesis supervisor, Dr. Kirk W. Madison. Dr. Madison is a really nice advisor, he never hesitates to help me clarify the puzzles both academically and in my usual life.

Secondly, thank Dr. Bruce G. Klappauf, our research associate. Bruce is wonderful partner for doing experiments, also for playing Frisbee.

At last, I would like to thank all the other people once or now in our lab. Thank Swati Singh and Janelle Dongen, they helped me a lot during the usual lab work. Thank Friedrich Kirchner for his great help in LaTex programming during my thesis writing.

Chapter 1

Introduction

With the development of laser cooling and trapping techniques, atomic collisional properties in the ultracold regime have become directly accessible. Today these properties play a crucial role for the realization of Bose-Einstein condensates (BECs) and quantum degenerate samples of Fermions. An ultracold collision experiment can be used to deduce these properties and thus give guidance for the design of future experiments, such as a Bose-Fermi quantum degenerate mixture, and the production of ultracold molecules.

Until now evaporative cooling is still the only way by which quantum degeneracy could be reached. During the evaporative cooling process, the highly energetic atoms are removed from the ensemble so that the remaining atoms can rethermalize by elastic collision and the overall temperature could be lowered. However, this mechanism only works for Bosons, not for Fermions, since the dominant collision at ultralow temperature, s-wave collision, are forbidden for Fermions. To circumvent this problem the fermionic cloud can be brought into thermal contact with a cold bosonic cloud, since collisions between Fermions and Bosons are not forbidden by the Pauli-principle. The temperature of the fermionic ensemble is thus lowered down and quantum degeneracy could be reached. This cooling method is called sympathetic cooling.

1.1 Motivation for The Ultra-cold Collision Experiment between ^{6}Li and Rubidium

Mixtures of ultracold gases are a field of growing interest which enables the generation of quantum degenerate Fermi gases and leads to many interesting effects in Fermi/Bose quantum systems. In our lab, fermionic Lithium and bosonic Rubidium combination is chosen as our research topic, since this combination hasn't been investigated at the time this research project was proposed. Other reasons come from the fact that the molecular dipole moment between Lithium and Rubidium is the second largest in all kinds of alkali combinations[1]. ^{6}Li and ^{85}Rb is also an ideal system for studying the BEC-BCS crossover regime since the current experimental limit for T_F due to Fermi hole heating can be reduced with this mixture [2].

By investigating the ultra-cold collisions between ^{6}Li and Rubidium isotopes, inter-species Feshbach resonances and background scattering length can be found first. This work will be done in the experimental setup of our test chamber. In which a dual MOT for both ^{6}Li and Rubidium will be built to trap the atoms from the atomic sources, and then an optical tweezer will be used to hold these atoms for further investigation of Feshbach resonances.

Using the Feshbach resonances found, we can control the elastic collisions between ^{6}Li and Rubidium during sympathetic cooling and thus optimize the efficiency of sympathetic cooling, since scattering length is a very critical factor for evaporative cooling, and it is also the only tunable factor once the experiment setup is determined. A numerical simulation is designed to simulate this process, in which the tunability of the scattering length could be simulated and provide an independent verification for the experimental results. Also new trap geometries can also be investigated with this simulation to aid in the design of future experiment.

Besides the two species mixture of quantum degeneracy, hetero-nuclear ultra-cold molecules can also be made. Other topics, such as the many body physics of mixed Fermi-Boson systems, will also be investigated by tuning the interaction between them using Feshbach resonance.

1.2 Outline of This Thesis

This thesis is divided into two parts, corresponding to two projects I worked on for the ultracold collision experiment.

The first part is about the simulation of the sympathetic cooling of Bose-Fermi mixture inside the magnetic trap. Chapter 2 describes the related theoretical background. Chapter 3 discusses the numerical simulation and the application of this simulation to actual experiments. This code was also developed to aid in the design of future degenerate gas experiments, and as a first test of the code, comparisons between the original single species code and the new multi-species code was tried using parameters from the experiment of Claus Zimmermann [3] in which Lithium was sympathetically cooled using Rubidium.

The second part of this thesis is concerned with the design of the experimental apparatus to study hetero-nuclear Feshbach resonances between Lithium and Rubidium. This work involved some experiments to calibrate the alkali metal dispensers to be used in this apparatus. In Chapter 4, the required experimental subsystems for the Feshbach resonance studies are discussed, such as the vacuum system, the atomic sources for both species. In Chapter 5, the results for the dispenser calibration experiment are described.

Chapter 2

Theoretical Backgroud

This chapter can be divided into two parts. Section 2.1 describes the basic theory of quantum degenerate gases and evaporative cooling. Section 2.2 and 2.3 discuss some aspects of ultracold collision theory.

2.1 Quantum Degenerate Gases

2.1.1 Bosons and Fermions

Depending on the spin, all the particles in nature can be divided into two categories: Bosons or Fermions. If the total spin of the particle is integer multiple of $\hbar$, it is a Boson, for half integer it is a Fermion. For a collection of identical particles, the many body wave function is either symmetric for Bosons or antisymmetric for Fermions under the exchange of two particles. The result of this fundamental symmetry of nature is that Bosons can and are likely to occupy the same energy state, this is called the Bose enhancement. Fermions, on the other hand, can't occupy the same energy state, which is the well known Pauli Exclusion principle.

2.1.2 Bose Einstein Condensation

Bose enhancement results in a phase transition at phase space densities above $\rho = 2.612$ [4]. At this critical point, most of the atoms in the dilute bosonic gas will occupy the ground state, forming a new state of matter, a Bose Einstein Condensate. This new state of matter was predicted by S. Bose and A. Einstein in 1924.

The way to realize BEC state is difficult, due to the fact that high phase space density requires high spatial density atoms at a very low temperature. Compressing an ensemble of atoms will lead to an increase of density but at the same time increase their temperature, adiabatically expanding the atoms will decrease the temperature but also decrease the density, leaving the phase space density to remain constant. The phase space density is defined by the following expression (Eqn. 2.1).

$$\rho = n\lambda_{deB}^3 = \frac{nh^3}{(M\bar{v})^3} = \frac{nh^3}{(3Mk_BT)^{\frac{3}{2}}} \tag{2.1}$$

In 1995, three American research groups published the first results of Bose Einstein Condensation at about the same time[5, 6, 7]. They all used laser cooling to precool atoms of different types (Na, Li, and Rb). Laser cooling lowers the temperature of the atoms without changing the density and thus leads to an increase in the phase space density. However, laser cooling alone is not enough to achieve the required

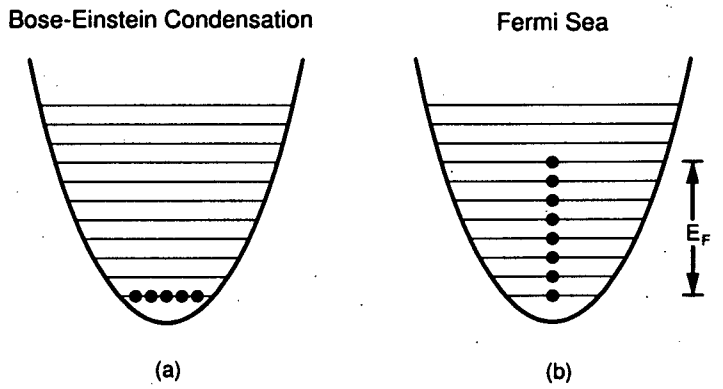


Figure 2.1: Schematics of quantum degeneracy. a: Bose Einstein Condensation (BEC), b: Fermi Sea, E_F is the Fermi energy level.

critical phase space density to form BEC, as the last step of the cooling process, they all used evaporative cooling [8].

2.1.3 Fermionic Degenerate Gas

Analogous to Bosons, Fermions will also fall into lower energy states when the temperature is decreased; however, since the Pauli Exclusion principle forbids two fermions to share the same energy level, they will not coalesce together like Bosons at zero temperature, but rather the are pile up one after another, as shown in Fig. 2.1 . When all of the lowest states are occupied up to the Fermi energy, this state is referred to as a *degenerate Fermi gas* or *Fermi sea*, it was firstly described by E. Fermi in 1926[9, 10].

At zero temperature, all the Fermions will be in the states below the Fermi energy E_F, leaving the above states empty. The Fermi energy has a corresponding temperature, Fermi temperature, $T_F = E_F/k_B$, and the degeneracy parameter T/T_F represents the deviation from the classical distribution of the Fermi Gas. Unlike

Bosons, there is no phase transition to degeneracy, and the system transits smoothly from a classical (where the state occupation is low enough that a particle's state is independent of the other particles) to non-classical behavior.

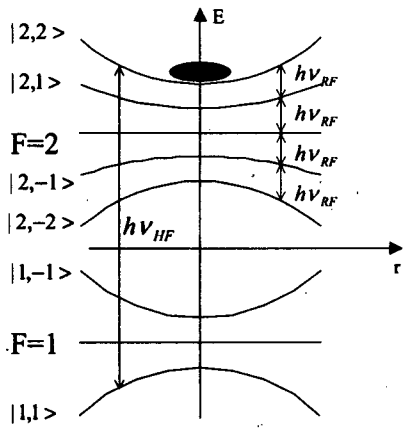


Figure 2.2: Principle of evaporative cooling. Shown is the energy of the ground state hyperfine splitting of ^{87}Rb versus the distance from the magnetic trap center, ^{87}Rb atoms are trapped in the highest $|2,2>$ state. Two kinds of evaporative cooling mechanisms are shown in this figure. One is the forced radio-frequency (ν_{RF}) evaporative cooling, the other one is the forced microwave-frequency (ν_{HF}) evaporative cooling. Radio frequency couples adjacent Zeeman sublevels, it takes several steps for atoms in $|2,2>$ states to be transferred into untrapped states. Microwave frequency couples Zeeman sublevels in different hyperfine states, it can transfer atoms directly into the untrapped state.

2.1.4 Boson-Fermi Mixture

To get Fermi degeneracy, the temperature of the Fermi gas has to be lowered to the order of T_F. Evaporative cooling has been used to cool bosons to produce a quantum degenerate gas. This technique works by removing the high energetic atoms, and then allowing the remaining atoms to rethermalize by elastic collisions which produces new highly energetic atoms to remove. A problem occurs when applying this technique to

fermions. Because the many body wavefunction for fermions is antisymmetric, s-wave collisions are forbidden between fermions; however, at low temperature, only s-wave collisions are dominant. This means that particles in a gas of identical fermions do not collide with each other at low temperature. This makes evaporative cooling impossible for identical fermions.

The solution to this problem is to cool a mixture of distinguishable particles. Then the anti-symmetry of the wave function is not required and the atoms can collide even at low temperature. This cooling scheme is called sympathetic cooling and was first proposed for two-component plasmas[11].

2.1.5 Evaporative Cooling

Until now evaporative cooling is still the only cooling method by which quantum degeneracy can be reached. The main point of evaporative cooling is to remove highly energetic atoms from the ensemble, so that the remaining atoms can rethermalized by elastic collisions and the overall temperature of the system is reduced.

The atoms that are to be cooled are prepared in a low magnetic field seeking Zeeman state and kept in a magnetic trap, as shown in Fig.2.2. The magnetic field has a non-zero minimum at the center and increases in all directions. Selective removal of the high energy atoms is achieved by driving transitions to high field seeking Zeeman states ($m_F g_F < 0$) using radio frequency or microwave frequency radiation. The radio frequency radiation (ν_{rf} in Fig.2.2) drives transitions between different Zeeman sublevels inside the same F state, microwave frequency on the other hand, drives transitions between Zeeman sublevels in different hyperfine states, such as ν_{hf} in Fig.2.2. For a given RF/MF frequency, atoms at a specific spatial position are removed. The frequency acts like a knife to cut away the high energy tail of the distribution and is usually called a RF/MF *frequency knife*.

After most of the high energy atoms are removed from the sample and the remaining atoms are rethermalized, the sample temperature will decrease more and more slowly. To force the sample to lower temperatures at a rate faster than the intrinsic heating mechanisms (due to collisions with background vapor and three-body recombination), the RF/MF frequency has to be lowered as the sample cools. This process is called *forced evaporative cooling*, and the frequency decreasing procedure is called an RF/MF *frequency ramp*.

2.2 Ultracold Collisions without External Magnetic or Optical Field

2.2.1 Elastic Collision

Elastic collisions conserve the total kinetic energy and momentum of the two colliding particles, but can change the momenta of each particle. This kind of collision is responsible for establishing thermal equilibrium in the sample. In the ultracold regime, elastic collisions between neutral atoms is dominated by s-wave scattering

which can be characterized by a single parameter, the s-wave scattering length a. Positive (negative) values of a correspond to repulsive (atrractive) interactions. In the s-wave limit, two-body collisions at a finite energy are described by the scattering amplitude:

$$f = -\frac{a}{1 + ika} \tag{2.2}$$

where $\hbar k$ is the relative momentum of the colliding particles. The total cross-section for elastic collisions is then given by:

$$\sigma = 4\pi|f|^2 = \frac{4\pi a^2}{1 + k^2a^2} \tag{2.3}$$

In the weak interaction limit, $ka << 1$, σ reduces to $4\pi a^2$. On the contrary, at the strong interaction limit (or for large relative momentum), σ reduces to $4\pi/k^2$.

Cross-thermalization experiments, in which one kind of atomic cloud is set to be hotter than (or in motion with respect to) the other one, can be used to determine the scattering length a between the colliding particles. Thermal equilibrium of the whole atomic sample is be reached by sympathetic cooling. The inter-atomic collision cross section can then be extracted from the thermalization speed.

2.2.2 Inelastic Collision

Inelastic collisions involve an energy transfer between the internal and external degrees of freedom. In two-body inelastic collisions, the internal Zeeman and/or hyperfine energy of the colliding particles is converted into kinetic energy. The released internal energy is usually higher than the trap depth and therefore the products of inelastic collisions are lost from the trap. In cases when the released internal energy is smaller than the trap depth, inelastic collisions lead to a heating of the sample.

Another example of inelastic collisions is the three-body recombination process. In a collision between three particles, two of them can form a bound molecular state, usually in a highly excited vibrational state, while the third one carries off the released binding energy. Since the energy involved is usually much larger than the trap depth, all three particles will be lost from the trap[12].

2.3 Ultracold Collisions with External Magnetic or Optical Field

2.3.1 Feshbach resonance

The scattering length of two colliding atoms can be tuned using magnetic field. One special case is the Feshbach resonance, which was first studied in nuclear physics by Herman Feshbach[13]. The main idea of a magnetic-field Feshbach resonance can be explained using Fig.2.3.

Consider two atoms in a collisional state with a kinetic energy E_{kin} in their mutual potential, which is called the open channel in the figure. For the same pair of atoms in

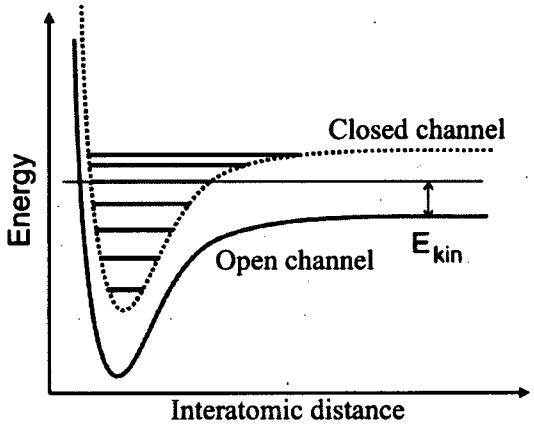


Figure 2.3: Principle of Feshbach resonance. A Feshbach resonance occurs, when a bound state of a closed channel is tuned into degeneracy with the kinetic energy E_{kin} of colliding particles using an external magnetic field.

different internal states, the potential energy surface is different and if the continuum lies above the energy of the collisional state, this is a so-called closed channel. If E_{kin} is equal to the energy of a bound molecular state of these two atoms, the free colliding atoms can be coupled into this bound state of the closed channel and a Feshbach resonance appears. If the energies of the open and closed channels are close but not exactly equal, $E_{kin} \neq E_{bound}$, coupling between the channels gives rise to a repulsive interaction when E_{kin} is greater than the bound state energy, and an attractive one if it is less. When the closed and open channels have different magnetic moments, these states (and the corresponding interaction) can be tuned against each other by applying an external magnetic field B, $\Delta E = \Delta\mu \times B$.

A typical Feshbach resonance is shown in Fig.2.4, which is calculated using the following scattering length equation from a simple model.

$$a(B) = a_{bg} \cdot (1 - \frac{\Delta B}{B - B_0}) \tag{2.4}$$

Where a_{bg} is the background scattering length far from the resonance, ΔB is the width of the resonance, B_0 is the center of the resonance. In Fig.2.4, $\Delta B = 25$G, $B_0 = 200$G, $a_{bg} = 10\ a_0$.

2.3.2 Photoassociation spectroscopy

Photoassociation (PA) spectroscopy was first proposed by Thorsheim[14]. Now it plays a significant role in understanding the collisional properties of ultracold atoms, and can be used to test the calculations of inter-atomic potentials, to make precise measurements of the molecular excited state lifetimes. Knowing the shape of the inter-atomic potentials provides the s-wave scattering length. Most important of all, photoassociation is another method to create ultracold molecules besides Feshbach resonance. The main idea of photoassociation is the following.

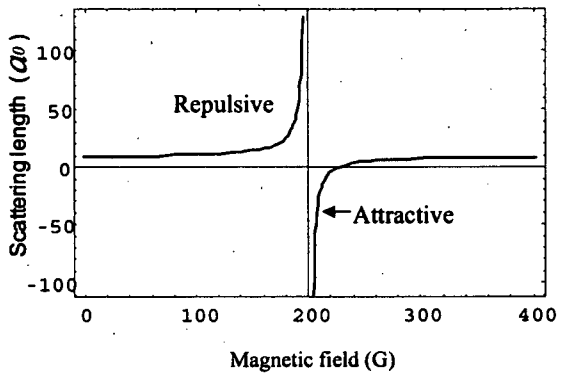


Figure 2.4: A sample Feshbach resonance calculated using eqn.2.4. When $\Delta B = 25$G, $B_0 = 200$G, $a_{bg} = 10\ a_0$

During the photoassociation process, two colliding atoms absorb a single photon from a photoassociation laser beam whose energy is on resonance with a bound but electronically excited molecular state, and then these two atoms are converted into an electronically excited molecule. These excited state molecules can decay into singlet or triplet ground state molecules depending on the symmetry of the excited molecular state. Raman photassociation can be used to convert these excited molecules into a specific ro-vibrational state molecule.

Chapter 3

Numerical Simulation of Sympathetic Cooling

3.1 Introduction

In order to provide quantitative theoretical account for the ultra-cold collision experiment, we developped a numerical simulation of the motion of the atoms in the magnetic trap. This approach consists in a molecular dynamics evolution of the atomic sample. It allows a precise description of the atomic dynamics in the magnetic trap including elastic and inelastic collisions, collisions with background vapor, and Majorana spin flip losses. By adjusting initial parameters in the program, we can simulate our experiment on the computer before actually performing it and use these results to deduce optimized evaporation parameters for future experiment. After the experiment is finished, we can compare the resutls with our simualtion parameters to deduce the scattering length between Rubidium and Lithium atoms.

The main idea of the simulation comes from a numerical model developed in J. Dalibard's group[15]. In which a concept of macro-atoms and the related duplication technique are used to simulate hundreds of millions of real atoms using several thousands of macro-atoms. The collisions between macro-atoms are taken into account using a boxing technique, in which the volume occupied by the atomic sample is divided into many small boxes, collisions may if the macro-atoms are found to be in the same small box. This greatly reduces the number of pairs considered to accurately simulate collisions in the ensemble. In the simulation, the radio/microwave frequency ramping is converted into a magnetic field boundary. During evaporative cooling procedure, the local magnetic field for each macro-atom will be calculated and compared with the magnetic field boundary to determine whether it still stays in the trap or not.

Originally the above model was specially designed for simulating the forced radio-frequency evaporation of ^{133}Cs atoms inside the magnetic trap within the harmonic approximation (for the trap), my work was to generalize this simulation so that it can simulate the dynamics of a multi-species trap including inter-atomic collisions and non-harmonic trapping potentials.

The primary code for this simulation is written in Fortran 77. To utilize the new advanced features of Fortran 90, I transformed the old Fortran 77 code to Fortran 90. The complete program consists of the main file "ev.f90" and the input file "ev.in" [16].

The simulation is running on our group server: qdg.physics.ubc.ca using the Intel Fortran Compiler 8.1 with free student license, which has dual 3.06GHz Xeon CPUs and 1GB of memory. Running the simulation once takes from several minutes to a

few hours depending on the size of the atomic sample and some other factors.

This chapter is divided into two parts. In the first part, section 3.2 to 3.11, the numerical model of this simulation is discussed; In the second part, section 3.12 to 3.14, this model is applied to the experiments performed in Claus Zimmerman's group on mixture of Lithium and Rubidium [3] to check the code's validity.

3.2 Molecular Dynamics

Molecular Dynamics (MD) is a computer simulation technique where the time evolution of a set of interacting particles is followed by integrating their equations of motion. It has been widely used in computer experiments, here I just give the basic introduction, please check reference[17] for further information.

In this simulation, we evolve the distribution of particles according to the laws of classical mechanics, most notably Newton's Law:

$$F_i = m_i a_i \tag{3.1}$$

for each atom i in a system of N atoms. Here, m_i is the i^{th} atom's mass, $a_i = d^2r_i/dt^2$ is its acceleration, and F_i is the force acting upon it, due to the interactions with other atoms or the external potential. Therefore, in contrast with the Monte Carlo method, a molecular dynamics simulation is an *initial value problem*: given an initial set of positions and velocities, the subsequent time evolution is in principle completely determined.

In brief, using a molecular dynamics simulation, one has access to the macroscopic properties of a system through the microscopic simulations. *Statistical mechanics* provides the rigorous mathematical expressions that relate macroscopic properties to the distribution and motion of the atoms or molecules of the N-body system.

3.3 Box-Muller Method and Random Numbers

Random numbers are used quite often in this simulation, such as to decide from their probability if an elastic or inelastic collision has occurred or if a background collisional loss has occurred. In these situations we need uniformly distributed random numbers. On the other hand, when we try to initialize the positions and velocities of the atoms, Gaussian distributed random numbers are required. Function RAN3(idum) provides the uniformly distributed random numbers in the range of $[0, 1]$, Fig. 3.1 shows the results of 10^6 times samplings of function RAN3(idum).

The Box-Muller method[18] is utilized to transform the uniformly distributed random numbers to a new set of random numbers with Gaussian (normal) distribution. The most basic form of this transformation looks like:

$$y = sqrt(-2 \times ln(x_1)) \times cos(2\pi x_2) \tag{3.2}$$

x_1 and x_2 are random numbers created by RAN3(idum), y is a random number in the Gaussian distribution with zero mean value and a standard deviation of one, as

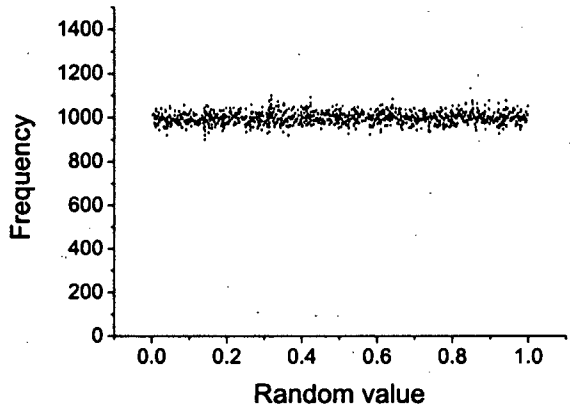


Figure 3.1: Uniformly distributed random numbers produced by sampling function RAN3f(idum) for 10^6 times. The values of these random numbers are in the range of [0,1].

shown in Fig. 3.2 . $xc = 0.00125 \pm 0.00099$ is the mean value, $w = 2.00025 \pm 0.00271$ is two times the standard deviation.

The basic form of Box-Muller method is not very fast, since it calls the math library all the time, the cosine function and the natural logarithm. However, this method is only used to initialize the positions and velocities of several thousand macro-atoms at the beginning of the simulation, so the basic form is fast enough for our application.

3.4 Macro Atoms and Duplication Process

Since the main point of evaporative cooling is to get rid of the highly energetic atoms to lower the overall temperature of the remainder atoms, we need to load lots of atoms at the beginning of the evaporative cooling, mostly in the range of $10^6 \sim 10^9$, as shown in Table 3.1.

Keeping track of all these atoms during the simulation is beyond the ability of today's desktop computers, however, since what we need is just some statistical physical parameters, like temperature, density, it is not necessary to keep track of every single atom. In this program, a distribution of Macro-atoms, whose number is maintained between 4000 and 10000, is evolved to be compared with the $10^6 \sim 10^9$ real atoms initially present. Each Macro-atom represents p real atoms, with $p=2^x$, but has the same mass and the same magnetic moment as a single atom so that it moves within the trap as a single atom. It's collision cross section is p times bigger than the collision cross section of a single atom.

Every time the number of Macro-atoms is lower than half of the initial value, either because of the evaporation or because of loss due to collisions with the residual

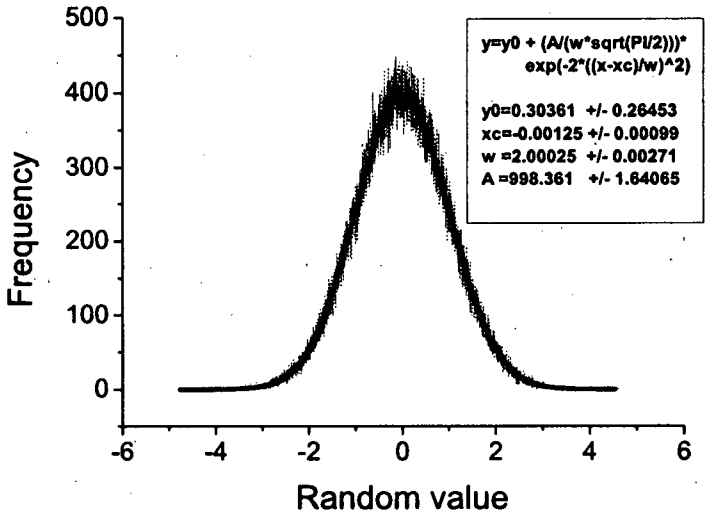


Figure 3.2: Gaussian distributed random numbers produced by sampling function RAN3(idum) for 10^6 times.

background gas, every Macro-atom will be replaced with two new Macro-atoms, each of which represents $p/2$ real atoms, this is also the reason that p is base 2 exponential. These two new Macro-atoms are created symmetrically about z axis, in other words, if the parent macro-atom is at (x, y, z) with velocity (v_x, v_y, v_z), then one of two new macro-atoms is placed at the same point with the same velocity, and the other one is placed in $(-x, -y, z)$ with the velocity $(-v_x, -v_y, v_z)$. This duplication, which exploits the symmetry of the trap, guarantees that these two new macro-atoms will not undergo a collision with each other immediately after the duplication process. The duplication process stops when p is equal to 1.

3.5 Boxing Technique

The collisions between the macro-atoms are taken into account using a boxing technique[19]. After each evolution time step δt, the position of each particle is discretized with a step δr, so that each particle is assigned to a cubic box of volume δr^3.

$$\delta r = 4 \times MAX\left(\frac{\sigma(x)}{nmaxx}, \frac{\sigma(y)}{nmaxy}, \frac{\sigma(z)}{nmaxz}\right) \tag{3.3}$$

In which $\sigma(i)$ $(i = x, y, z)$ is the standard deviation of position for the atom cloud in i direction, $nmaxi$ is the number of boxes in i direction. The overall number of boxes is $nmaxx \times nmaxy \times nmaxz = 3.24 \times 10^5$; Since $\sigma(i)$ represents the size of the atom cloud, δr is adjusted as the cloud cools down, so that the probability for having two

Group	Atom	N (10^6)
Rice	^{7}Li	200
		0.1
MIT	^{23}Na	1000
		0.7
JILA	^{87}Rb	4
		0.02

Table 3.1: Results obtained with evaporative cooling for the achievement of BEC, the first line represents the starting point in each case, and the second line represents the end point.

particles in the same box is much smaller than 1. When two macro-atoms are found in the same box, a collision may take place between them.

To determine the number of macro-atoms in the same box, each macro-atom is indexed by the discretized distance to the center of mass, ix, iy, iz are defined to be the indexes.

$$ix = NINT((x - \bar{X})/\delta r) \tag{3.4}$$

$$iy = NINT((y - \bar{Y})/\delta r) \tag{3.5}$$

$$iz = NINT((z - \bar{Z})/\delta r) \tag{3.6}$$

The box structure is realized by TABLE(ix, iy, iz), which stores the serial number "iat" of the atom at position (x, y, z). During each time step, for each atom, we will first find out the box it belongs to, then check if the box is empty or not, which means TABLE(ix, iy, iz) equals to zero or not. If the box is empty, that atom will be placed inside this box; Otherwise, if there is already an atom inside, then that atom may undergo a collision with the previously placed atom. After collision, both atoms will leave the box, and the table element TABLE(ix, iy, iz) will be set to 0.

3.6 The Magnetic Trap

Although Evaporative cooling is the only way to achieve the critical phase space density for BEC, there are different ways to implement it, such as in a magnetic trap or in an optical dipole trap[20], etc. The most common technique is evaporative cooling in the magnetic trap, which also has lots of different types. Here I will consider atoms in an Ioffe-Pritchard trap[21], since it is the first and most common trap used for evaporative cooling. For Ioffe-Pritchard trap, as shown in Fig. 3.3, the magnetic field takes this form:

$$\vec{B} = \begin{pmatrix} B'x - B''zx/2 \\ B_0 + B''y^2/2 - B''(x^2 + z^2)/4 \\ -B'z - B''zy/2 \end{pmatrix} \tag{3.7}$$

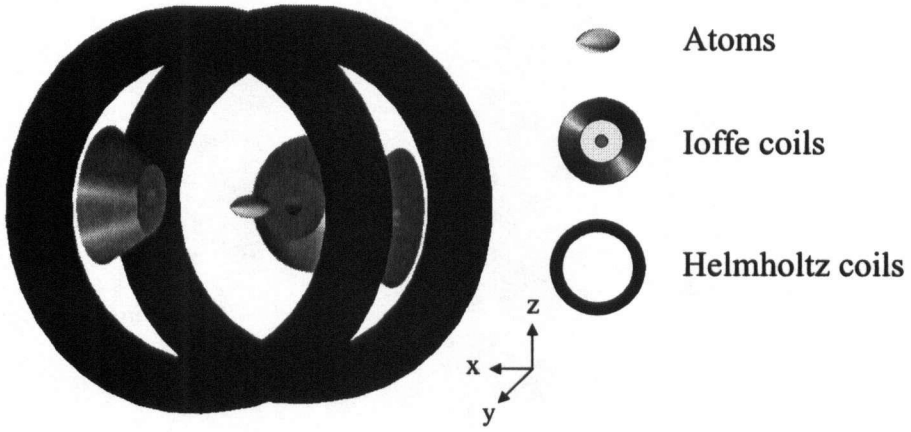


Figure 3.3: Ioffe-Pritchard Trap.

Then the magnitude of $\vec{B}$ is:

$$|\overrightarrow{B}| = \frac{1}{2}\sqrt{(2B'x - B''zx)^2 + (-2B'z - B''zy)^2 + (2B_0 + B''y^2 - B''(x^2+z^2)/2)^2} \tag{3.8}$$

If we do Taylor expansion to $\overrightarrow{B}$ around the coordinates center $(0,0,0)$,

$$|\overrightarrow{B}| = B_0 + (\frac{B'^2}{2B_0} - \frac{B''}{4})x^2 + \frac{1}{2}B''y^2 + (\frac{B'^2}{2B_0} - \frac{B''}{4})z^2 + \cdots \tag{3.9}$$

During the process of evaporative cooling, most trapping atoms just oscillate around the trap center, so we can just keep Eqn. 3.9 to the second order, now it's quite clear that the magnetic potential $\mu|\overrightarrow{B}|$ is approximately simple harmonic around the trap center.

$$|\mu\overrightarrow{B}| \simeq \mu B_0 + \frac{1}{2}\mu(\frac{B'^2}{B_0} - \frac{B''}{2})x^2 + \frac{1}{2}\mu B''y^2 + \frac{1}{2}\mu(\frac{B'^2}{B_0} - \frac{B''}{2})z^2 \tag{3.10}$$

By the definition of simple harmonic trap, $\mu|\overrightarrow{B}| = \frac{1}{2}m(\omega_x{}^2x^2 + \omega_y{}^2y^2 + \omega_z{}^2z^2)$, we can find the trapping frequencies in 3D:

$$\omega_x{}^2 = \omega_z{}^2 = \frac{\mu}{m}(\frac{B'^2}{B_0} - \frac{B''}{2}), \qquad \omega_y{}^2 = \frac{\mu B''}{m} \tag{3.11}$$

Where μ is the magnetic moment of the atom, B_0 is the bias field, B' is the field gradient, B'' is the field curvature, ω_i $(i = x, y, z)$ is the trapping frequency in i direction.

For the simplified Ioffe-Pritchard trap, the magnetic field looks like:

$$\overrightarrow{B} = \begin{pmatrix} B'x \\ B_0 + B''y^2/2 \\ -B'z \end{pmatrix} \tag{3.12}$$

By the same procedure as Eqn. 3.7- 3.11, we get the trapping frequencies for the simplified Ioffe-Pritchard trap:

$$\omega_x{}^2 = \omega_z{}^2 = \frac{\mu B'^2}{mB_0}, \qquad \omega_y{}^2 = \frac{\mu B''}{m} \tag{3.13}$$

3.7 Runge-Kutta Method and Equations of Motion

This program is just a classical simulation of the dynamics for the atoms inside the magnetic trap, quantum mechanics is not being used here, the atoms are described by classical positions and velocities at each time step, not by wavefunctions. Based on this assumption, we can solve the dynamics by Newton's Law, which means the equations of motion are just ordinary differential equations (ODEs), more specifically, they are just second order ODEs.

A problem involvling ODEs is not completely specified by the differential equations, more crucially it is determined by the boundary conditions. The nature of the boundary conditions determines which numerical methods will be feasible to solve the ODEs. Despite their diversities, all the boundary conditions fall into two broad categories:

- *initial value problems*: all the variables are given at some starting value x_s, and it is desired to find these variables at some final point x_f, or at some discrete list of points.
- *two-point boundary value problems*: on the other hand, boundary conditions are specified at more than one x. Typically, some of the conditions will be specified at x_s and the remainder at x_f.

Since our simulation is about the time evolution of the atoms, this is an *initial value problem.* There are many methods to solve the *initial value problems* for ODEs, like the Runge-Kutta method, the Bulirsch-Stoer method, and the Predictor-corrector method, etc. The most often used is the classical *fourth-order Runge-Kutta formula*[22], which has a certain sleek organization:

$$k_1 = \delta t \cdot f(t_n, y_n) \tag{3.14}$$

$$k_2 = \delta t \cdot f(t_n + \frac{\delta t}{2}, y_n + \frac{k_1}{2}) \tag{3.15}$$

$$k_3 = \delta t \cdot f(t_n + \frac{\delta t}{2}, y_n + \frac{k_2}{2}) \tag{3.16}$$

$$k_4 = \delta t \cdot f(t_n + \delta t, y_n + k_3) \tag{3.17}$$

$$y_{n+1} = y_n + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6} + O(H^5) \tag{3.18}$$

The fourth-order Runge-Kutta method requires four evaluations of the right-hand

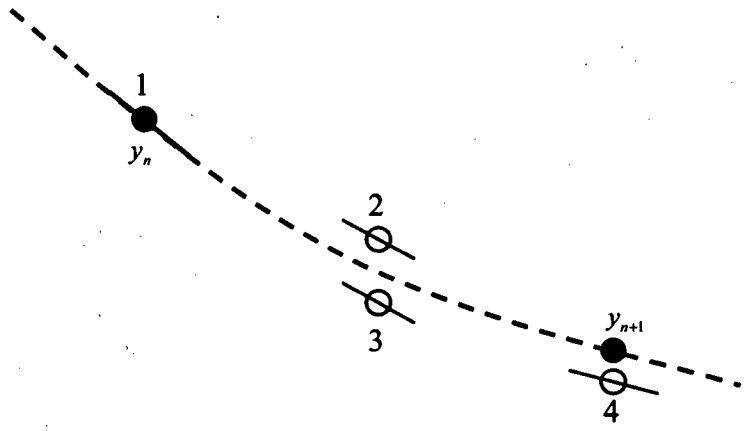


Figure 3.4: Fourth-order Runge-Kutta method. In each time step the derivative is evaluated four times: once at the initial points, twice at trial midpoints, and once at a trial endpoint. From these derivatives the final function value(shown as a filled dot) is calculated.

side per step δt (see Fig. 3.4). In our program, subroutine RK4$(t, y, \delta t)$ does the algorithm from Eqn. 3.14 to Eqn. 3.18 , DERIVS(t, y, yt) calculates the derivatives. To carry out one classical Runge-Kutta step, we input the independent variable $y(1:6)$, which stores 3 dimensional positions and velocities, and then get the new values which are stepped by a stepsize δt and stored in the same array $y(1:6)$.

In DERIVS(t, y, yt), $yt(1:6)$ stores the derivatives of the corresponding elements in $y(1:6)$, it is obvious that $yt(1:3)$ are just equal to $y(4:6)$, since $y(1:3)$ are positions and $y(4:6)$ are velocities. $yt(4:6)$ are the derivatives of $y(4:6)$, in other words, they are the accelerations, dv/dt. By Newton's second law, $a = F/m$, we can calculate out $yt(4:6)$, since $F = -\nabla E = -\nabla(\mu|\vec{B}|)$.

3.8 Initialization of the Simulation

Some of the parameters, such as initial temperature, initial number of atoms and vacuum lifetime, etc., are stored in the input file "ev.in". Besides these common parameters, there still several other parts that need to be initialized, such as the initialization of the positions and velocities of the macro-atoms with a Gaussian distribution, the initialization of the spin states and the corresponding Landé g factor, and the initialization of the elastic/inelastic collisional cross sections between different combination of atoms.

3.8.1 Initialization of positions and velocities

The initial positions and velocities in the simulation are realized by the Box-Muller method:

$$pos(iat, i) = dep_g + pos_0(i)\, cos(2\pi x_2)\sqrt{-2ln(x_1)} \tag{3.19}$$

$$vel(iat,i) = vel_0(i)\ sin(2\pi x_2)\sqrt{-2ln(x_1)} \tag{3.20}$$

in which x_1 and x_2 are random numbers in the range of [0,1] produced by RAN3(idum), $pos_0(i)$ and $vel_0(i)$ are the initial position amplitude and velocity amplitude in the i $(i = x, y, z)$ direction determined by the initial temperature of the atom cloud.

$$\frac{1}{2}m\,\omega_i^2\,pos_0(i)^2 = \frac{1}{2}k_BT_0, \qquad \frac{1}{2}m\,vel_0(i)^2 = \frac{1}{2}k_BT_0 \tag{3.21}$$

We know that in thermal equilibrium, kinetic energy of the atoms should satisfy the Maxwell-Boltzmann distribution,

$$f(E_k) \propto e^{-\frac{E_k}{k_BT}} \tag{3.22}$$

Box-Muller method gives us a Gaussian distribution with the following fitting equation (Fig. 3.2):

$$y = y_0 + \frac{A}{w\sqrt{\pi/2}}\,e^{-2\frac{(x-x_c)^2}{w^2}} \tag{3.23}$$

After the approximation $x_c \approx 0$ and $w \approx 2$, Eqn. 3.23 turns to be:

$$y = y_0 + \frac{A}{\sqrt{2\pi}}\,e^{-\frac{x^2}{2}} \tag{3.24}$$

here x represents some random number produced by Box-Muller method, in the other words,

$$x = sin(2\pi x_2)\sqrt{-2ln(x_1)} \tag{3.25}$$

If we multiply x with the velocity amplitude $vel_0(i)$, as in Eqn. 3.20, we can get the velocity, $vel(iat,i)$. Now we do this transformation to Eqn. 3.24, we get

$$y = y_0 + \frac{A}{\sqrt{2\pi}}\,e^{-\frac{(vel_0x)^2}{2vel_0{}^2}} = y_0 + \frac{A}{\sqrt{2\pi}}\,e^{-\frac{vel(iat,i)^2}{2vel_0{}^2}} \tag{3.26}$$

From Eqn. 3.21, we know that $vel_0{}^2 = k_BT_0/m$, then

$$y = y_0 + \frac{A}{\sqrt{2\pi}}\,e^{-\frac{m\,vel(iat,i)^2}{2k_BT_0}} = y_0 + \frac{A}{\sqrt{2\pi}}\,e^{-\frac{E_k}{k_BT_0}} \tag{3.27}$$

So the velocities produced by Box-Muller method do satisfy the Maxwell Boltzmann distribution.

If we do the same transformation to positions, which means multiplying $pos_0(i)$ to x, followig the same procedure as Eqn. 3.26 to Eqn. 3.27, we can get a similar result for the position distribution.

$$y = y_0 + \frac{A}{\sqrt{2\pi}}\,e^{-\frac{m\omega(i)^2pos(iat,i)^2}{2k_BT_0}} = y_0 + \frac{A}{\sqrt{2\pi}}\,e^{-\frac{E_p}{k_BT_0}} \tag{3.28}$$

Since the trapping frequencies ω_i are different, according to the trap geometry, the initial postion distribution looks like a cigar in the Ioffe-Prichard trap, as shown in Fig. 3.3 .

3.8.2 Initialization of spin states

Evaporative cooling always involves atoms in their ground electronic states, which for alkalis means the orbital angular momentum of the electron is $L = 0$, then the total angular momentum ($\vec{J} = \vec{L} + \vec{S}$) for the single valence electron ($S = 1/2$) of the alkali atom is

$$J = |0 \pm 1/2| = 1/2 \tag{3.29}$$

Since the total angular momentum of the atom is $\vec{F} = \vec{I} + \vec{J}$, in which $\vec{I}$ is the nuclear spin, the ground state hyperfine splittings always have two sublevels, $F = |I \pm 1/2|$. Due to the uniqueness of the the F states in each specific L energy level, we can define the atom's spin state by the combination of the F state and the related Zeeman sublevel ($m_F = -F....F$).

In the program a new data type is defined as following.

```
type atom
   real*8 F
   real*8 mF
end type atom
```

"type(atom) a(1:nat)" allots the real memory space for all the atoms (from "1" to "nat"), it includes both F state and the Zeeman sublevel m_F.

User-defined Data Type is a new feature in Fortran 90, it allows us to define new data types derived from any combination of the intrinsic data types and derived types. The derived-type object can be accessed as a whole, or its individual components can be accessed directly. For example, we can do operations on "a(1:nat)" directly, or access the individual components F and mF by "a(iat)%F", "a(iat)%mF" respectively.

3.8.3 Hyperfine Landé g-factor

Although g_F, the hyperfine landé g-factor, is available in most of the references and there is also simpler formulae to calculate this factor for alkali atoms in the ground states, such as:

$$g_F = \pm\frac{2}{2I+1} \tag{3.30}$$

in the simulation we still use the general formulae[23]:

$$g_F = g_J\frac{F(F+1) - I(I+1) + J(J+1)}{2F(F+1)} + g_I\frac{F(F+1) + I(I+1) - J(J+1)}{2F(F+1)} \tag{3.31}$$

in which F is the total angular momentum of the atom, I is the nuclear spin and J is the electron total spin. The reason is that Eqn. 3.31 is the general formulae to calculate the g_F of atoms in all different spin states, not only the alkali atoms in the ground states, this is also consistent with our purpose of generalizing the original program to satisfy wider applications.

Function gF(type,F) returns the value of the hyperfine landé g-factor for specific type of atom in specific spin state F, since for ground state alkali atoms ($J = 1/2$,

constant I, g_I and g_J), the only variable in Eqn. 3.31 is F. For situations dealing with atoms in the excited states, we just need to add one more parameter J in gF(type,F).

3.8.4 Initialization of collisional cross section

Since this program is purely a classical simulation of the dynamics of an atomic sample, the elastic/inelastic collisional cross sections are assumed to be classical quantities and are stored in two separate matrices at the beginning of the simulation.

"sigma_in(:,:)" and "sigma_e(:,:)" return the inelastic and elastic collisional cross sections between atoms with specific spin states respectively. "sigma_e(:,:)" comes from the scattering length matrix sl(:,:) by the relationship, sigma_e(:,:) = $8\pi \times$sl(:,:)^2, this expression is just the simplified version at low relative velocity limit. The kinetic energy dependence of the elastic collisional cross section is accounted for in a different section of the code by the following lines of code:

```
sigmaloc1=sigma/(1.d0+(m1*vcoll*a0/(2.*hbar))**2)
sigmaloc2=sigma/(1.d0+(m2*vcoll*a0/(2.*hbar))**2)
sigmaloc=sqrt(sigmaloc1*sigmaloc2)
```

In which "vcoll" is the relative velocity of the colliding particles, $m \cdot vcoll/(2\hbar)=k$ is the relative wave vector. The above code represents the following equation for $\sigma(k)$.

$$\sigma(k) = \frac{4\pi a^2}{1 + k^2 a^2} \tag{3.32}$$

Table 3.2 shows the structure for "sigma_in(:,:)" and "sigma_e(:,:)". Suppose that atomic species A has M spin states and atomic species B has N spin states, then the matrix size is (M+N)×(M+N). These spin states are ranked from the highest to the lowest. The arguments for "sigma_in(:,:)" and "sigma_e(:,:)" are not spin states, but they are associated with spin states. To relate these arguments to spin states, two extra functions, G_elastic(i1,i2) and G_inelastic(i1,i2), are used to calculate the array indices in the collisional cross section matrices for atoms with specific spin states and return the corresponding collisional cross sections.

Table 3.2: Matrix structure for "sigma_in(:,:)" or "sigma_e(:,:)". S_a1 and S_aM are the highest and lowest spin states for atomic species A respectively, S_b1 and S_bN are the highest and lowest spin states for atomic species B respectively.

		Atomic species A			Atomic species B		
		S_a1	...	S_aM	S_b1	...	S_bN
Atomic species A	S_a1	σ_{aa}^{11}	$\sigma_{aa}^{1...}$	σ_{aa}^{1M}	x	x	x
	...	x	x	x	x	x	x
	S_aM	x	x	x	x	x	x
Atomic species B	S_b1	x	x	x	x	x	x
	...	x	x	x	x	x	x
	S_bN	x	x	x	x	x	x

3.9 Collisions

Elastic and inelastic collisions are discussed in this section. During evaporative cooling, elastic collisions are critical for the re-thermalization of the atomic sample after the loss of the high energy tail of the distribution. Inelastic collision, on the other hand, release internal energy by changing the spin states of the colliding atoms and convert it into kinetic energy. Thus inelastic collisions increase the overall temperature of the atomic sample.

The probability for an elastic collision is $n\sigma v dt$, in which σ is the elastic collisional cross section and v is the relative velocity between the colliding particles. For inelastic collisions the collision rate is $n\mathrm{G}_{\mathrm{inelastic}}dt$, where $\mathrm{G}_{\mathrm{inelastic}}$ is the inelastic collisional cross volume and is assumed to be velocity independent. In the program, collision rates are calculated by the following lines of code, in which "macroloc/dx^3" equals to the atomic density n.

```
probacoll=macroloc*sigmaloc*vcoll*dt/(dx**3) !elastic collision rate
proba_coll_in=macroloc*G_in*dt/(dx**3) !inelastic collision rate
```

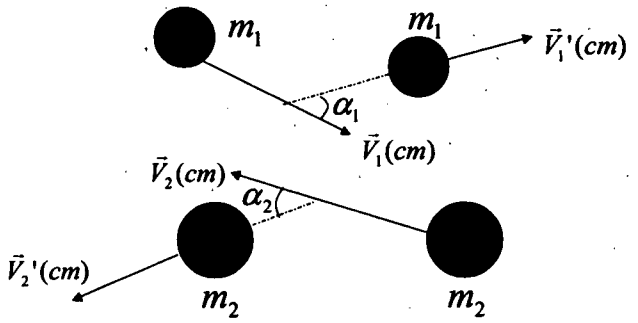


Figure 3.5: Collision diagram in the center of mass frame. $\vec{V_1}(cm)$ and $\vec{V_2}(cm)$ are the initial velocities of atom 1 and atom 2 respectively in the center of mass frame, $\vec{V_1'}(cm)$ and $\vec{V_2'}(cm)$ are the final velocities correspondingly, α_1 and α_2 are the scattering angles of atom 1 and atom 2 respectively.

3.9.1 Elastic collision

During elastic collisions, both the total momentum and the kinetic energy of the atoms undergoing the collision is conserved, as shown below:

$$m_1\vec{V_1} + m_2\vec{V_2} = m_1\vec{V_1}' + m_2\vec{V_2}' \tag{3.33}$$

$$\frac{1}{2}m_1V_1^2 + \frac{1}{2}m_1V_2^2 = \frac{1}{2}m_1V_1'^2 + \frac{1}{2}m_1V_2'^2 \tag{3.34}$$

$$\tag{3.35}$$

To figure out $\vec{V_1}'$ and $\vec{V_2}'$, we turn to the center of mass frame, as shown in Fig. 3.5. Suppose $\vec{V}_{cm}$ is the velocity of the center of mass.

$$\vec{V}_{cm} = \frac{m_1\vec{V_1} + m_2\vec{V_2}}{m_1 + m_2} \tag{3.36}$$

The velocities of atom 1 and atom 2 in the center of mass frame will be:

$$\vec{V_1}(cm) = \vec{V_1} - \vec{V}_{cm} = \frac{m_2(\vec{V_1} - \vec{V_2})}{m_1 + m_2} = \frac{m_2\vec{V}_{coll}}{m_1 + m_2} \tag{3.37}$$

$$\vec{V_2}(cm) = \vec{V_2} - \vec{V}_{cm} = -\frac{m_1(\vec{V_1} - \vec{V_2})}{m_1 + m_2} = -\frac{m_1\vec{V}_{coll}}{m_1 + m_2} \tag{3.38}$$

in which $\vec{V}_{coll} = \vec{V_1} - \vec{V_2}$ is the relative velocity between atom 1 and atom 2.

Now if we treat these atoms as hard spheres and neglect the interaction potential between them, we can calculate out the scattering angles for them, correspondingly the final velocities $\vec{V_1}'$ and $\vec{V_2}'$ can also be figured out by transferring $\vec{V_1}'(cm)$ and $\vec{V_2}'(cm)$ back to the laboratory frame. However, hard sphere treatment is only an approximation and such a calculation is computationally intensive since it requires a calculation of the exact point of contact between the two spheres. Therefore, in the program we adopt a much more simple method to find the final velocities after collision.

This simplified method calculates the new velocities by simply rotating the relative velocity vector $\vec{V}_{coll}$ by a random angle (θ, ϕ) in the three dimensional space. This process respects conservation of total momentum and kinetic energy and produces a final velocity distribution not unlike that from a hard sphere collision. The new relative velocity vector is then

$$\vec{V}_{coll} = (V_{coll}\sin\theta\cos\phi, V_{coll}\sin\theta\sin\phi, V_{coll}\cos\theta), \tag{3.39}$$

where θ and ϕ are random numbers created by function ran3(idum):

```
ctheta= 2.d0*ran3(idum)-1.d0
stheta=sqrt(1.d0-ctheta**2)
phi=2.*pi*ran3(idum)
```

Here "ctheta" represents $\cos\theta$, and "stheta" represents $\sin\theta$.

The expressions for the final velocities after collision are shown in the following equations:

$$V'_{1x} = V_{\mathrm{cm},x} + \frac{m_2}{m_1 + m_2}V_{coll}\ \sin\theta\ \cos\phi \tag{3.40}$$

$$V'_{1y} = V_{\mathrm{cm},y} + \frac{m_2}{m_1 + m_2}V_{coll}\ \sin\theta\ \sin\phi \tag{3.41}$$

$$V'_{1z} = V_{\mathrm{cm},z} + \frac{m_2}{m_1 + m_2}V_{coll}\ \cos\theta \tag{3.42}$$

$$V'_{2x} = V_{\mathrm{cm},x} - \frac{m_1}{m_1 + m_2}V_{coll}\ \sin\theta\ \cos\phi \tag{3.43}$$

$$V'_{2y} = V_{\mathrm{cm},y} - \frac{m_1}{m_1 + m_2}V_{coll}\ \sin\theta\ \sin\phi \tag{3.44}$$

$$V'_{2z} = V_{\mathrm{cm},z} - \frac{m_1}{m_1 + m_2}V_{coll}\ \cos\theta \tag{3.45}$$

With final velocities obtained from the above equations, both the total momentum and the kinetic energies are conserved.

3.9.2 Inelastic collision

In an inelastic collision, atoms change their spin state, and the change in internal energy is compensated by a change in the relative kinetic energy (while conserving the total momentum). Since the projection of the total spin is conserved in such a collision, m_F=$m_{F_1} + m_{F_2}$, different combinations of m_{F_1} and m_{F_2} can occur, and we assume (as an approximation) that they occur with the same probability. Clearly, some of the combinations may induce a release of the hyperfine energy and Zeeman energy and thus lead to the heating of the system.

In the program, an inelastic collision is realized by considering all (and then choosing one at random) of these possible combinations for a specific collision, which means this part has to be dealt with specially for each experiment and can't be generalized. The released internal energy, Zeeman energy or hyperfine energy, is converted into kinetic energy. This increased kinetic energy is first transformed into extra relative velocity, ΔV_{coll}, and then the total resultant relative velocity is transformed for the pair by the same method as in Eqn. 3.40$\sim$ 3.45.

3.10 Loss Mechanism

Different loss mechanisms are taken into account in the simulation, such as the evaporative cooling loss, the background loss and Majorana spin-flip losses.

3.10.1 Evaporative cooling

Radio frequency forced or microwave frequency forced evaporative cooling is usually used to get rid off the highly energetic atoms and thus decrease the overall temperature of the atomic sample. In the program, evaporative cooling is not done by simulating the RF/MF frequencies. Actually the RF/MF frequency knife is converted into magnetic field amplitude "Bevap" at which atoms will undergo a spin flip.

Fig. 3.6 shows one example of linear ramping, which is usually adopted by most evaporative cooling experiment. B_i and B_f are initial and final magnetic field amplitudes, which correspond to the initial and final RF/MF frequencies respectively. $B_{evap}(t)$ is the magnetic field amplitude which is related to the cutting energy E_{cut} at time t. The analytical expression for $B_{evap}(t)$ is given in Fig. 3.6 already. In the simulation, the local magnetic field of each macro-atom, $B_{loc}(iat)$, is calculated at the end of every time step and compared with the cutting magnetic field amplitude $B_{evap}(t)$. If $|B_{loc}(iat) - B_{evap}(t)| < \Delta B$, where $\Delta B = \hbar\omega_{rabi}/m_F g_F \mu_B$, the macro-atom with index "iat" will undergo a spin flip to the untrapped state with a probability $\omega_{rabi}^2 dt^2$ and finally be lost.

The straight line in Fig.3.6 is just the simplest version of linear ramping. In actual experiments, linear ramping is usually broken into several steps with different ramping slopes. At the the beginning step, quick ramping is utilized to get rid of

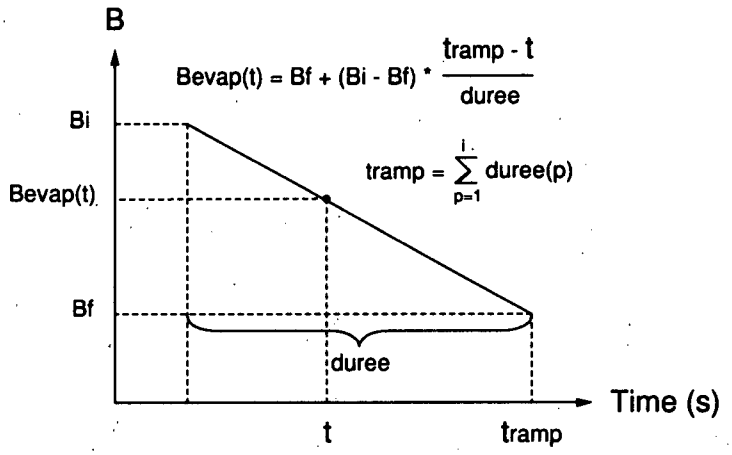


Figure 3.6: Linear ramping of the cutting magnetic field amplitude $B_{evap}(t)$.

the highly energetic atoms, and then the ramping speed is slowed down so that the atomic sample has enough time to re-thermalize. In the program, the duration of each step is represented by "duree(isegment)". t_{ramp} is the time interval from the beginning evolution time t_0 to the current ramping step, in other words, t_{ramp} is the sum from duree(1) to duree(i), as shown by the equation in Fig. 3.6.

3.10.2 Background loss

As we know, the magnetic lifetime is limited by the background pressure inside vacuum chamber. Collisions between the trapped atoms and room-temperature background vapor determines how long the ensemble of ultra-cold atoms can exist. In the simulation, background loss is realized by the following code.

```
if (indic(iat).and.(ran3(idum)*tau.lt.dt)) indic(iat)=.false.
```

where "tau" is the magnetic trap lifetime. The probability for an atom (or group of atoms) to be lost in some time interval dt is simply dt/tau (assuming this ratio is a number much smaller than 1). Using this probability, a macro-atom is lost from the trap if a random number between [0,1] is less than this ratio.

3.10.3 Majorana loss

Majorana loss is caused by a spin-flip transition of the trapped atoms which fly through or near the magnetic field minimum where the changing rate of the B-field vector is greater than Larmor precession frequency. Because of the spin transition, low field seeking atoms may switch to high field seeking states and consequently will be

accelerated out of the magnetic trap. In our program, Majorana loss is implemented as changing spin m_F to some random value between $-F$ and $+F$ when the particles venture closer than a specific distance r_l to the trap minimum. The distance r_l is given by [24]

$$r_l = \sqrt{\frac{\hbar V}{\Delta\mu \mathrm{B}'}} \tag{3.46}$$

Where $\Delta\mu$ is the magnetic moment difference between adjacent Zeeman levels, B′ is the magnetic field gradient for a quadruple trap, $V = \sqrt{k_B T/m}$.

The rate at which the trapped particles change their magnetic sublevels is given by

$$\Gamma = \frac{\hbar\mu}{m\Delta\mu R^2} \tag{3.47}$$

Where $R = k_B T/\mu \mathrm{B}'$.

3.11 Measurement

Just like actual experiments, the macroscopical properties of the atomic sample are measured in the simulation at regular intervals. In the input file "ev.in", there's a parameter named as "nmes" which determines how many times of measurements are going to be taken during the simulation. The measurement interval is decided by T/nmes, T is the total time consumed by the evaporation process.

During each measurement, many parameters, such as temperature, number of atoms, density and phase space density etc., are measured for each atomic species and the overall atomic sample. The results for these measurements are displayed on the screen and also stored into the output files. One example of the output file is shown below.

t	xrms	yrms	zrms	temp	N_atomes	taux_c	densit	phase	Pcol	Pocc	Bevap
0.00	118.	496.	120.	.126E+03	299991040	0.0	270.4	0.13E-04	0.00	0.00	0.00
0.04	275.	568.	264.	.893E+02	299761664	37.2	46.2	0.36E-05	0.01	0.69	98.37
0.08	289.	567.	269.	.907E+02	298942464	30.8	43.2	0.33E-05	0.01	0.75	97.97
0.12	375.	596.	339.	.918E+02	297697280	23.6	24.9	0.19E-05	0.01	0.95	97.58
0.17	390.	596.	334.	.915E+02	296222720	21.1	24.2	0.18E-05	0.01	1.03	97.18
...	...	...	...	.....	.....	....	....	.....	....	....	.....

3.12 Comparison

To validate the multi-species code, we did the following comparisons and tests.

The first is the comparison of results between original single species code and two species code. In this case, both the single species code and the multi-species code simulate the same dynamics of a rubidium sample, and in the multi-species code, the number of atomic species is set to be one.

The other test is a check of the results with different macro atom distributions using the multi-species code. Parameters are from the same experiment as the first comparison. In this simulation, a certain number of rubidium atoms are distributed

Table 3.3: Parameters for the ^{87}Rb and ^{6}Li mixture experiment in Zimmermann's group [3].

	^{87}Rb	^{6}Li
Initial spin states	$\|2,2>$	$\|3/2,3/2>$
Initial temperature	200-300μK	200-300μK
Initial number of atoms	3×10^{8}	2×10^{7}
Final number of atoms	1.2×10^{6}	2×10^{5}
Final temperature	620 nK	2.4μK
Trapping frequency	$\omega_x=\omega_y=2\pi\times206Hz$, $\omega_z=2\pi\times50.1Hz$	
B_0, B', B''	$B_0=3.5G$, $B'=3.02T/m$, $B''=1542T/m^2$	
Ramping frequency	from 200Mhz-100Mhz-100kHz above Rb trap bottom	

by different number of initial macros and a different number of atoms inside each macro so that the total number of atoms represented is the same.

3.12.1 Comparison of results between single species code and two species code

Fig. 3.7 shows the results for the comparison between single species code and two species code, which is composed of four graphs describing the time evolution of temperature, collision rate, density and phase space density separately. In this simulation, the size of the rubidium macro-atom is set to be 2^{15}=32768, the other parameters are from the rubidium part of table 3.3.

Most parts of these curves in Fig. 3.7 overlap perfectly, except that the beginning part and the small dips in collision rate graph are a little bit diverged. These dips correspond to the position of duplication processes where the atomic sample is duplicated and separated from the original one.

The perfect overlapping of most parts of these curves, especially the latter parts after the first two seconds, ensures that our multi-species code is good enough to give approximately the same results as the single species code.

3.12.2 Comparision of results between different macro atom distributions

As we discussed in section 3.4, the atomic sample is reduced into many macro-atoms, each of which represents $p=2^x$ real atoms, total number of the atoms equals the number of macro-atoms times p. The speed of the simulation is greatly influenced by the size of the sample atoms and the number of macro atoms, in order to shorten the time consumed by the simulation loop so that we can debug the code faster, fewer macro atoms are preferable when the size of the sample atoms is constant. However, we were not sure whether this kind of treatment will affect our results or not. To clarify this puzzle, we did the following test.

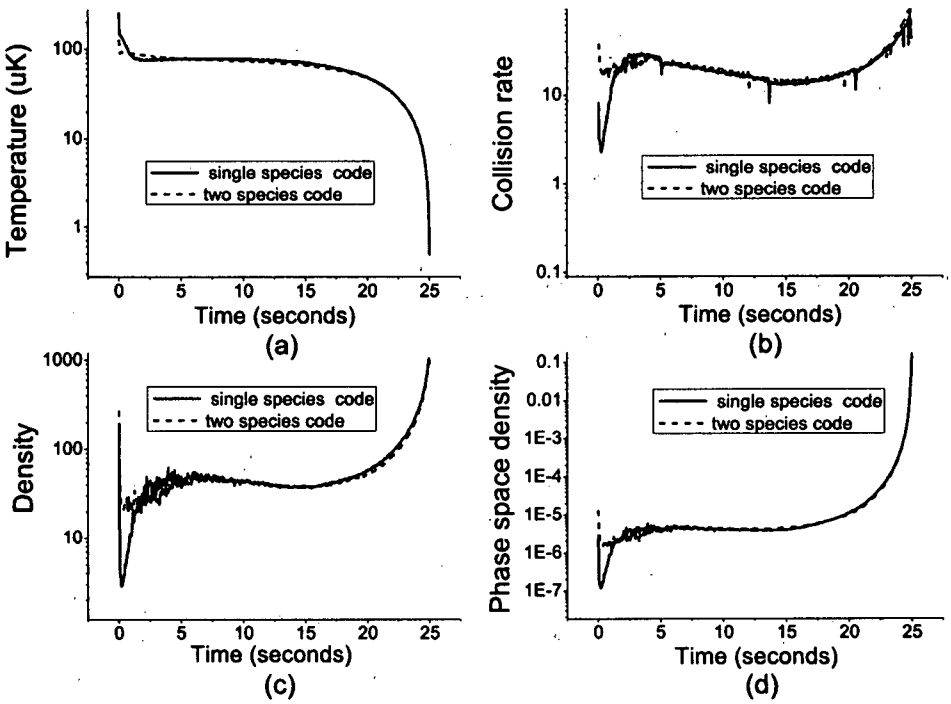


Figure 3.7: Comparison of the results between single species code and two species code. (a): Evolution of temperature. (b): Evolution of collision rate. (c): Evolution of phase space density. (d): Evolution of density.

By keeping the total number of atoms constant, we distribute the sample of atoms using different combinations between the number of macro atoms and the number of atoms inside each macro, such as 36621*8192 (2^{13}) and 9155*32768 (2^{15}), all the other parameters are the same and come from the rubidium part of Zimmermann's sympathetic cooling experiment [3]. The results for this comparison are shown in Fig. 3.8.

Clearly all the curves in Fig. 3.8 overlap perfectly for different macro atom distributions. The comparison of collision rate is not shown, since collision rate is related to the number of macro-atoms, which is not the same for different macro atom distributions and thus not worthy of comparison. Based on the perfect overlapping of the curves in Fig. 3.8, we can say that bigger size of the macro-atoms and less number of macro-atoms will not affect the simulation result too much, we can use less number of macro atoms to speed up the simulation for debugging purposes.

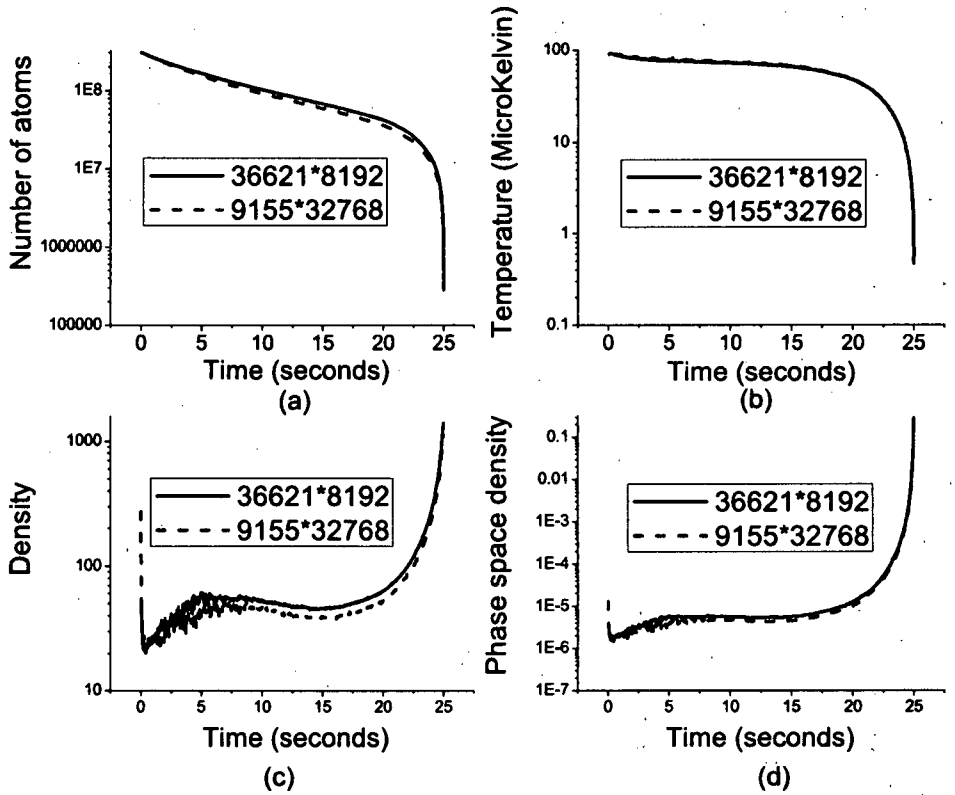


Figure 3.8: Evolution of two different macro atom distributions (36621×2^{13} and 9155×2^{15}) with equal number of total atoms. (a): Evolution of the number of atoms. (b): Evolution of temperature. (c): Evolution of density. (d): Evolution of phase space density.

3.13 Finding parameters for Zimmermann's experiment

Most of the parameters required for simulating Zimmermann's experiment are shown in table 3.3. In this section, we discuss the process for obtaining some important parameters.

3.13.1 B_0, B' and B''

In Zimmermann's experiment, the magnetic trap creates a Ioffe-Pritchard type potential, which can be simplified by a three-dimensional harmonic trap around the trap center. The trapping frequencies for this trap are given as, $\omega_x \approx \omega_y = 2\pi \times 206Hz$ and

Table 3.4: ^{87}Rb Physical Properties[23].

Atomic Number	Z	37
Total Nucleons	Z + N	87
Relative Natural Abundance	$\eta(^{87}\mathrm{Rb})$	27.83(2)%
Atomic Mass	m	86.909 180 520(15) u 1.443 160 60 × 10^{-25} kg
Nuclear Spin	I	3/2
Nuclear g-factor	g_I	-0.000 995 141 4(10)
Fine structure Landé g-factor	$g_J(5\ ^2S_{1/2})$	2.002 331 13(20)
Electron spin g-factor	g_S	2.002 319 304 373 7(80)
Electron orbital g-factor	g_L	0.999 993 69
Ground state hyperfine splitting	ω_{hf}	2π × 6.834 682 611 GHz

Table 3.5: ^{6}Li Physical Properties [27].

Atomic Number	Z	3
Total Nucleons	Z + N	6
Relative Natural Abundance	$\eta(^{40}\mathrm{K})$	7.6%
Atomic Mass	m	6.015 121 4 u 9.988 341 4 × 10^{-27} kg
Nuclear Spin	I	1
Nuclear g-factor	g_I	-0.000 447 654 0
Fine structure Landé g-factor	$g_J(2\ ^2S_{1/2})$	2.002 301 0

$\omega_z = 2\pi \times 50.1Hz$, also the bias field $B_0 = 3.5G$. For consistency with our program, let's switch ω_y and ω_z.

$$B' = \omega_x\sqrt{\frac{mB_0}{\mu}}, B'' = m\frac{\omega_y^2}{\mu} \tag{3.48}$$

Eqn. 3.48 describes the relationship between trapping frequencies and B', B'' of the Ioffe-Pritchard type magnetic trap, as discussed in section 3.6. B', B'' are the gradient and curvature of the IP trap. Here in Zimmermann's experiment, m is the mass for ^{87}Rb, μ is the magnetic moment of Rb atoms sitting at $|2,2>$ state. After calculation, we get:

$$B' = 302T/m, B'' = 1542T/m^2 \tag{3.49}$$

3.13.2 B_i and B_f

Microwave frequency is used to resonantly couple the Rb trapping Zeeman state $|2,2>$ and the untrapped $|1,1>$ state in Zimmermann's experiment, as shown in Fig. 3.9. At the beginning of the forced evaporative cooling, the microwave frequency is ramped down quickly from 200MHz+f_0 to 100Mhz+f_0 in 5 seconds, the next ramping step takes 20 seconds from 100Mhz+f_0 to 100khz+f_0. Here f_0 is the equivalent

	a_S	a_T	
^{6}Li-^{6}Li	45.5 ± 2.5	-2160 ± 250	[28]
^{7}Li-^{7}Li	33 ± 2	-27.6 ± 0.5	[28]
^{6}Li-^{7}Li	-20 ± 10	40.9 ± 0.2	[28]
^{87}Rb-^{87}Rb	93 ± 5	102 ± 6	[29]
^{23}Na-^{85}Rb	167^{+50}_{-30}	59^{+12}_{-9}	[30]
^{6}Li-^{87}Rb	N.A.	20^{+9}_{-6}	[3]
^{23}Na-^{87}Rb	55^{+3}_{-3}	51^{+9}_{-6}	[30]
^{40}K-^{40}K	N.A.	78 ± 20	[31]
^{40}K-^{85}Rb	N.A.	-38^{+37}_{-17}	[31]
^{40}K-^{87}Rb	N.A.	-261^{+170}_{-159}	[31]

Table 3.6: Singlet (a_S) and triplet (a_T) scattering lengths in units of a_0 (Bohr radius) for isotopically pure and mixed gases.

frequency for ^{87}Rb trap bottom, $f_0 = 2|g_F|\mu_B B_0/h \simeq 4.9$Mhz.

As mentioned in section 3.10.1, we are not simulating the RF/MF frequencies in the program, actually we convert the initial and final frequencies to the magnetic field boundaries, B_i and B_f. Eqn. 3.50 shows the method for this conversion.

$$h(f + f_0) = |g_F|\mu_B B \times \text{number of Zeeman sublevels} \tag{3.50}$$

Here in this experiment, the number of Zeeman sublevels should be two, since the trap state is $|2, 2>$ and m_F=2. $|g_F|$ equals to 1/2 for ground state ^{87}Rb atoms. After calculation, the initial and final magnetic field boundaries are found to be:

- B_i=1.463263215$\times 10^{-2}$T (200Mhz)
- B_f=3.571413161$\times 10^{-4}$T (100Khz)
- B_{middle}=7.491316073$\times 10^{-3}$T (100Mhz), the middle point.

3.13.3 Collision process

Another thing need to be discussed is the collision process between different spin states. For ^{87}Rb there are 3 low-field seeking Zeeman states: $|2, 2>$, $|2, 1>$ and $|1, -1>$, as shown in Fig. 3.9. In Zimmermann's experiment, the $|1, -1>$ atoms are removed from the trap once for all at the beginning of the evaporation ramp. The $|2, 1>$ state atoms are eliminated by applying a magnetic field offset chosen high enough to energetically separate the $|2, 2>$ from the $|2, 1>$ cloud, and then tuning the microwave frequency between the potential minimum seen by $|2, 1>$ atoms and the untrapped $|1, 0>$ state. Pulse durations of 5 ms is long enough to empty the undesired $|2, 1>$ states.

Based on the above discussion, only $|2, 2>$ state is considered in evaporative cooling process in the simulation.

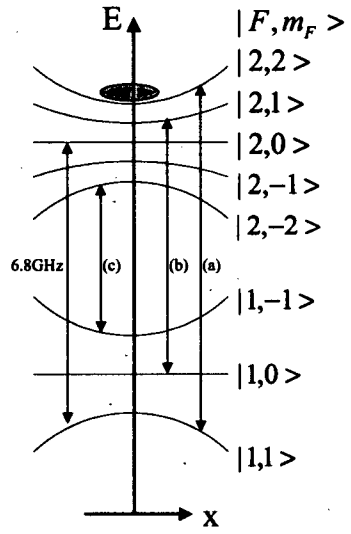


Figure 3.9: Scheme of the microwave transitions used (a) for evaporating the Rb cloud in the $|2, 2>$ state, (b) for removing atoms from the $|2, 1>$ state and (c) for removing atoms from the $|1, -1>$ state.

3.14 Next Step of Simulation

Two-species mixture experiment of quantum degenerate Bose and Fermi gases has been investigated by many groups, which covers various kinds of combinations, such as ^{40}K + ^{87}Rb [37], ^{6}Li + ^{7}Li [25], ^{6}Li + ^{23}Na [26] and ^{6}Li + ^{87}Rb [3], etc. Since our future experiment is about the ultra-cold collision between Lithium and Rubidium, the ^{6}Li + ^{87}Rb experiment performed in C. Zimmermann's group becomes to be the best choice for calibrating this simulation.

The next step for our simulation is to run the code for Zimmermann's sympathetic cooling experiment, from which to get useful information for our Li+Rb ultra-cold collisions experiment.

3.15 Conclusion

In this chapter, firstly, we discussed a numerical model for simulating evaporative cooling. This model was originally developed from J. Dalibard's group, and was specially designed for simulating the forced radio-frequency evaporation of ^{133}Cs atoms inside a magnetic trap within harmonic approximation. In my work, the single species code was developed so that it can simulate the dynamics of a multi-species trap including inter-atomic collisions and non-harmonic trapping potentials.

In the numerical model, the concept of macro-atoms and the related duplication technique were introduced. We also discussed the boxing technique, in which the whole atomic sample is divided into many small boxes, atoms will undergo collisions if they are found to be in the same box as the others. Different kinds of loss mechanisms

are included, such as the forced radio-frequency (or microwave frequency) evaporative cooling, background collisions, and Majorana spin flip losses.

To validate the consistence between our multi-species code and the original single-species code, a comparison was tried using the parameters from C. Zimmermann's sympathetic cooling of fermionic Lithium using Rubidium atoms. The result shows a very good consistence between the multi-species and single species codes. Based on the same parameters from Zimmermann's experiment, another comparison was performed to check the influence of different macro-atom distribution on the simulation results when the total number of real atoms is constant, since the number of macro-atoms determines how fast this program can run. Two kinds of distributions are compared, 36621×2^{13} and 9155×2^{11}, in which 2^x is the number of atoms inside each macro, the number ahead of it is the number of macro-atoms. The comparison shows that results from different macro-atom distributions agree and give us confidence that the finite sample size does not result in spurious results.

Chapter 4

Experimental Setup

A test chamber has been built in our lab in order to study the ultra-cold collisional properties between lithium and rubidium atoms, as shown in Fig. 4.1. The whole experimental setup consists of several subsystems:

- The atomic sources
- Vacuum system
- Laser system
- Computer control system

In the laser system, the rubidium laser was built by our research associate, Dr. Bruce Klappauf. Two summer students, Peter Eugster and Aviv Keshet, built a Broad Area Laser (BAL) amplifier system, producing laser light for a Lithium Magneto-Optic Trap. The computer control system was mainly constructed by a co-op student, Raymond Gao.

My contribution focuses on the vacuum system and part of the atomic sources. For the vacuum system, I made a 3D design of the whole vacuum test chamber using IronCad 8.0, as shown in Fig. 4.1. This design allows us to assemble all the parts required for the ultra-cold collision experiment virtually on the computer, such as the magnetic coils, dispensers, the glass chamber and vacuum pumps etc., and thus gives us useful guidance for actual setup. Also a temperature monitoring box was made by me to monitor various baking temperature of the vacuum system during bakeout. This system is described in Appendix A. For the atomic source system, the mounting scheme inside vacuum chamber for Rubidium dispensers is simulated in the above 3D design. Spot welding of the dispensers onto copper conductors was investigated and an efficient way is found by us to do the work and described in Appendix B.

In the following sections, I will discuss the atomic sources and vacuum system. The laser system and the computer control system are not included here, details for them can be found in the summer reports written by our undergraduate students.

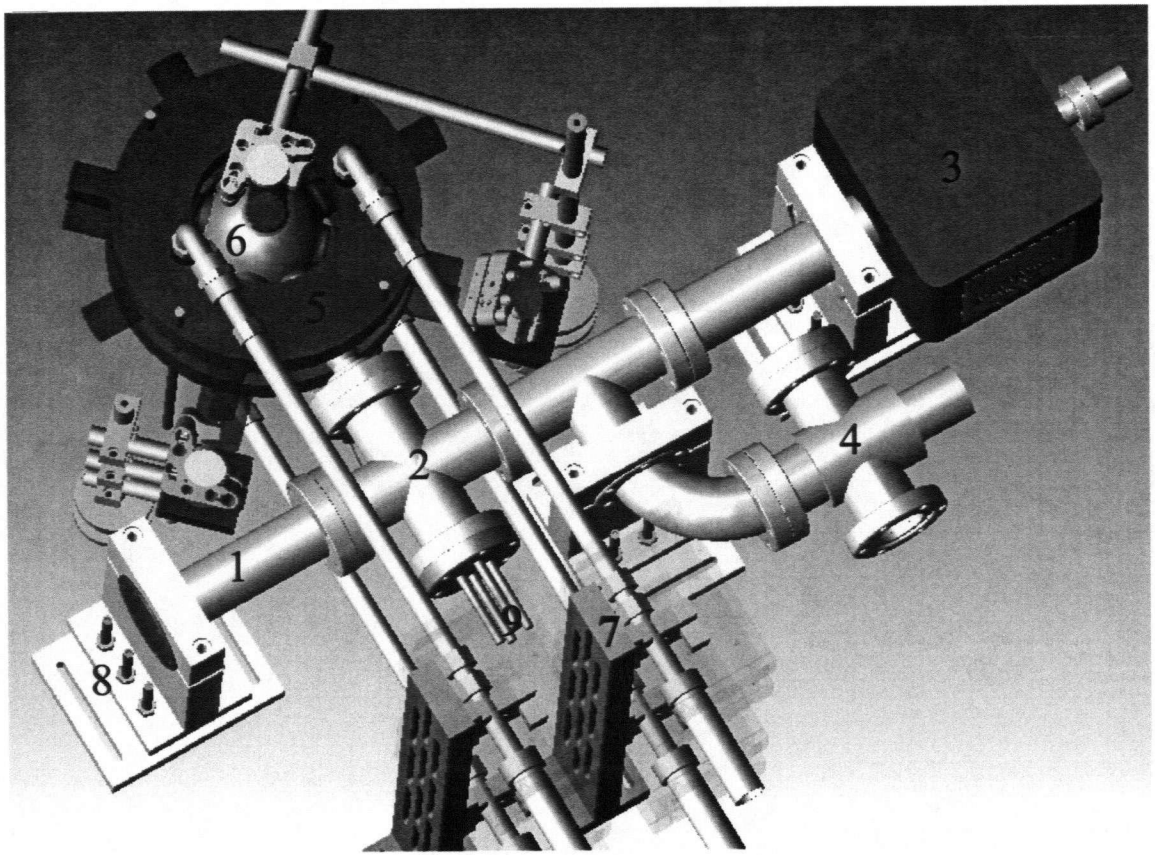


Figure 4.1: Experiment setup of the test chamber. 1: Getter Pump, CapaciTorr D400 from Saes Getters. 2: 4-way cross with dispensers inside, alkali metal dispensers from Saes Getters. 3: Ion Pump, VacIon Plus 20 (Varian, part number 9191145). 4: all-metal valve (Varian, part number 9515017). 5: Magnetic coils. 6: Glass chamber. 7: Support of the water pipes and electrical connections to the magnetic coils. 8: Support of the whole setup. 9: Electrical feedthru for the dispensers (MDC Vacuum, part number 640004).

4.1 Experimental Strategy for The Study of Hetero-nuclear Feshbach Resonances between Lithium and Rubidium

The experimental setup shown in Fig. 4.1 will be used to study the Hetero-nuclear Feshbach Resonances between Lithium and Rubidium. The procedure for this experiment will take the following sequence.

Firstly, a dual species MOT will be created to trap both ^{6}Li and Bosonic Rubidium istopes, ^{85}Rb or ^{87}Rb. Here the magnetic coils are set to produce a quadrupole magnetic field by using inverse current in the upper and lower coils, the current flow is about 2A which can provide a 10 Gauss/cm magnetic field gradient for our dual species MOT.

Once the atoms are cooled by the dual species MOT, an optical tweezer will be used to trap these atoms for later Feshbach resonance experiment. The optical tweezer is created by a 10mW CW Yb Fiber laser with center wavelength 1064 nm and linewidth <100kHz, after amplified by the Yb Fibe amplifier, it will give a 20W CW saturated output power. A resonance cavity will be used to enhance the optical dipole trap [33].

After the atoms are trapped by the optical tweezer, the magnetic coils are changed to Helmholtz coils, which provides a uniform magnetic field between the coils. The current here will be 10A $\sim$ 20A and the magnetic field can be adjusted between 0 $\sim$ 2k Gauss. At a specific magnetic field valve, the temperature and the number of trapped atoms will be measured by shining Lithium and Rubidium resonant lasers into the optical trap. Based on the measured thermalization rate, the inter-species cross section between Lithium and Rubidium can be determined. Also, by checking the relationship between the decay rate of the number of trapped atoms with corresponding magnetic field intensity, the Hetero-nuclear Feshbach Resonances between Lithium and Rubidium can be found. Since inelastic loss rate also diverges at the Feshbach resonance.

4.2 The Atomic Source

During the history of laser cooling and trapping experiments, many atomic sources have been tested and used to provide atoms for the loading of a trap, such as ovens, metal dispensers, Light Induced Atomic Desorption (LIAD) and laser ablation. The main point of all the atomic sources is to provide a large (and slowly moving) particle flux while not spoiling the vacuum so that on retains the long vacuum life time required for effective evaporative cooling to reach quantum degeneracy.

Initially ovens were used to produce a large atomic beams for MOT loading, but these sources were not ideal since the vapor pressure was too high and couldn't be lowered quickly. The next generation used two MOTs, where one MOT was used to trap atoms in a region of high background pressure and the precooled atoms are then transferred to a region of better vacuum where they are captured by a second MOT. A variation of this dual trap system involved a pure magnetic transfer of the

precooled atoms and requires only a single MOT, thus improves optical access for further experiments [34, 35].

Besides the large particle flux provided by ovens, the background pressure also increases to a very high point and the vacuum lifetime is greatly decreased. To retain low background pressure after the MOT phase, two other techniques have recently been investigated. One is the metal dispenser, which can be turned on/off very quickly (compared to ovens). In a UHV environment, many alkali atoms have a high room-temperature vapor pressure, and metal dispensers store the alkali atoms as alkali chromates with reducing agents. By ohmic heating, these alkali metal chromates, which have a very low room-temperature vapor pressure, undergo a chemical reaction with the reducing agents, and alkali atoms are liberated and escape out of the dispenser. The reducing agent is SAES' St101 (Zr 84%, Al 16%) getter alloy. In addition to reducing the alkali back to its metallic state, St101 also removes chemically reactive gases from the device, preventing them from contaminating the alkali metal vapor.

The other technique for providing atoms for trapping while still achieving a low background vapor is light induced atom desorption (LIAD). For incomplete coverings of the vacuum chamber walls, the vapor pressure of the alkali atoms in question can be well below the equilibrium room-temperature vapor pressure that one would expect from a chunk of the metal inside the chamber. In this case, atoms that are absorbed on the walls of a vacuum chamber can be quickly desorbed by irradiation with weak and incoherent light. This allows for a temporary increase of desired partial pressure. LIAD has been investigated for a number of alkalis since 1993 [36]. For the case of rubidium a 410 nm LED light has been found to greatly increase the MOT loading rate while keeping a long vacuum lifetime at the same time [37].

4.2.1 Rubidium Source

In our experiment, metal dispensers are used to provide rubidium atoms. These dispensers are mounted on copper conductors inside a four way cross and face towards the glass chamber, as shown in Fig. 4.2. This kind of orientation can help atoms sprayed from the dispensers pass through the glass tube and be collected inside the glass chamber. However, most of these atoms will be coated on the wall when they try to pass the glass tube, the LIAD method will be used to desorb the rubidium atoms on the walls for the MOT loading phase. This will be discussed further in the next chapter on dispenser calibration.

In order to maintain an ultra high vacuum inside the test chamber, we chose spot welding instead of screws to connect the dispenser terminals and the copper conductors. However, there are two problems concerning this technique in our application. Spot welding is a type of resistance welding used to weld various sheet metals. Typically the sheets are in the 0.5-3.0 mm thickness range. In our experimental setup, the copper conductor on the electrical feedthru is a cylinder with 0.25" diameter, which is not suitable for spot welding application. To solve this problem, vacuum soldering is used to first solder a piece of thin copper sheet to the top of the copper conductor, and then the dispenser terminal is spot-welded onto the copper extension

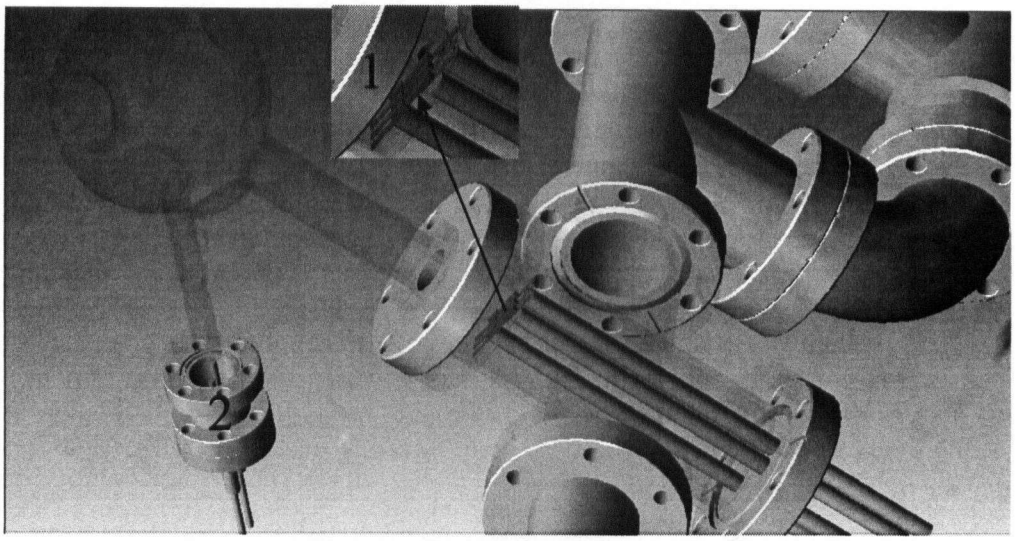


Figure 4.2: Atom sources for the test chamber. 1: metal dispensers which are mounted on the copper conductors by spot welding. Here the four way cross is set to be transparent for observation of the dispensers. 2: position for the lithium oven(Fig. 4.4), which is not shown in this design.

sheet. Another problem is the incompatibility between the dispenser terminals and the copper sheet. The dispenser terminal is made of nickel alloy (78% nickel and 22% chromium), and it is quite difficult to spot weld this nichrome terminal to the copper sheet. After investigation and lots of practice, we figured out a convenient way to do the job, details of this method are discussed in Appendix B.

4.2.2 Lithium Source

Like the rubidium source, a lithium metal dispenser is mounted on the copper conductor of the electrical feedthru as one possible lithium source for our experiment. However, the LIAD method may not work as well for lithium atoms as for rubidium atoms, and to ensure that we can get enough lithium atoms into our trapping region, a lithium oven was designed and built by Swati Singh.

The lithium oven is located at position 2 in Fig. 4.2, inside the small nipple. An electrical feedthru (1.33" flange size) is connected to the oven to supply necessary current. Schematics of the lithium oven are shown in Fig. 4.4, Fig. 4.3 shows an actual picture.

As pointed out in Fig. 4.4, stainless steel is used as the oven material. Inside the oven, there is a lithium metal chunk which will be heated by the current passing through the electrical feedthru. At a high temperature, lithium vapor will be emitted out of the oven, after being adjusted by the collimator, a lithium atomic beam will be pointed to the chamber center. Because it takes very long for the lithium oven to cool down, and we need the lithium source to be shut off quickly after the MOT loading, a shutter is placed above the collimator to block the lithium atomic beam

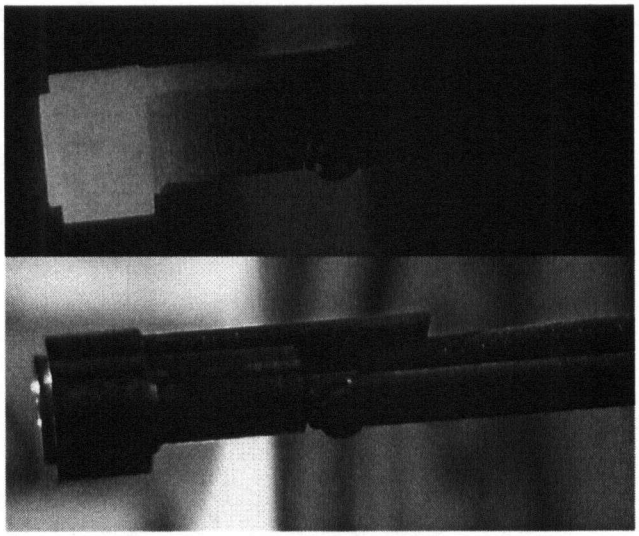


Figure 4.3: Outlook of the lithium oven, the top one is the picture of the working situation.

when necessary. This shutter is a piece of aluminum foil connected to the end of a tungsten wire which runs parallel to the lithium oven, the tungsten wire is fastened to the lithim oven by Kapton wire and has a magnet on the end. The magnet is spot-welded onto the tungsten wire. We can change the orientation of the magnetic and thereby rotate the shutter by changing the orientation of an applied magnetic field outside of the vacuum chamber.

4.3 Vacuum System

Three different kinds of vacuum pumps are used to realize the necessary ultrahigh vacuum environment in our experiment, the turbo pump (Turbo-V 70 from Varian), the ion pump (VacIon Plus 20 from Varian) and the Non-Evaporable Getter (NEG) pump (CapaciTorr D400 from Saes Getters).

4.3.1 Turbo Pump Station

Our turbo pump is located in a mobile station, the turbo pump station, as shown in Fig. 4.5. It includes an ionization gauge, thermocouple gauge, Residual Gas Analyzer (RGA) and a dry scroll pump, and can be moved around in our lab and connected to various chambers to pump them down independently. Table 4.1 shows the part list for our turbo pump station.

The turbo pump in our turbo pump station is Varian's Turbo V 70, which has a pumping speed varying from 42 L/S (for H_2) to 52 L/S (for He) for 2.75" ConFlat flange size. Because of the big volume of our vacuum system, the ultimate pressure for the turbo pump station is only 1×10^{-8} torr, about 10 times higher than the lower limit

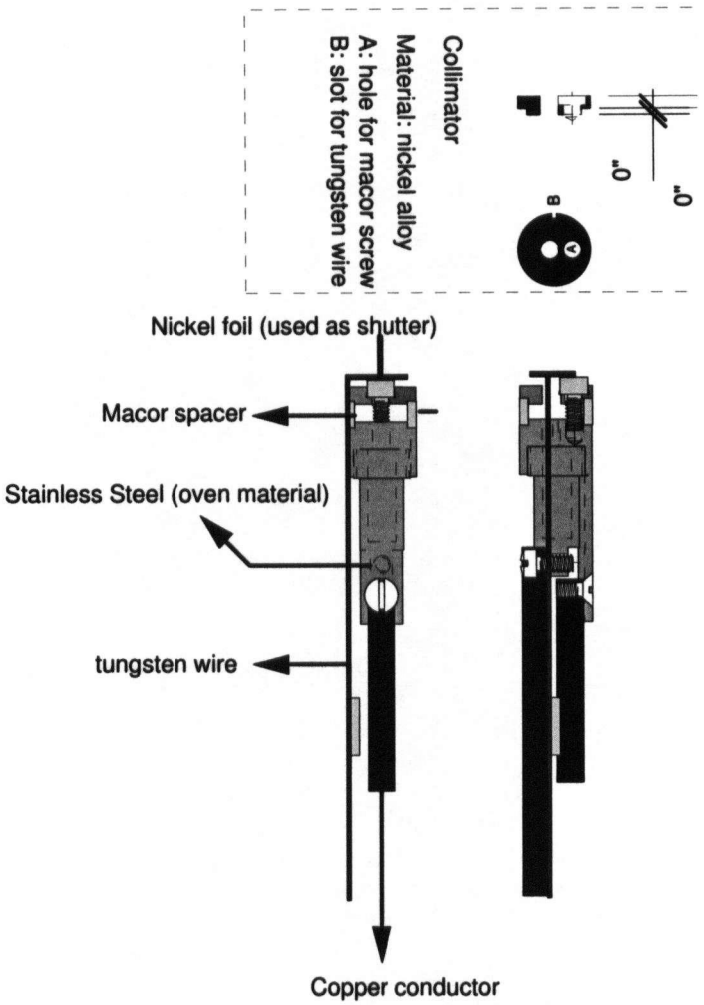


Figure 4.4: Schematics of the Lithium oven, which is designed by Swati Singh. Lithium source coming out of the stainless steel oven is calibrated by the collimator, a piece of aluminum foil is connected to the end of a tungsten wire (which is controlled by the magnet on it) to be the shutter.

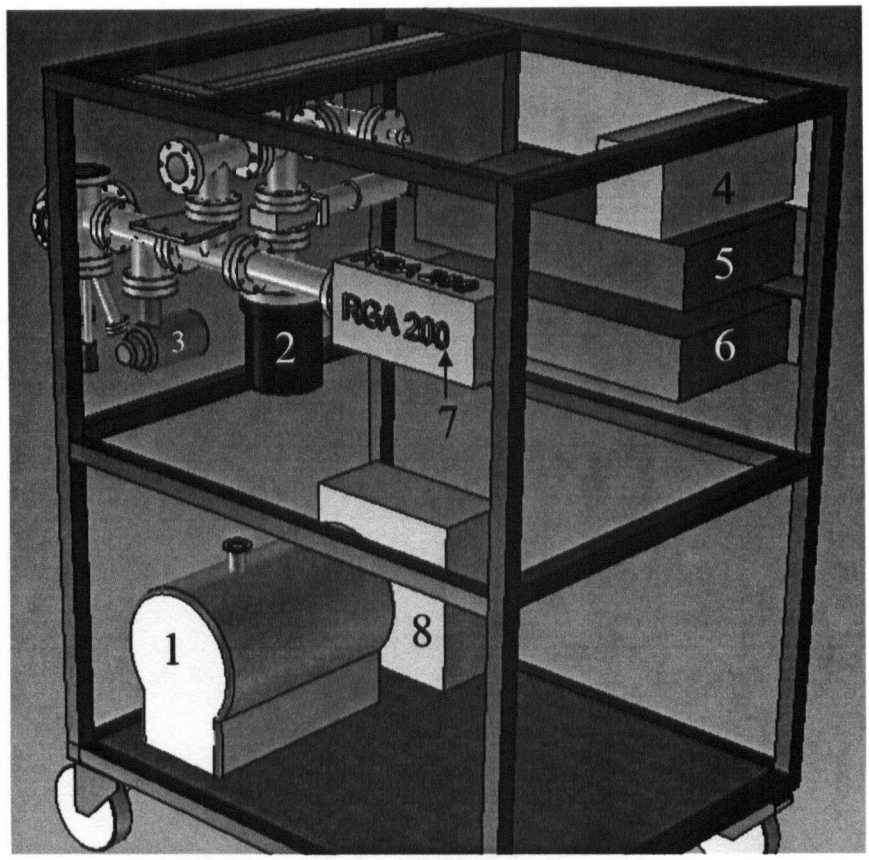


Figure 4.5: Turbo pump station. 1: SH-100 Dry Scroll Pump, 2: Turbo-V 70 Pump 3: Varian 571 Ionization gauge, 4: Turbo-V 70 Rack Controller, 5: 843 Ratiomatic Gauge Control, 6: temperature probing box, 7: RGA 200 Residual Gas Analyzer, 8: power box.

of the Turbo V 70. The Turbo V 70 controller is a microprocessor-controlled frequency converter with self-diagnostic and protection features that ensures the highest degree of reliability. It displays rotational speed as the pump starts up and indicates when full speed is reached. At any time during the operation of the pump, the speed, current, power, and bearing temperature can be displayed.

In order to be compatible with the turbo pump and to avoid the possibility of oil contamination of our pumping station or UHV vacuum system, a Varian SH-100 is used as our dry scroll pump. This single-stage pump produces a pumping speed of 100 L/m (on 60 Hz power) and achieves an ultimate pressure of 50 mtorr (0.07 mbar).

An Ionization gauge and thermocouple gauge are used to monitor vacuum pressure during the pumping down procedure, the acquired pressure values are displayed on gauge controllers of the 843 Ratiomatic Gauge Control box. The thermocouple gauge measures vacuum pressure from atmosphere to 1×10^{-3} torr). The ionization gauge tube offers high performance over a wide range of pressures (2×10^{-10} torr to

Table 4.1: Part list for turbo pump station.

Part	Company	Description
Turbo Pump	Varian	Turbomolecular pump, Turbo V 70
Turbo Pump Controller	Varian	Turbo V 70 controller
Dry Scroll Pump	Varian	Dry Scroll Pump, SH-100
Ionization Gauge	Varian	Bayard-Alpert type, 571 series
Ion Gauge Controller	Varian	843 Ratiomatic Gauge Control box
Thermocouple Gauge	Varian	Model 536
RGA	SRS	Residual Gas Analyzer, RGA 200

1×10^{-3} torr), it can withstand long periods of degassing or accidental exposure to atmospheric pressure while at operating temperature and will still recover its original characteristics. Because of these different ranges of measurements, TC gauge is used to monitor pressure when only the dry scroll pump is running, after the vacuum pressure drops to about miliTorr, the turbo pump is turned on and the ion gauge can continue with the vacuum measurement.

RGA 200 is the Residual Gas Analyzer from SRS (Stanford Research Systems). It offers exceptional performance and detailed gas analysis of our vacuum systems. The RGA system is comprised of a quadrupole probe, an electronics control unit (ECU), and a real-time Windows software package that is used for data acquisition and analysis, as well as probe control. The standard Faraday cup detector allows partial pressure measurements from 10.5 torr to 5×10^{-11} torr for all the elements between AMU (atomic mass unit) 1- 200. For increased sensitivity and faster scan rates, an electron multiplier is used and can detect partial pressures down to 5×10^{-14} torr.

4.3.2 Ion Pump

The ion pump in our vacuum system is Varian's Vaclon Plus 20 (StarCell version). It can handle a high amount of noble gases (better than Noble Diode version) and hydrogen (comparable to the Diode version).

4.3.3 Non Evaporable Getter Pump

To optimize the performance of the StarCell Ion Pump, Non Evaporable Getter Pump is added into our vacuum system. The NEG pump excels in pumping hydrogen, which is usually a shortcoming for the other kinds of pumps. However, It is very poor at pumping hydrocarbons at room temperature, and it will not pump noble gases at all. The combination of a NEG pump and an ion pump allows to contribute their strengths and for each to compensate for the deficiencies of the other.

NEG pumps also have some other very interesting capabilities that make them attractive in many circumstances. They usually operate without power. They are clean, lightweight, compact and vibration-free. They can help provide better ultimate

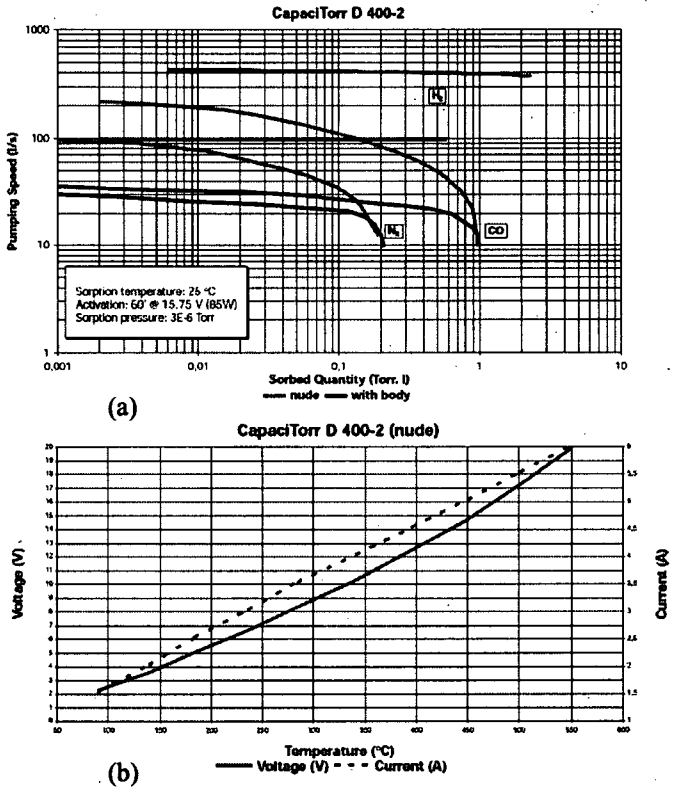


Figure 4.6: (a): Sorption curve for CapaciTorr D400-2, (b): Heating curve for CapaciTorr D400-2.

vacuum than is achievable without them. They operate unaffected by magnetic fields, and do not generate magnetic fields.

In our vacuum system, CapaciTorr D 400-2 from SAES Getters is chosen as our getter pump. It uses St 172 (Zr-V-Fe) material in form of disks to achieve high pumping performance in a very compact configuration. Fig. 4.6 shows the sorption and heating curves. NEG pumps are usually shipped with a passivation layer on the getter alloy, they need to be activated to be ready for use. For St 172 cartridges, 45 minutes heating at 450 °C (4.7A) is sufficient for activation. Because we use the NEG pump inside UHV environment and the gas load is very small, reactivation of the pump can be greatly reduced.

4.3.4 Baking System and Pumping Sequence

Baking of the vacuum system is essential for achieving ultra high vacuum. It can increase the outgassing rate of the vacuum chamber, and thus reduce the pumping time to get the vacuum system clean.

Our baking system can be divided into two parts. One is the baking of the turbo pump station, which is done by covering all the bakeable parts with heating tapes

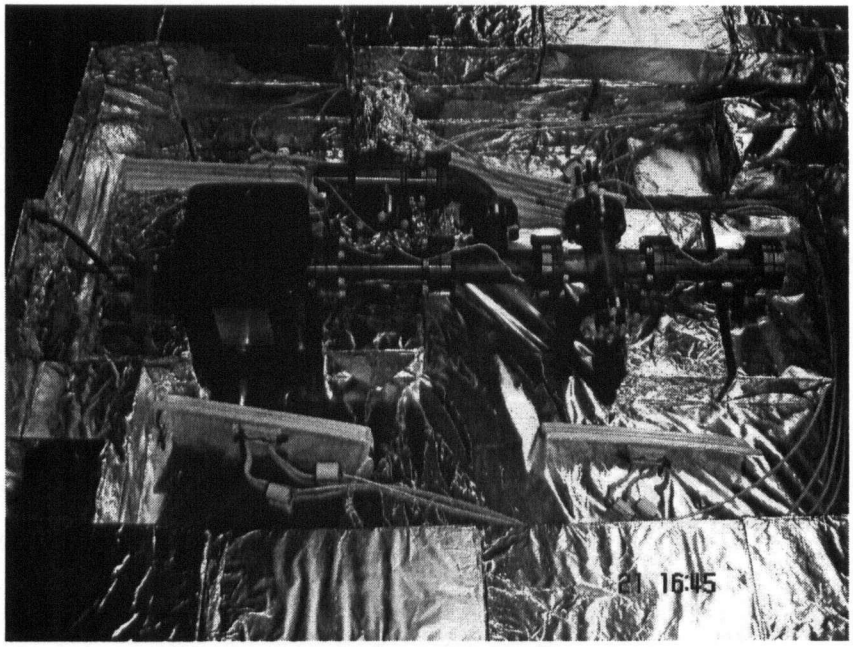


Figure 4.7: Brick oven for hard baking our test chamber. In this picture the glass chamber is not connected, due to the big baking temperature difference between the glass chamber and the other vacuum parts. Infrared ceramic heaters (the white tile in front of the black ion pump) are used to heat the oven.

(Omega, FGS/FGH/FWH Series) and adjusting power through each tape to achieve the desired temperature. The other one is the baking of the vacuum part in Fig. 4.1, which will be put inside a large oven. The oven is built of ceramic "fire brick" that are coated with aluminium foil to contain the dust created when the bricks crumble, as shown in Fig. 4.7. Ceramic infrared heaters (Salamander Ceramic, IRCER10272) are used to control the temperature inside the oven.

All the heaters we used are controlled by variacs inside a mobile variac station, each variac is adjustable between 0-120V. The baking temperature of every part is monitored by a temperature probing box I made, which is capable of monitoring 16 channels of thermocouple probes at the same time, details about this box will be discussed in Appendix A. The maximum baking temperature for all the bakeable parts in our vacuum system is shown in Table 4.2.

Because the maximum bakeout temperature for the glass chamber is much smaller than the steel part, and the NEG pump doesen't need to be baked, we divide our baking procedure into two steps. In the first step, we bake the test chamber inside the brick oven at a very high temperature (400 °C) without glass chamber and NEG pump, this "hard bake" lasts about one week. The turbo pump station is connected to the test chamber by a flexible bellows through the all-metal valve to pump the system during the bakeout. Baking of the turbo pump station is carried out at the same time. During the second step, the test chamber and turbo pump station

Table 4.2: Maximum bakeing temperature for different vacuum parts.

Part	Maximum Baking Temperature
RGA probe	300 °C
Variable Leak Valve	450 °C
All-metal Valve	450 °C
Turbo-V 70	120 °C
Ionization Pump	350 °C
NEG Pump	×
Nipple, 2.75" Flange	450 °C
Rotary Feedthrough	200 °C
Electrcal Feedthrough	450 °C
Quartz Viewport	200 °C
Sapphire Viewport	450 °C
Zero profile wide angle glass Viewport	400 °C

are firstly cooled down to room temperature, the vacuum pressure now will drop to less than 10^{-8} torr. Next we vent the test chamber to dry nitrogen and open it to install the glass chamber and the NEG pump, during the installation, clean and dry nitrogen will flow through the vacuum system to prevent contaminants (water, carbon compounds, etc.) from getting inside. After installing the glass chamber and the NEG pump, the test chamber will be baked again at a lower temperature (200 °C) for several days. Before cooling down, the metal dispensers and Lithium oven will be heated below threshold to release possible contaminants. After the second baking is finished and the whole vacuum system is cooled down, the NEG pump will be activated and the all-metal valve will be closed. The turbo pump station will be shut down and disconnected. The ion pump and the NEG pump will continue to pump down the system and maintain vacuum pressure around 10^{-11}~10^{-10} torr.

Chapter 5

Calibration of The Alkali Dispensers

As we discussed in section 4.2, metal dispensers are adopted as our Rubidium atomic source and a potential source for lithium atoms. To get a better understanding of the performance of these dispensers, a calibration experiment was conducted on the Rubidium dispenser to investigate its properties, such as the atomic partial pressure, resistance, heating time constant, etc. This experimental setup was designed and used to test LIAD for Rb+Li; However, our detection sensitivity was too small to study this phenomenon.

This chapter can be divided into three parts. Section 5.1 describes the experimental setup for the calibration experiment. Section 5.2 explains the procedure for the experiment. Finally, Section 5.3- 5.5 discuss the various properties of the Rubidium dispenser.

5.1 Experiment Setup

5.1.1 Dispenser holder

Because we need to investigate the properties for both the Rubidium and lithium dispensers, the dispenser holder was designed to be capable of installing two dispensers. Fig.5.1 shows the 3D design for the dispenser holder. The dispensers used for the calibration experiment are mounted on an aluminium support, which is resting on the multi-channel reducing flange (2.75" to 1.33"). The top end of the dispensers is connected to the aluminum support, the other end is connected to copper connectors which are separated from the aluminum support by a Macor plate. Electrical power is supplied to the dispensers by Kapton wires that are connected to the aluminum support and the copper connectors.

The multi-channel reducing flange has three extension tubes for connecting electrical feedthroughs, and one smaller tube in the center for the rotary shaft, which is used for holding the glass slide. The rotary shaft can be rotated outside the vacuum and the angle of the glass slide is adjusted correspondingly. Atoms sprayed out of the dispensers can be coated on the glass slide and then used for testing the light induced photo desorption mechanism.

The whole setup of the dispenser holder is placed inside a six-way cross, which is connected to the turbo pump station (Fig.4.5). In the calibration experiment, vacuum pressure doesn't need to be very low, the turbo pump only is able to pump down the system to around 10^{-8} torr and satisfies the vacuum requirement for this experiment.

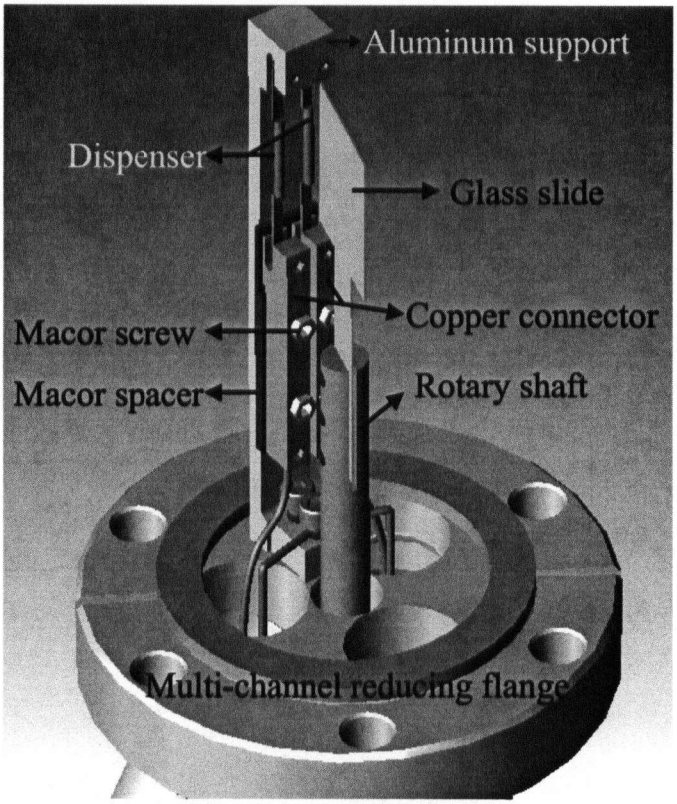


Figure 5.1: 3D design of the dispenser holder.

5.1.2 Optical setup

The optical setup for the dispenser calibration experiment is shown in Fig.5.2, which is the top view from the six-way cross. Light coming from the diode laser is locked to be resonant with Rubidium D2 transition by Doppler-free saturated absorption spectroscopy method.

Two beams are sent into the six-way cross through the front window. The one passing between the dispensers and the glass slide is called as the inside beam, it will be detected by photo diode A, the other one behind the glass slide is called as outside beam and will be detected by photo diode B. In this setup, both the inside and outside beams are resonant light, the partial pressure on both sides of the glass slide can be detected. Some of the tests require us to measure the transmission through the thin film of alkali metal that is deposited on this slide by the dispenser in order to estimate the atom flux emitted by the dispenser as well as the evaporation rate from the slide. In this case the outside beam will be replaced with a non-resonant transmission light and the position will also be moved so that the transmission beam passes through the center of the glass slide.

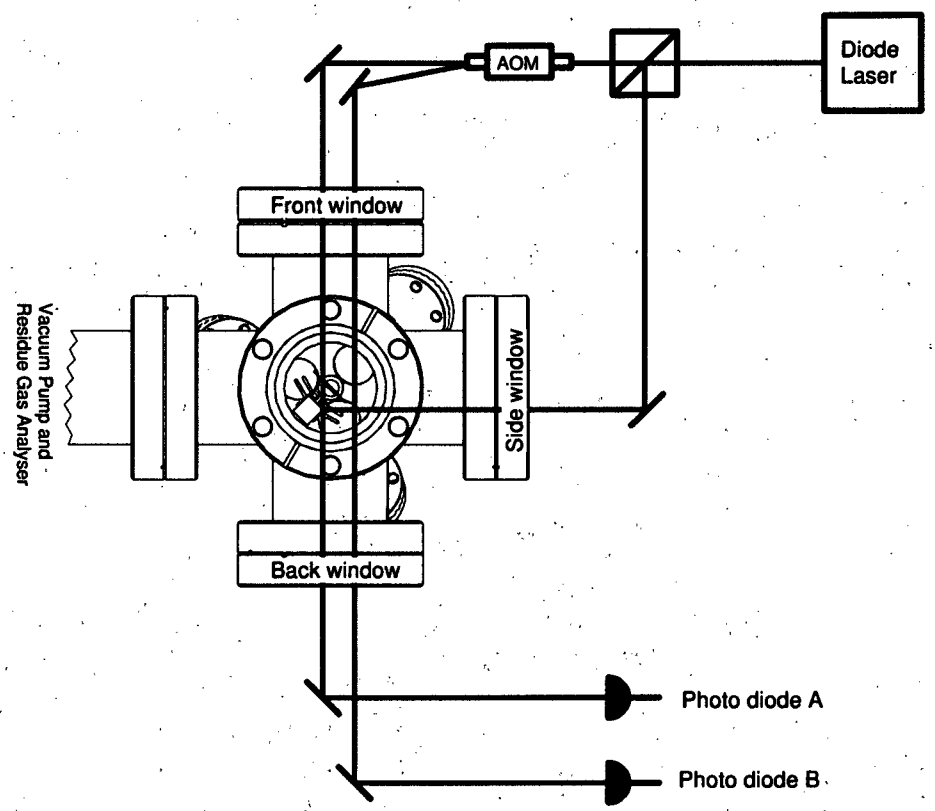


Figure 5.2: Optical seutp for the calibration experiment.

5.2 Experiment Procedure

This experiment can be divided into three periods. Fig. 5.3~ 5.5 (corresponding to periods 1~3) show the time evolution of the partial pressure for different elements, N_2,^{85}Rb, ^{87}Rb, ^{6}Li, ^{7}Li and H_2O. The partial pressure is measured by our Residual Gas Analyzer, RGA200. Nitrogen is the most dominant vacuum background component, it can give us valuable information about the quality of the vacuum environment. The partial pressure of H_2O, on the other hand roughly tells us whether there is a leak in the vacuum system or not.

It is true that the atomic mass units we monitored can also represent other kinds of chemicals. For example, AMU 28 can also be CO_2, and AMU 18 can also be CH_4. However, since N_2 and H_2O are more common in vacuum than the other chemicals having the same molecular weight as them, here in this experiment, we call AMU 28 as nitrogen, and AMU 18 as water.

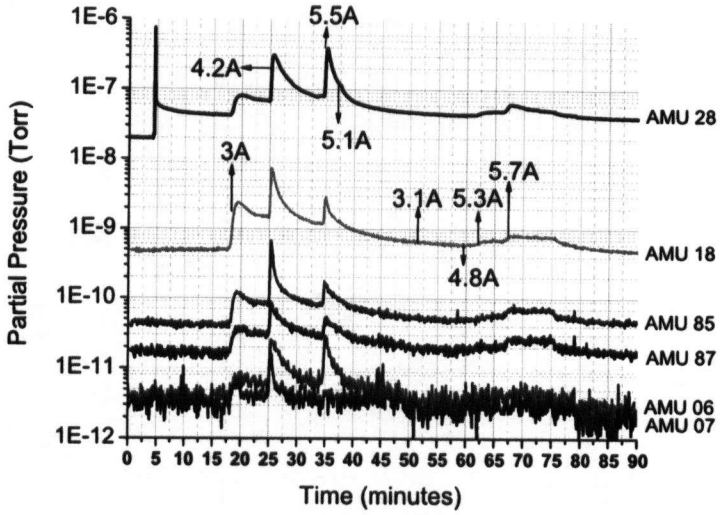


Figure 5.3: Degassing of the dispenser. Partial pressure of different elements inside vacuum chamber was recorded by RGA. The up arrows indicate current changes at related time.

5.2.1 First period: degassing period

During the first period, the Rubidium dispenser is firstly heated up by increasing current gradually to threshold (~5.3A), the current through the dispenser is then changed several times and finally the dispenser is turned off. The chamber pressure measured by ion gauge was below 10^{-8} torr before the heating. At 3A (minute 17:30) we saw partial pressure increase of all these five elements, as shown in Fig. 5.3, this heating results in the degassing of the dispenser, similar to the degassing of the ion gauge filament. We believe that the Rubidium and lithium atoms produced in this period do not come from the supposed chemical reaction ACl+B$\longrightarrow$BCl+A, where A is the alkali we want to produce, B is the reducing agent. But rather from alkali atoms stuck to the outer surface of the dispenser. One reason is the immediate decay of the partial pressure after each peak; another reason is, later on when we heated the dispenser again at 50:00 minutes while below threshold current (5.1A), we didn't see partial pressure increase again.

One weird thing is the partial pressure increase of the AMU 28 element at about 05:00 minute, since there's no partial pressure increase for the other elements. One possibility is that a small amount of this kind of gas is trapped in somewhere of the dispenser holder, and released when we heat the dispenser.

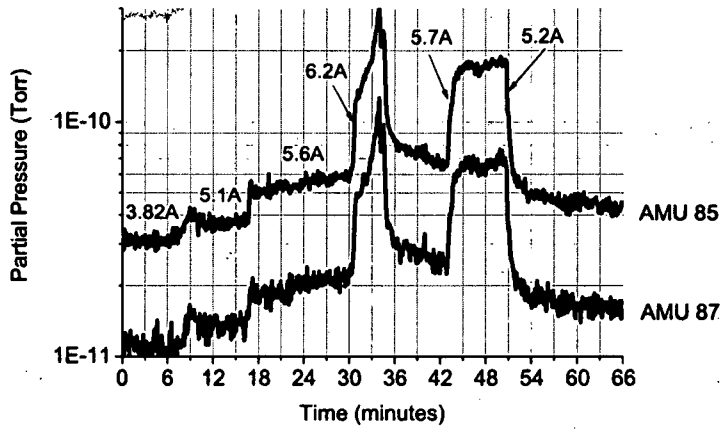


Figure 5.4: Dispensing Rubidium atoms by heating above threshold. Partial pressure of different elements inside vacuum chamber was recorded by RGA, based on the data collected in the second period. The up arrows indicate current changes at specific time. The average partial pressure of ^{85}Rb during each step is also shown at corresponding position.

5.2.2 Second period: heating above threshold to dispense Rubidium atoms

Fig. 5.4 records the experimental procedure of the second period. Based on the voltage and current data, we plot the Voltage-Current curve to investigate the behavior of the getter resistance during the heating process. Although we didn't check the absorption signal at this time, we still got useful information, which will be discussed in the following sections. Before taking the data shown in Fig. 5.4, the chamber pressure measured by ion gauge is 2×10^{-8} torr.

5.2.3 Third period: measuring absorption signal and finding Rubidium partial pressure

Fig. 5.5 records the experimental procedure of this period. This time, besides the usual tests, the absorption signal is also measured so that the Rubidium partial pressure can be deduced. Different light sources, Rubidium resonant laser and a 650W light bulb are used to investigate the effect of external radiation on the flux of Rb atoms coming out of the dispenser, and on changes to the electrical resistance of the dispenser. This test was done to investigate claims that dispenser output can be modified by incident radiation. Since the external radiation can provide additional heat to the dispensing chemical reaction. These external lights are sent into the six-way cross through the side window and focused on the Rubidium dispenser.

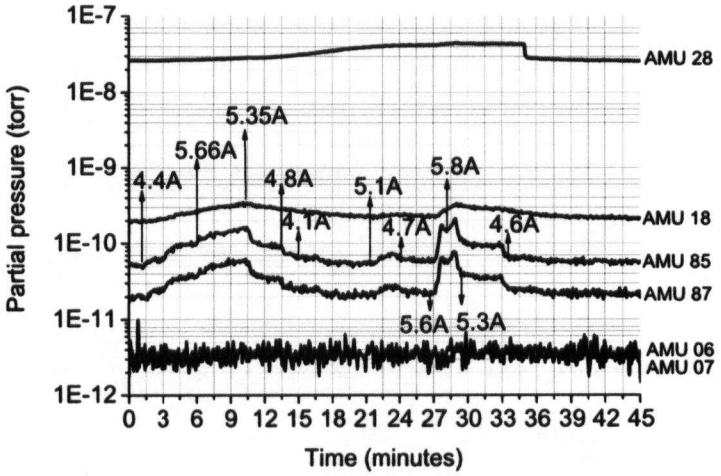


Figure 5.5: Partial pressure of different elements inside vacuum chamber was recorded by RGA, based on the data collected in the third period. The up arrows indicate current changes at related time.

5.3 Getter Resistance

The current and voltage across the Rubidium dispenser are measured during the heating process, consequently we plot the V-I curve to figure out how the Getter resistance changes during the experiment. There are two graphs showing this, Fig. B.4 (a) and (b), corresponding to the second and third period respectively.

It's quite clear that the V-I curve is linear, which means the Getter resistance is constant. From the V-I curve, we get an approximate value for Getter resistance, which is 0.25~0.26 ohms, this value is the series resistance of the dispenser and the conductor (about 0.08 ohms). A direct resistance measurement of some burned dispensers is also done using a Hewlett Packard 34401A multimeter, the result is 0.17±0.01 ohms for the dispenser only. Clearly the Getter resistances from these two methods are quite close.

5.4 Time Constant of The Rubidium Dispenser

Because it is important to load the MOT but still maintain a good vacuum, we investigated the speed at which the disensers could be switched on and off.

Based on the data from Fig. 5.4, we calculated the time constant for the heating and cooling of the Rb dispenser. We choose the big increase at 42:00 minutes to do the analysis. Fig. 5.7 (a) gives the amplified plot in this range, and also the time axis has been reset to start from the beginning of the pressure increase. Fig. 5.7 (b) gives us the relationship between time t and $ln(\frac{p(t_f)-p(t)}{p(t_f)-p(t_0)})$, the slope of the line is just

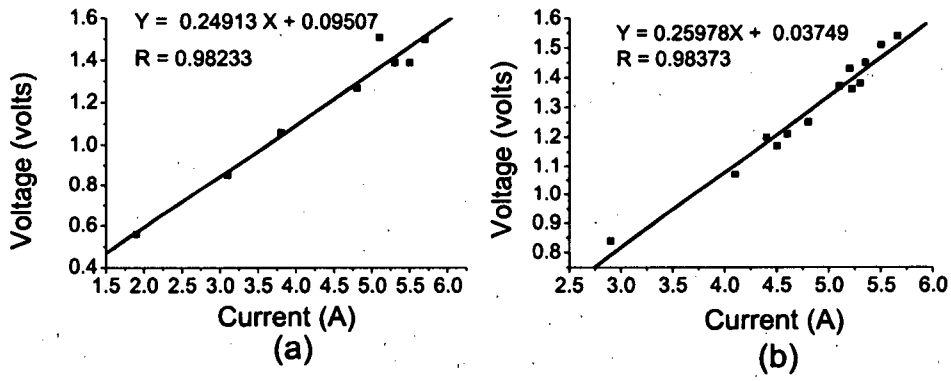


Figure 5.6: (a): voltage and current across the Rubidium dispenser, based on the data collected in the second period. (b): voltage and current across the Rubidium dispenser, based on the data collected in the third period.

$-1/\tau$, since

$$p(t) = p(t_f) - (p(t_f) - p(t_0))e^{-t/\tau} \tag{5.1}$$

Now we can find the time constant for the heating procedure, which is $\tau = 1/0.0224 \simeq$ 45 seconds.

5.5 Estimation of The Rubidium Vapor Pressure generated by the atom flux from dispenser

To get a quantitative estimation of the Rubidium vapor pressure inside the six-way cross produced by the atom flux from the dispenser, we created a model and found a general formula (Eqn.5.12) to describe the relationship between vapor pressure and the attenuation parameter of the absorption signal. And then we performed an absorption experiment on the Rubidium vapor cell to calibrate the general formula for Rubidium.

5.5.1 Model for the atomic vapor pressure from an absorption measurement

A beam of N photons (or any particles for that matter) propagating along z in a scattering medium will lose particles as

$$dN(z) = N(z)p(z)dz \tag{5.2}$$

where $p(z)$ is the probability per unit length of a scattering event and is given by

$$p(z)dz = n_0\sigma dz \tag{5.3}$$

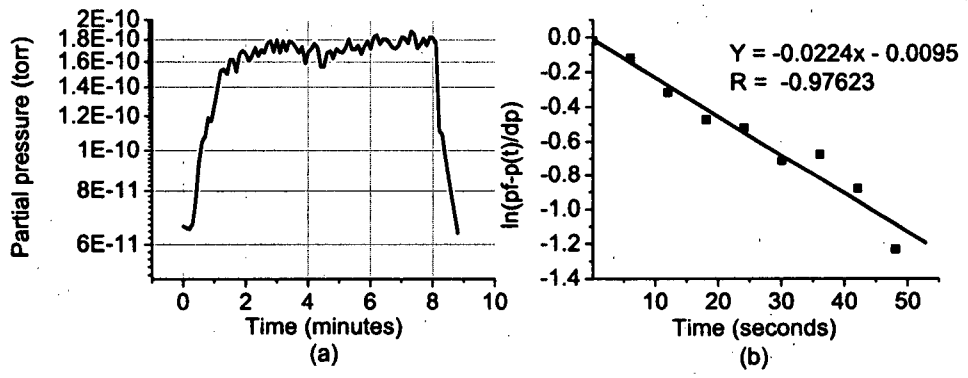


Figure 5.7: (a): Plot of some data from Fig. 5.4, from 42:00 minutes to 54:00 minutes. (b): relationship between time t and $ln(\frac{p(t_f)-p(t)}{p(t_f)-p(t_0)})$ of the increasing part in Fig. 5.7 (a), from which we can find the time constant of the heating procedure. Based on the data collected in the second period.

where n_0 is the spatial density of scatterers and σdz is the cross-sectional volume swept out by the moving particle. Integrating the equation for dN over a path of length L gives an attenuation exponent of

$$a = \ln(\frac{N(L)}{N(0)}) = -\int_0^L p\, dz \tag{5.4}$$

For a photon in an atomic vapor, the cross-section is frequency dependent and given by

$$\sigma = \sigma_0 \frac{\Gamma^2/4}{\delta^2 + \Gamma^4} \tag{5.5}$$

where $\Gamma = 2\pi/\tau$ is the atomic line width (τ the excited state lifetime), $\sigma_0 = \lambda^2/2\pi$ is the resonant cross section, and $\delta = \omega$ - ω_{atom} is the frequency "detuning" of the incident photon from the atomic resonance. Although the spatial density in an atomic vapor is uniform, the atoms are moving due to their thermal energy and this motion doppler-shifts the atomic resonance by an amount $k_L v_z$ where $k_L = 2\pi/\lambda$ is the wavevector and v_z is the component of the velocity along the photon beam axis. The spatial density of atoms at a given velocity is given by a Maxwell-Boltzman distribution

$$n(v_z) \propto n_0 e^{-\beta \frac{1}{2} m v_z^2} \tag{5.6}$$

where $\beta = (k_B T)^{-1}$. We consider that the laser is on resonance and has a negligible line width so that the detuning is simply $\delta = k_L v_z$, and so the density can be written as a function of the velocity dependent detuning as

$$n(\delta) = \frac{n_0}{\sigma_\delta \sqrt{2\pi}} e^{-\frac{\delta^2}{2\sigma_\delta^2}}. \tag{5.7}$$

where now $n(\delta)$ is normalized to n_0 and the thermal broadening is given by

$$\sigma_\delta^2 = \frac{k_L^2}{m\beta}. \tag{5.8}$$

Now the attenuation factor, Eqn.5.4, can be calculated by integrating over all detunings (velocities) and over the path of length L giving

$$a = -L\int_{-\infty}^{+\infty} n(\delta)\sigma\delta d\delta. \tag{5.9}$$

Since the typical atomic line width $\Gamma \sim 10$ MHz is much smaller than the doppler broadening width $\sigma_\delta \sim 1$ GHz, we can approximate $\sigma(\delta)$ with a dirac delta function with the same area

$$\sigma(\delta) \approx \sigma_0 \frac{\Gamma}{2}\pi\delta(\delta) \tag{5.10}$$

so that the integral is easily evaluated and we have

$$a = -L\frac{n_0}{\sigma_\delta\sqrt{2\pi}}\sigma_0\frac{\Gamma}{2}\pi. \tag{5.11}$$

Using the ideal gas law, $P = n_0/\beta$, we can express the pressure as a function of the attenuation exponent

$$P = \frac{-ak_L^3\sqrt{2}}{\Gamma L\sqrt{m\beta^3\pi^3}}. \tag{5.12}$$

For the specific case of a room temperature vapor of ^{85}Rb and ^{87}Rb we have

$$k_B = 1.3804\times10^{-23}\,\text{J/K} \tag{5.13}$$

$$T = 296\,\text{K},\ \lambda = 780.02\,\text{nm} \tag{5.14}$$

$$\Gamma = 2\pi\times5.41\,\text{Mhz},\ m = 1.45\times10^{-25}\,\text{kg} \tag{5.15}$$

$$1\,\text{bar} = 1\times10^5\,\text{Pa},\ 1\,\text{torr} = 132\,\text{Pa},\ 760\,\text{torr} = 1\,\text{bar} \tag{5.16}$$

and therefore that

$$P = \frac{a}{L}\times2.6787\times10^{-8}\,\text{mbar m} \tag{5.17}$$

or equivalently

$$P = \frac{a}{L}\times2.0293\times10^{-8}\,\text{torr m} \tag{5.18}$$

5.5.2 Calibration of the formula about Rubidium vapor pressure

Eqn.5.18 describes the relationship between Rubidium vapor pressure and the attenuation exponent a, which is equal to $\ln(I/I_0)$. The theoretical parameter , 2.0293×10^{-8}, has to be calibrated by absorption experiment.

To calibrate this parameter, we sent a resonant laser light through the Rubidium vapor cell (L=7.5cm), and then ramped the laser frequency around resonance to get the absorption signals of both Rubidium isotopes, ^{85}Rb and ^{87}Rb, as shown in Fig. 5.8. In which, the black line is the absorption signal and the laser power coming out of the vapor cell is 6.45μW. We also tried different input laser powers, as shown in Table 5.1, to insure the transition was not power saturated. Based on the data in this table, the average attenuation parameter is found to be $a = \ln(I/I_0) \simeq$ -0.93 for ^{85}Rb.

P(μW)	I_0(volts)	I(volts)	I/I_0	ln(I/I_0)
2.15	0.0074	0.0029	0.39189	-0.93677
4.25	0.0103	0.0040	0.39024	-0.94098
5.25	0.0265	0.0105	0.39623	-0.92577
6.45	0.0415	0.0165	0.39759	-0.92233

Table 5.1: Absorption data for ^{85}Rb at different input laser power. I_0 and I are input and output signal strength respectively, the background light intensity has been excluded.

The room temperature Rubidium vapor pressure can be calculated out using the formula from Daniel Steck's Rubidium 87 D Line Data[23].

$$\log_{10}P = -94.04826 - \frac{1961.258}{T} - 0.03771687\cdot T + 42.57526\log_{10}T \qquad (5.19)$$

$$\log_{10}P = 15.88253 - \frac{0.4529.635}{T} + 0.00058663\cdot T - 2.99138\log_{10}T \qquad (5.20)$$

Eqn. 5.19 and Eqn. 5.20 describe the vapor pressure of solid phase Rubidium and liquid phase Rubidium respectively. The validity of these two equations has been checked by calculating Rubidium vapor pressure at 39.64°C, and comparing with the one we got on internet, 1.18×10^{-6} Torr. The results from these two equations are very similar, both equal to 1.32×10^{-6} Torr and very close to 1.18×10^{-6} Torr.

Now we can find the room temperature Rubidium vapor pressure using either Eqn. 5.19 or Eqn. 5.20. The room temperature in our lab is about 22°C, the room temperature Rubidium vapor pressure is therefore 2.1×10^{-7}torr. Taking into account the isotope abundance, the vapor pressure for ^{85}Rb at room temperature should be $2.1\times10^{-7} \times 73\% = 1.53\times10^{-7}$ torr

By substituting the attenuation parameter a and the room temperature vapor pressure for ^{85}Rb into Eqn. 5.18, we got the experimental value, 1.23×10^{-8} torr.

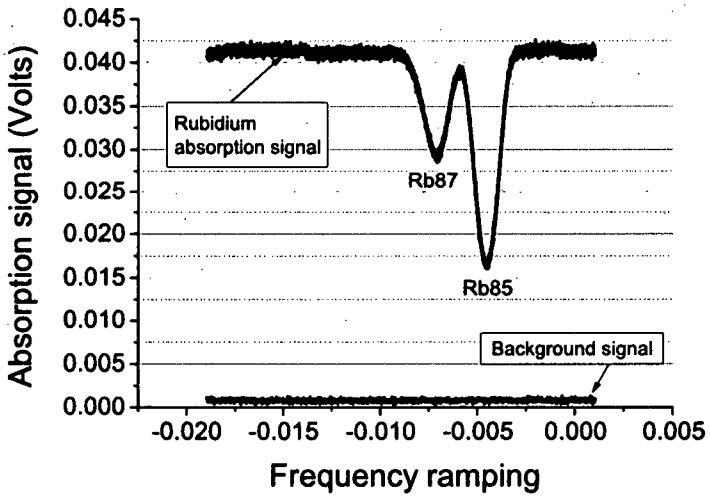


Figure 5.8: One example of the Rubidium absorption signal through a 7.5cm long vapor cell, the output laser power is 6.45μW.

Comparing with the theoretical value, 2.0293×10^{-8} torr, it's clear that the above model is very successful for describing the Rubidium vapor pressure.

Finally, the vapor pressure equation we should use for the Rubidium calibration experiment takes the following form.

$$P = \ln(\frac{I}{I_0}) \times \frac{1.23\times10^{-8}}{0.126} = \ln(\frac{I}{I_0}) \times 9.76190476 \times 10^{-8} \text{ torr m} \qquad (5.21)$$

where 0.126 (meter) is the length for our six-way cross. Based on the vapor pressure calculated from Eqn.5.21, the relationships between Rubidium vapor pressure and the current/power through the Rubidium dispenser are shown in Fig.5.9.

5.6 Conclusion

The performance of the Rubidium dispensers which will be used as our Rubidium atomic source for the future ultra-cold collision experiment was studied and several properties for the Rubidium dispenser were investigated.

Getter resistance was found to be $0.17 \sim 0.18$ ohms and independent of power for the operating currents. This finding was important since we would like to control the power delivered to the dispensers. Based on the voltage and current data from the first and second periods. This is also consistent with the direct resistance measurement (0.17 ± 0.01 ohms) using Hewlett Packard 34401A multimeter.

Based on the data from a heating process in the second period, the heating time constant for the Rubidium dispenser was found, which is about 45 seconds. This

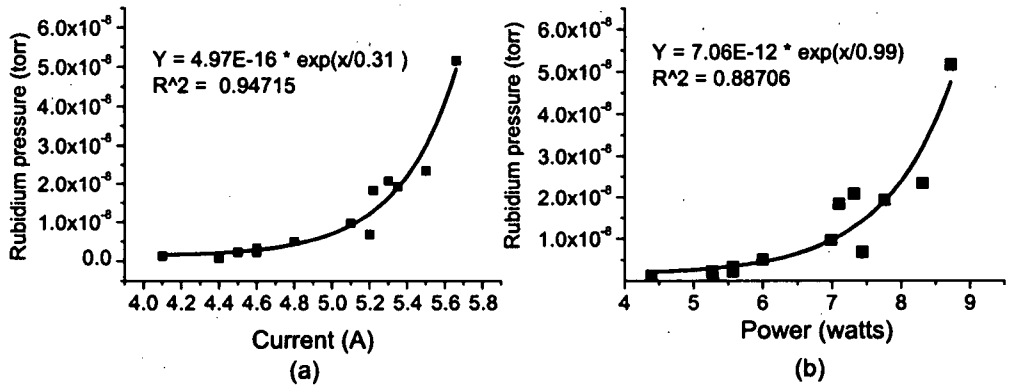


Figure 5.9: (a): ^{85}Rb pressure vs current through the Rubidium dispenser, based on the data collected in the third period. (b): ^{85}Rb pressure vs power through the Rubidium dispenser, based on the data collected in the third period.

timescale sets a limit on the time the vacuum will take to recover after the MOT loading phase.

To describe the vapor pressure produced by the Rubidium dispenser, a model was developed by us and parameters in this model were calibrated by a test of the absorption signal for resonant Rubidium laser, when the light is passing through a Rubidium vapor cell. Using this, we calibrated the flux produced by the dispensers for different operating currents.

Finally, Light Induced Atomic Absorption was investigated but the sensitivity of our absorption probe was not high enough to see changes in the partial pressure induced by external radiation.

Bibliography

[1] M. Ayamar and O. Dulieu, The Journal of Chemical Physics, 122, 204302 (2005).

[2] R. Côté, R. Onofrio and E. Timmermans, Phys. Rev. A 72 041605 (2005).

[3] C. Silber, S. Günther, C. Marzok, B. Deh, Ph. W. Courteille, and C. Zimmermann, Phys. Rev. Lett., 95, 170408 (2005).

[4] K. Huang. Statistical Mechanics. Wiley, New York (1963).

[5] M. H. Anderson, J. R. Ensher, M. R. Matthews, C. E. Wieman, and E. A. Cornell, *Observation of Bose-Einstein Condensation in a Dilute Atomic Vapor*, Science 269, 198 (1995).

[6] C. C. Bradley, C. A. Sackett, J. J. Tollet, and R. G. Hulet, *Evidence of Bose-Einstein Condensation in an Atomic Gas with Attractive Interactions*, Phys. Rev. Lett. 75, 1687 (1995).

[7] K. B. Davis, M.-O. Mewes, M. R. Andrews, N. J. van Druten, D. S. Durfee, D. M. Kurn, and W. Ketterle, *Bose-Einstein Condensation in a Gas of Sodium Atoms*, Phys. Rev. Lett. 75, 3969 (1995).

[8] W. Ketterle and N.J. van Druten. Evaporative Cooling of Trapped Atoms. Adv. Atom. Mol. Opt. Phys. 37, 181 (1996).

[9] E. Fermi. Sulla quantizzazione del gas perfetto monoatomico. Rend. Lincei, 3, 145, (1926).

[10] E. Fermi. Zur Quantelung das idealen einatomigen Gases. Z. Phys., 36, 902, (1926).

[11] L. Spitzer. Physics of Fully Ionized Gases. Interscience, New York, (1962).

[12] J. Söding, D. Guéry-Odelin, P. Desbiolles, F. Chevy, H. Inamori, and J. Dalibard, Appl. Phys. B. 69 257, (1999).

[13] H. Feshbach, *A unified Theory of Nuclear Reactions*, Ann. Phys. 5, 337 (1958).

[14] H. R. Thorsheim, J. Wiener and P. S. Julienne, Phys. Rev. Lett. 58 (1987) 2420.

[15] D. Guéry-Odelin, J. Söding, P. Desbiolles, and J. Dalibard, *Strong evaporative cooling of a trapped cesium gas*, Optics Express, Vol. 2, No. 8 (1998).

[16] B. Einarsson and Y. Shokin, *Fortran 90 for the Fortran 77 Programmer*, http://www.nsc.liu.se/~boein/f77to90/f77to90.html.

[17] F. Ercolessi, *A molecular dynamics primer*, an excllent beginner's book.

[18] G.E.P Box, M.E. Muller, *A note on the generation of random normal deviates*, Annals Math. Stat, Vol. 29, 610-611, (1958).

[19] H. Wu and C. Foot, *J. phys. B*, 29, L321-L328(1996).

[20] M. D. Barrett, J. A. Sauer, and M. S. Chapman. All-Optical Formation of an Atomic Bose-Einstein Condensate. Phys. Rev. Lett., 87, 010404, (2001).

[21] D. E. Pritchard. Cooling of neutral atoms in a magnetic trap for precision spectroscopy. Phys. Rev. Lett., 51, 1336, (1983).

[22] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, *Numerical Recipes in Fortran 77: The Art of Scientific Computing*, second edition, Cambrige University (1992).

[23] D. A. Steck, "Rubidium 87 D Line Data," unpublished, available on line, http://george.ph.utexas.edu/~dsteck/alkalidata/rubidium87numbers.pdf

[24] J. D. Weinstein, Ph.D. dissertation, *Magnetic Trapping of Atomic Chromium and Molecular Calcium Monohydride.*

[25] F. Schreck, et al., Phys. Rev. Lett. 87, 080403 (2001).

[26] Z. Hadzibabic, et al., Phys. Rev. Lett. 88, 160401 (2002).

[27] M. Gehm, "Properties of ^{6}Li," unpublished, available on line, http://www.phy.duke.edu/research/photon/qoptics/techdocs/pdf/PropertiesOfLi.pdf

[28] E. R. I. Abraham, W. I. McAlexander, J. M. Gerton, and R. G. Hulet, Phys. Rev. A, 55, R3299 (1997).

[29] J. M. Vogels, C. C. Tsai, R. S. Freeland, S. J. J. M. F. Kokkelmans, B. J. Verhaar, and D. J. Heinzen, Phys. Rev. A, 56, R1067 (1997).

[30] S. B. Weiss, M. Bhattacharya, and N. P. Bigelow, Phys. Rev. A, 68, 042708-1 (2003).

[31] G. Ferrari, M. Inguscio, W. Jastrzebski, G. Mondugno, and G. Roati, Phys. Rev. Letter, 89, 053202-1 (2002).

[32] G. Modugno, G. Ferrai, G. Roati, R. J. Brecha, A. Simoni, M. Inguscio, Science, 294, 1320 (2001).

[33] A. Mosk, S. Jochim, H. Moritz, T. Elssser, M. Weidemller, and R. Grimm, *Resonator-enhanced optical dipole trap for fermionic lithium atoms*, Opt. Lett. 26, 1837-1839 (2001).

[34] M. Greiner, I. Bloch, T. W. Hansch, and T. Esslinger, Phys. Rev. A 63, 031401 (2001).

[35] H. Lewandowski, D. Harber, D. Whitaker, and E. Cornell, J. Low Temp. Phys. 132, 309 (2003).

[36] A. Gozzini, F. Mango, J. H. Xu, G. Alzetta, F. Maccarone, and R. A. Bernheim, Nuovo Cimento D 15, 709 (1993).

[37] S. Aubin, M. Extavour, S. Myrskog, L. LeBlanc, J. Esteve, S. Singh, P. Scrutton, D. McKay, R. McKenzie, I. Leroux, A. Stummer, and J. H. Thywissen J. Low Temp. Phys. 140 377-396 (2005) F.

Appendix A

Temperature Monitoring Unit

As mentioned in Chapter 4.3, our vacuum system includes many different parts, each of them can sustain a specific maximum baking temperature (shown in table ...). To obtain the required vacuum environment for our ultra-cold collision experiment, we need to bake the assembled vacuum system as hot as possible but withoult locally exceeding the maximum temperature of a given part. Monitoring the baking temperature is realized by a Data Acquisition (DAQ) system which is packaged in a rack-mounted box with a 19" front panel, as shown in Fig. A.1.

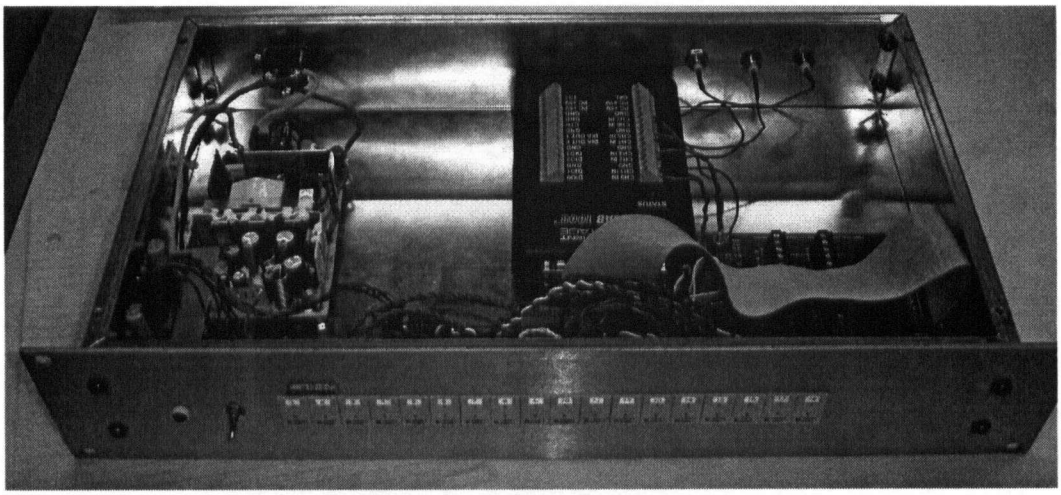


Figure A.1: Overview of the temperature monitoring box.

Required Parts and Functionalities

This temperature monitoring unit is composed of several parts: 16-channel temperature probing circuit, 19" front panel (with 18 k-type thermocouple connectors on it), k-type thermocouple probes, power supply and Minilab 1008. Ordering information about most of the parts in this box is shown in table A.1.

The thermocouple probes will be plugged into the connectors on the 19" front panel. By the connection between this 19" panel and the 16-channel temperature probing circuit, these thermocouples are linked to the related channels on the analog multiplexers. The temperature signal on each thermocouple is picked up by the probing circuit periodically and sent to the analog input on the DAQ device, Minilab 1008.

Part Number	Manufacturer	Price	Qty.	Description
19MJP1-18-K	Omega	\$61.45	1	19" front panel (with 18 thermocouple connectors)
N.A.	Measurement Computing	\$109.00	1	Minilab 1008
AD595	Analog Device	\$10.43	1	k-type thermocouple amplifier
ADG407	Analog Device	\$6.33	2	analog multiplexer
INA 128	Burr-Brown	\$3.05	1	op-amp

Table A.1: Ordering information for the essential parts used in the Temperature Monitoring Unit.

Finally these temperature data are sent to computer through the USB connection on the Minilab and further processed in LabView.

In the following paragraphs, details about the probing circuit and Minilab 1008 will be discussed.

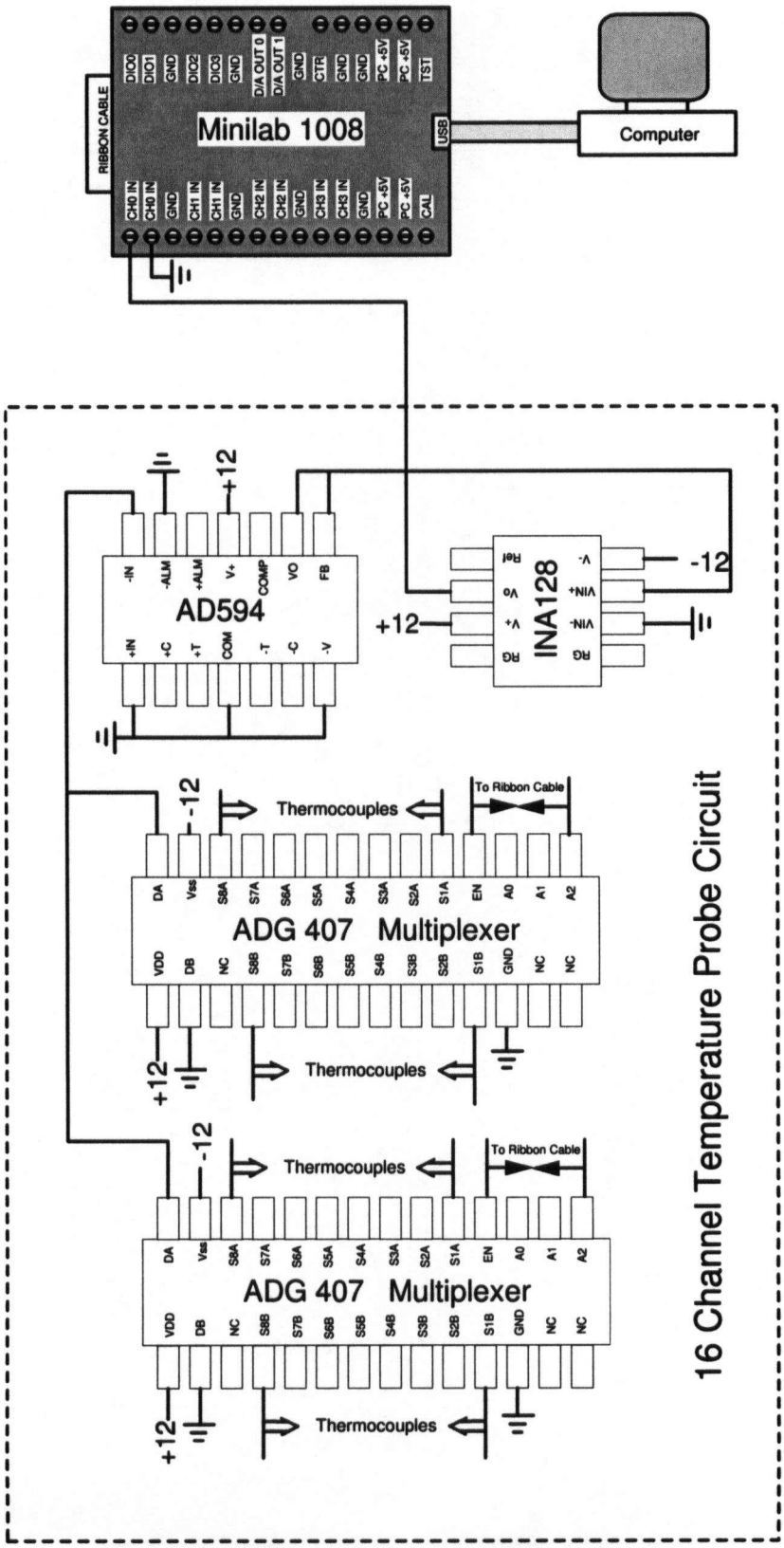
16-channel Temperature Probing Circuit

A2	A1	A0	EN	ON SWITCH PAIR
X	X	X	0	NONE
0	0	0	1	1
0	0	1	1	2
0	1	0	1	3
0	1	1	1	4
1	0	0	1	5
1	0	1	1	6
1	1	0	1	7
1	1	1	1	8

Table A.2: Truth table for analog multiplexer, ADG 407. EN is the chip enable pin, combining EN with the other three pins, A0, A1 and A2, determines which channel is switched on. On Switch Pair means SXA is connected to DA, SXB is connected to DB.

The 16-channel temperature probing circuit is designed to switch between different thermocouples and export the analog voltage signals detected by each thermocouple periodically. This circuit is designed by me and etched in our lab on a double-sided PCB (Printing Circuit Board). The circuit schematics and PCB layout are shown in Fig. A.2 and Fig. A.3, respectively.

As shown in Fig. A.2, several chips are combined together to realize the required functions for this circuit: 2 analog multiplexers (ADG407), 1 k-type thermocouple


Minilab 1008
RIBBON CABLE
USB
Computer
AD594
INA128
+12
-12
Thermocouples
To Ribbon Cable
ADG 407 Multiplexer
16 Channel Temperature Probe Circuit

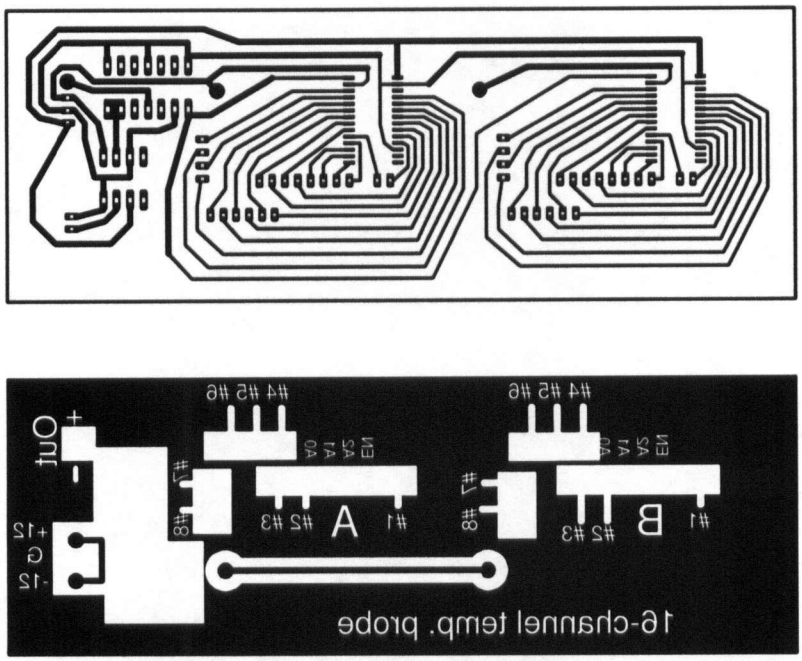


Figure A.3: PCB layout for 16 channel temperature probes circuit.

amplifier (AD595) and 1 op-amp (INA128). The analog multiplexer, ADG407, can support 8 channel thermocouples, each channel is controlled by a 3-digit code (A0, A1 and A3), table A.2 shows the logic between different codes and channel numbers. Each multiplexer can be distinguished by the EN pin. AD595 is a complete instrumentation amplifier and thermocouple cold junction compensator on a monolithic chip. It combines an ice point reference with a pre-calibrated amplifier to produce a high level ($10mV/$ °C) output directly from a k-type thermocouple signal. INA128, the op-amp, offers an amplification of the input signal, the gain of this amplification is set by connecting a single external resistor, R_G, between pin 1 and pin 8. Eqn A.1 shows the relation between R_G and the gain. In our circuit, R_G is not used, since we choose to use unit gain between input and output signals.

$$G = 1 + \frac{50k\Omega}{R_G} \tag{A.1}$$

Minilab 1008

The Minilab 1008 is a USB-based DAQ device with 8 channels of 12-bit analog input, two analog outputs and 28 Digital I/Os. It is used to periodically sample the thermocouple signals and send data into a computer through USB connection.

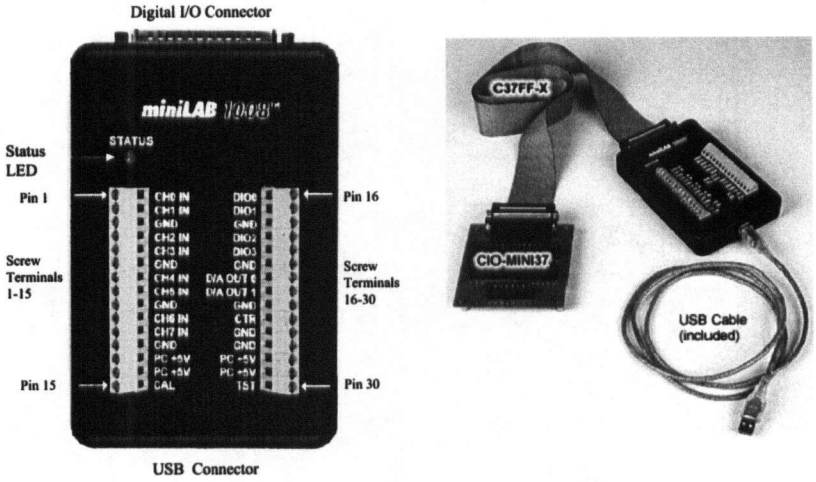


Figure A.4: (a): Minilab 1008, (b): general accessories. CIO-MINI37 is the extension board for external I/Os, it can be connected to Minilab through ribbon cable C37FF-X.

Fig. A.4 shows the picture of Minilab 1008 and all the required accessaries.

In our circuit, 5 digital I/Os are necessary to control all-16 channels on two analog multiplexers (ADG 407), they will be connected to EN1, EN2, A0, A1 and A2 separately. However, there are only 4 digital I/Os (DI0-DI3) on the top panel of the Minilab 1008, so we have to use extension board CIO-MINI37 to get the access to the other 24 digital I/Os.

There are two different kinds of modes for the analog inputs, 4-channel differential mode and 8-channel single-ended mode. The differential mode provides more ranges of measurement, from ±1.0V to ±20V. Since four analog channels are enough for our application and different ranges are necessary for measuring different signals, Minilab 1008 is set to be differential mode in our circuit. The 1st channel is used for measuring thermocouple signals, the other three are connected to BNC connectors on the back of package box (Fig. A.1) so that we can access to them easily later.

Two pieces of software are very important for using Minilab 1008, InstaCal and the LabView driver. InstaCal provides installation, calibration and test programs, mode setting can also be done using this software. The LabView driver gives us the chance to program Minilab 1008 in LabView, it also includes lots of useful sample VIs which can be utilized to simplify our work.

LabView Program and Temperature Monitoring

Since the Minilab 1008, the DAQ device, has plenty of LabView drivers and sample VIs, it is convenient to write the program in LabView. The resulting program has a very nice front panel (Fig. A.5) on which we can set different parameters and

monitor temperature of each channel very conveniently. This program scans all-

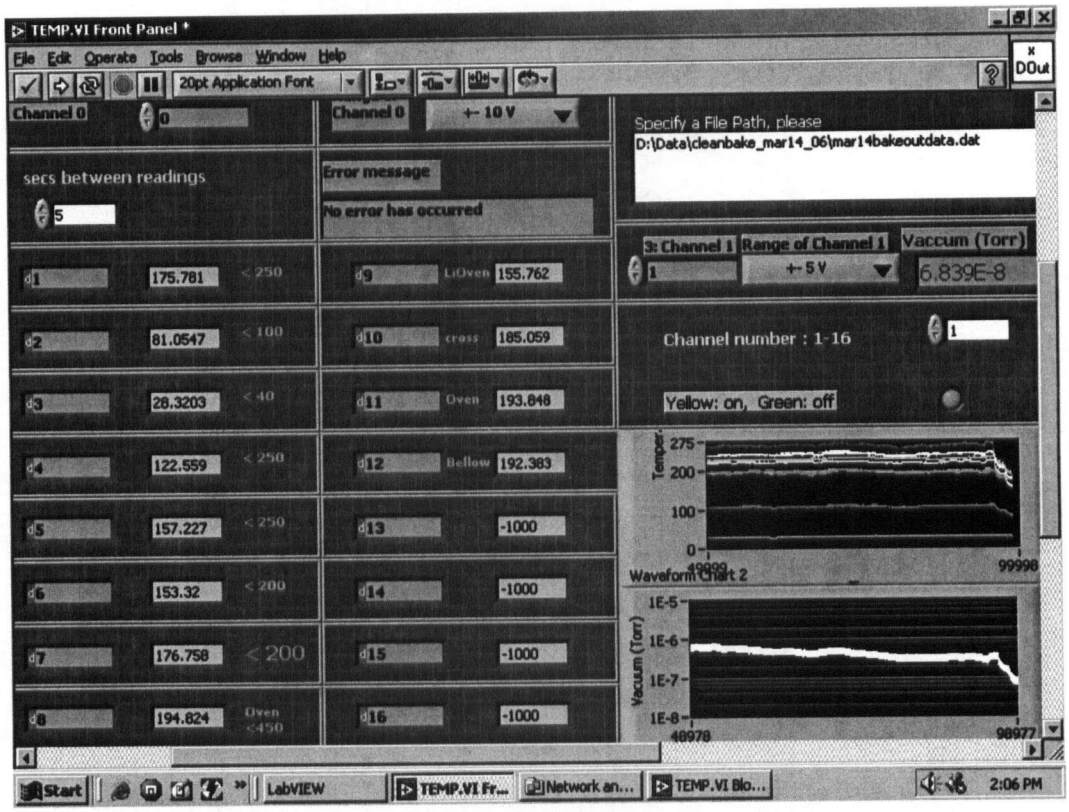


Figure A.5: Labview front panel for monitoring temperature and vacuum pressure during bakeout procedure.

16 channels periodically and converts the voltage signals into Celsius degrees, and the scanning period is determined by the parameter, "sec between readings". The obtained temperature data will be both displayed on the front panel and recorded into a file, the file name and storing directory can be set in the blank window "set a File Path".

On the front panel, there are two modes for displaying the temperature value, the numerical list and the time evolution waveform. On the numerical list, the integer number on the left side is the channel number, the real number is the corresponding temperature value; The maximum baking temperature on each channel is also indicated right after the temperature value so that we can make sure the baking temperature is in the safe range. On the time evolution waveform, all-16 temperature curves can be displayed at the same time, or you can turn on/off some channels by changing the round button above the waveform using the right channel number.

Below the temperature waveform, there's still one more graph for the vacuum pressure, which is determined by the signals from another channel on Minilab 1008. The x axes (evolution time) are the same for both waveforms, this synchronization

helps us inspect the changing of vacuum pressure with the baking temperature during bakeout.

Appendix B

Spot Welding of Alkali Dispenser With The Copper Conductor

To improve the connection between the dispenser and the copper conductor, and also to minimize the number of screws used in the vacuum chamber, we tried to use spot welding to connect the dispenser terminal with the copper conductor on the electrical feedthru.

Spot Welding Machine

The spot welding machine is located in John Hepburn's lab, it has a Miyachi Unitek Dual Pulse 125DP power supply, and a THP tweezer handpiece with ET0420 electrodes, the electrodes look like a taper with a 1/16" tip at the end, as shown in Fig. B.1.

Figure B.1: Outlook of the electrodes. ET0420 is made of molybdenum, ET0450 is made of copper alloy.

As programmed in dual pulse mode, the control will fire twice from a single actuation. The first pulse (Fig. B.2) is set at a lower energy level relative to the second pulse. It is specifically designed to properly seat the electrodes and prepare the parts by displacing surface contamination without significantly reducing the interface resistance between the parts. The second pulse (Fig. B.2), or "weld pulse" joins the base metals. The second pulse energy level is typically set three to four times that of the first. This important feature, pioneered by Miyachi Unitek, produces repeatable and acceptable results in difficult-to-weld situations.

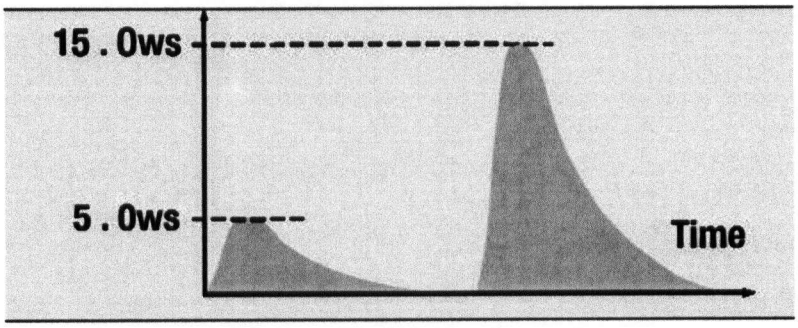


Figure B.2: Schematics of the dual pulse process of the Unitek 125DP spot welding machine . First Pulse seats the electrodes, second Pulse provides the welding energy, the unit "ws" means watt-seconds.

Procedure of Spot Welding

Before we actually did the spot welding, we were worrying about the compatibility of the dispenser and the conductor. It is said that spot welding between different materials is not easy to make. Our dispenser is made of nickel(78%) and chromium(22%), the conductor on the electrical feedthrough is made of copper (diameter: around 0.1"), they are not compatible. Here somebody may say that there are electrical feedthroughs with nickel conductors, this is true, however, due to the high current threshold (5-7A) of the dispensers and the low current limitation of the nickel electrical feedthroughs (8.2A for 0.05" diameter), and also due to the unpopularity of nickel electrical feedthroughs, the most likely electrical feedthrough we are going to order is the one with copper conductors.

When I tried to do the spot welding, the first practice supported the above argument, the dispenser and the conductor are not compatible, we failed to weld them together. At this time the dispenser was placed on the surface of the copper wire, the result was that the tip of the electrode just went through the dispenser terminal, and the dispenser was not welded onto the copper wire.

After the first practice, we also tried to wrap the dispenser terminal around the copper wire, still not working. Finally we planished the copper wire to be flat (Fig. B.3), like the dispenser terminal. This kind of treatment improved the welding, we can easily spot weld them together. We did 3-4 times spot welding between the dispenser terminal and the flat copper wire, as I remembered, the success rate is 100%.

Resistance Test

Although we successfully spot welded the dispenser and the conductor, there's still some unclarity about how good the connection is, whether this connection can take high current or not. Also the soldering tip always penetrates the dispenser terminal and prevents us from making a perfect connection between the copper wire and

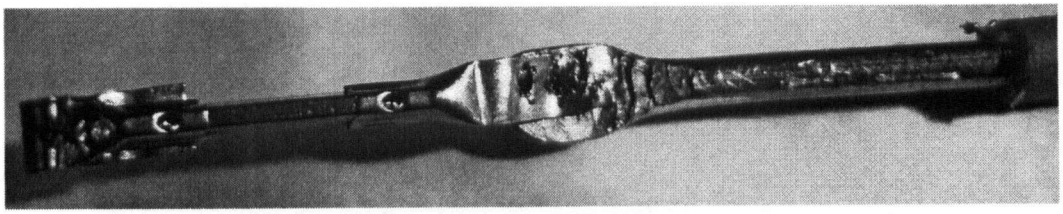


Figure B.3: Spot welding the dispenser terminal onto the pre-flattened copper wire.

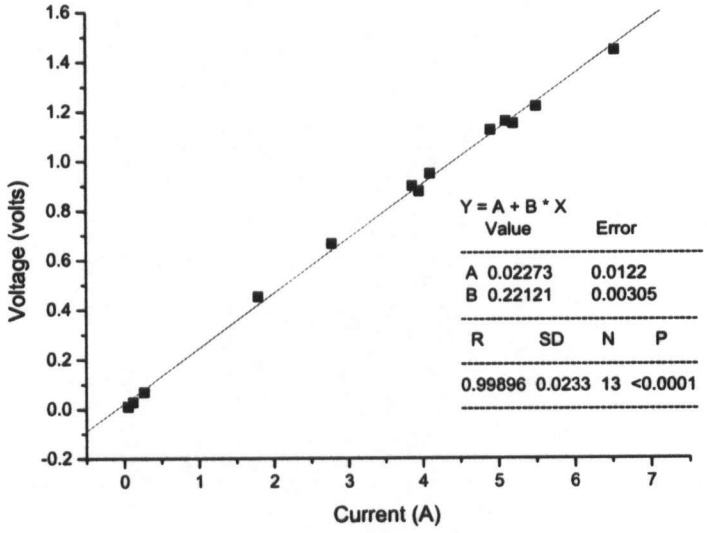


Figure B.4: Voltage vs current curve for the dispenser, which is welded to the copper wire by spot welding.

the dispenser terminal, as shown in Fig B.3. To clarify these doubts, I tried to measure the resistance of the dispenser by Voltage-Current method, the result is shown in Fig. B.4. It is quite clear that the V-I curve is linear during the heating process, and the resistance is 0.221±0.003Ohms, this is consistent with the direct resistance measurement using a precise multimeter Hewlett Packard 34401A, which is 0.138±0.010(dispenser)+0.076Ohms(copper wire)=0.214±0.010Ohms. Here the resistance of the dispenser is lower than the others (0.16∼0.18Ohms), this is because the resistance of the dispenser depends greatly on the length of the dispenser terminals and these dispenser terminals were damaged and shortened during the spot welding process; One terminal was even wrapped, as shown in Fig. B.3. The highest current I tried is 6.63A. After this measurement the connection didn't change at all, nothing weird happened.

Problems to Be Taken Care

One thing I need to mention here is the cleanness of the materials, we didn't take too much care about this during the spot welding process. However, both the dispenser and the copper wire are pretty clean, there's no rust on them.

Another thing is about the robustness of the connection. We found that the connection wouldn't be affected by the high current, even can take pulling force to a great point, however, it can't be bent too much. It seems that bending is the only method to easily destroy the connection. In the following paragraph we try to find out the solution to this problem.

Possible Methods to Improve The Spot Welding Connection

After checking the datasheet of the spot welding machine and consulting with the technical support in Miyachi Unitek Corporation, I find that the following solutions to improve the spot welding connection:

- Get the 88A/EZ weld head, which can make a perfect connection between the copper wire and the dispenser terminal, as shown in Fig. B.5.
- Machining the tip so that it is flatter with the part we want to spot weld, also radiusing the faces of the electrode to prevent catching the edges and creating hot spots.
- Get the ET0420 (besides the copper wire) and ET0450 (besides the nickel plate) electrodes. Try increasing the force on the hand piece by tightening the flat head screw on the handle. Make sure that the electrode which is in contact with the thicker copper flat wire is connected to the positive output terminal of the power supply.

The first method will cost us US$3,140 to buy the 88A/EZ weld head, and also the new 250DP power supply. The second one has been tried and didn't make great improvement.

The last method just requires to get another piece of electrode, ET0450, which only cost us less than US$100/pair. However, since we already tried using the ET0420 electrodes pair, it is not clear that how much improvements we can get by using different combination of electrodes. Anyway, this method still deserves to try, and we also ordered the new electrodes, as shown in Fig. B.1.

The Ultimate Method to Spot Weld The Copper Conductor With The Dispenser Terminal

The methods we discussed above are either not applicable, or can't make great improvements. Luckily, after numerous practice, We found the ultimate method to perfectly spot weld the copper conductor with the dispenser terminal.

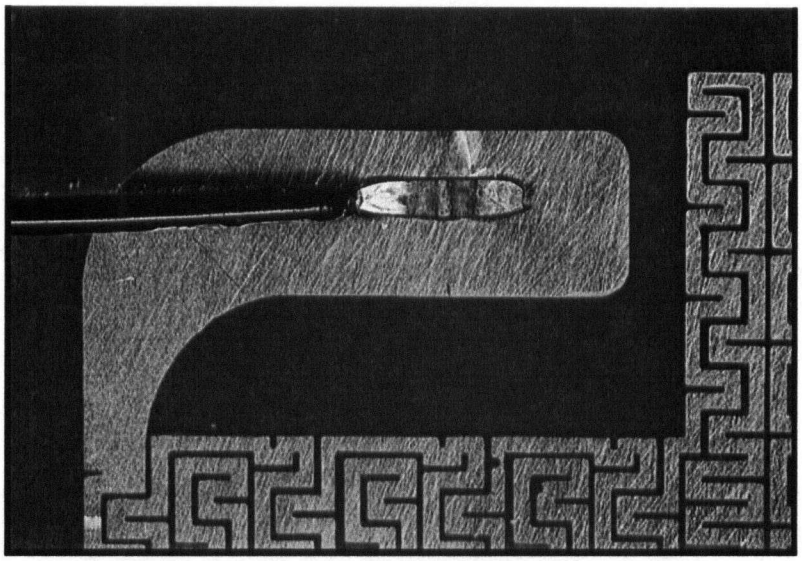


Figure B.5: Spot welding the pre-flattened copper wire to nickel heating element using 250DP and 88A/EZ weld head.

Due to the high threshold current requirement of the dispenser(5.3A~7.3A), and the availability of the suitable electrical feedthroughs, we ordered the one with 4 copper conductors (1/4" diameter), 2.75" flange size and can take 150A current at maximum.

Then we tried to spot weld the dispenser terminal onto with the 1/4" diameter copper conductor. The result is a little bit disappointing, we can't spot weld the dispenser directly onto the thick flat copper bar (flattened by milling) at all. However, the dispenser is stuck onto the pre-flattened copper wire (0.1" diameter, in Fig. B.3), when we sandwiched the dispenser terminal between the pre-flattened copper wire and the cylinder surface of the 1/4" diameter copper conductor. The reason we guess is that, the narrow contact between the dispenser terminal and the cylinder surface on the 1/4" diameter copper conductor serves as a current focus for the dispenser and the pre-flattened copper wire, this current focusing increases the local current intensity and then improves the spot welding between the dispenser and the 0.1" pre-flattened copper wire.

The lessons we learned from this practice are:

- To protect the dispenser terminal from being penetrated by the tips, we should sandwich the terminal with another piece of metal.
- We can't spot weld the dispenser terminal onto a big chunk of copper, like the 1/4" diameter copper conductor, especially not the thick copper block with a flat surface.
- Use the cylinder surface on the 1/4" diameter copper conductor as the base for the spot welding. In other words, use this cylinder surface as another part

to sandwich the dispenser terminal. The reason is discussed above, the less contact improves the current focusing.

- Following all the lessons above can't guarantee spot welding the dispenser with the copper conductor very efficiently. we need to try several times before making a good welding.

Summary

Although there is incompatibility between dispenser terminal and the copper conductor, we can still spot weld them together by planishing the copper wire. To avoid the penetration of the electrode tip through the dispenser terminal, we can sandwich the dispenser terminal between the pre-flattened copper conductor and a 1/4" diameter copper bar. Using the cylinder surface on the copper bar as a current focus can efficiently increase the local current intensity at the sandwich contact and therefore greatly improve the spot welding connection.

We also checked the connection by running high current through them and plotted the Voltage-Current curve, the linearity of the V-I curve clearly shows that the connection is good enough to satisfy our requirement in the future experiment.

Appendix C

Program Code for Simulating Zimmerman's Sympathetic Cooling

```
!-------------------------------------------------------------------------------
module parameters
  implicit none
  !Here define number of atoms, species and F states
  integer, parameter :: nat=9766, ntype=2 !"nat" should be the sum of all species.
  real*8,  parameter :: pi=3.1415926d0, uB=9.274d-24
  real*8   Fmax(1:ntype),    Fmin(1:ntype)   !Range of F state
  real*8   nivInit(1:ntype) !Initial spin state
  integer  iat, iat2, atype(1:nat), atypeaux(1:nat) !atype(iat):species of atom
  real*8   B0,Bprime,Bsec,mu(1:ntype),masse,mass(1:ntype), mi, m2
  integer  macroloc
  real*8   tempini(1:ntype)  , nat0(1:ntype)
  real*8   rabi(1:ntype), gFtemp(1:nat)
  real*8,  dimension(:,:), allocatable :: sigma_in, sigma_e, sl
  real*8   hyperfine(1:ntype) !hyperfine frequency splitting of the ground level.
  !-----------------------------------------------------------------------------
  type atom
     real*8 F
     real*8  mF
  end type atom
  type(atom) a(1:nat)
  type(atom) nivaux(1:nat) ! temperal storerage for a(1:nat)
end module parameters
!-------------------------------------------------------------------------------
real*8 FUNCTION gF(type,F)
  use parameters
  integer type
  real*8 F, gJ(1:ntype), gI(1:ntype), I(1:ntype), J(1:ntype)
  gJ = (/2.00233113d0,2.002301d0/)
  gI = (/-0.0009951414d0,-0.000447654d0/)
  I  = (/1.5d0,1.d0/)
  J  = (/0.5d0,0.5d0/)
  !Finding G factor for F level.
  gF = gJ(type)*(F*(F+1)-I(type)*(I(type)+1)+J(type)*(J(type)+1))/(2*F*(F+1)) &
       + gI(type)*(F*(F+1)+I(type)*(I(type)+1)-J(type)*(J(type)+1))/(2*F*(F+1))
  return
END FUNCTION gF
!-------------------------------------------------------------------------------
       program ev
!-------------------------------------------------------------------------------
       use parameters
       integer, parameter :: nmaxx=30,nmaxy=360,nmaxz=nmaxx,ndistrib=21
       real*8 pos(1:nat,1:3),vel(1:nat,1:3),kB,hplanck,hbar,dB
       real*8 posaux(1:nat,1:3),velaux(1:nat,1:3),temperature(1:ntype)
       real*8 alpha(1:3,0:3),beta(1:3,0:3)
       real*8 posAverr(1:ntype,1:3),posdevv(1:ntype,1:3),posdev(1:3), posAver(1:3)
       real*8 velAverr(1:ntype,1:3),veldevv(1:ntype,1:3),veldev(1:3)
       real*8 rayon0(1:ntype,1:3),velesse0(1:ntype,1:3),y(1:6), vquadd(1:ntype), vquad
       real*8 frequency(1:ntype,1:3),omega(1:ntype,1:3)
       real*8 vcdm(1:3),vrel(1:3),phi,stheta,ctheta,dx,dt,t,tmes
       real*8 Bevap_i(1:10),Bevap_f(1:10),duree(1:10),Bevap,dep_g(1:ntype,1:3,0:4)
       logical indic(1:nat),resonance
       integer table(-nmaxx:nmaxx,-nmaxy:nmaxy,-nmaxz:nmaxz)
       integer macro,macro0
       integer natrest, nrest(1:ntype), i ,j ! nrest:number of atoms in each species
```

```
integer mmm(1:ntype) !number of all spin states of each species
real*8  probacollmoy,probacoll, gammac1,gammac2,gammac,bound
real*8  sigma , sigmaloc,sigmaloc1,sigmaloc2
real*8  density(1:ntype),vbar(1:ntype),densmoy(1:ntype),phase(1:ntype)
real*8  temperatureAll,densityAll
integer Nneg,N0,N1,N2,N3
!---------------------------------------------------------------------
!parameters of different atoms
mass(1) = 1.44316060d-25; mass(2) = 5.*9.9883414d-27 ! 1:Rb87, 2:Li6
masse   = sum(mass)/dble(ntype) ! Average mass of all kinds of atoms
rabi(1) = 2.d0*pi*100.0d3 ; rabi(2) = 2.d0*pi*100.0d3
hyperfine = (/6.83468261090429d9,2.282d8/) !unit: Hz.
Fmax    = (/2.d0,1.5d0/) ; Fmin    = (/1.d0,0.5d0/)
nivInit = (/2.0d0,1.5d0/) ! Initial F state: Rb87:F=2, Li6:F=3/2
!---------------------------------------------------------------------
!initialization of matrixes for elastic and inelastic cross section
mmm(:) = (Fmax(:)-Fmin(:)+1)*(2*Fmax(:)+2*Fmin(:)+2)/2.d0
allocate(sigma_e(sum(mmm),sum(mmm)),sigma_in(sum(mmm),sum(mmm)))
allocate(sl(sum(mmm),sum(mmm)))
sigma_in(:,:)= 1.d0   * 5.291772083d-11
do i = 1, mmm(1) ! Rb-Rb
   do j = 1, mmm(1)
      sl(i,j) = 102.d0 * 5.291772083d-11
      sigma_e(i,j) = 8.d0 * pi * sl(i,j)**2
   end do
end do
do i = mmm(1)+1, mmm(1)+mmm(2) ! Li-Rb
   do j = 1, mmm(1)
      sl(i,j) = 20.d0 * 5.291772083d-11
      sigma_e(i,j) = 4.d0 * pi * sl(i,j)**2
   end do
end do
do i = 1, mmm(1) !Rb-Li
   do j = mmm(1)+1, mmm(1)+mmm(2)
      sl(i,j)=20.d0 * 5.291772083d-11
       sigma_e(i,j) = 4.d0 * pi * sl(i,j)**2
   end do
end do
do i = mmm(1)+1 , mmm(1)+mmm(2) !Li-Li
   do j = mmm(1)+1 , mmm(1)+mmm(2)
      sl(i,j) = 2160.d0 * 5.291772083d-11
       sigma_e(i,j) = 4.d0 * pi * sl(i,j)**2
   end do
end do
!sigma_e(:,:)= 8.d0 * pi * sl(:,:)**2
!---------------------------------------------------------------------
!physical constants
kB=1.38d-23             ! J/K
hplanck=6.62d-34        ! J*S
hbar=hplanck/(2.*pi)
g=9.81d0
!---------------------------------------------------------------------
open(unit=1,file='ev.in')
open(unit=11, file='parameter.dat')
open(unit=111,file='position.dat')
open(unit=2621,file='vquadd.dat')
read(1,*) tmax,nmes,tau,resonance
read(1,*) Bprime,B0,Bsec
read(1,*) macro0,tempini,nat0 ! tempini: initial temperature()
read(1,*) nsegment
nsegment=min(nsegment,10)
do isegment=1,nsegment
   read(1,*) Bevap_i(isegment),Bevap_f(isegment),duree(isegment)
enddo
close(1)
natini = sum(nat0) ! finding total initial number of atoms
!
!Initialization of atom type, the corresponding mass for
!that atom will be: mass(atype(iat))
```

```
ntemp = 0
do i = 1 , ntype
   do iat = 1 , nat0(i)
      atype(iat+ntemp) = i
   end do
   ntemp = ntemp + nat0(i)
end do
open(unit=234,file='atype.dat')
do i = 1, natini
   write(234,*) i, atype(i)
end do
close(234)
! mu = m_F * gF * U_B
do i = 1 , ntype
   mu(i)=uB * abs(gF(i,nivInit(i))) * abs(nivInit(i))
end do
frequency(:,1)=sqrt(mu(:)/(B0*mass(:)))*bprime/(2.*pi)
frequency(:,2)=sqrt(mu(:)*Bsec/mass(:))/(2.*pi)
frequency(:,3)=frequency(:,1)
write(11,*)'mass       ', mass
write(11,*)'frequency_A', frequency(1,:)
write(11,*)'frequency_B', frequency(2,:)
write(11,*)'B0,B',B"  ' , B0, Bprime, Bsec
write(11,*)'temperature', tempini
write(11,*)'initial N  ', nat0
write(11,*)'initial F  ', nivInit
!finding boundary of the atom clound.
bound = 10.d0*sqrt(kB*maxval(tempini)/minval(mass))/(2*pi*minval(frequency))
!write(*,*) bound
!
do idim = 1 , 3
   omega(:,idim)=2.*pi*frequency(:,idim)
   do i=0,4
      dep_g(:,idim,i)=0.d0
   end do
end do
do i=1,4
   dep_g(:,3,i)=-(3./i)*g/omega(:,3)**2
end do
!
open(unit=2,file='out.dat')
open(unit=21,file='out1.dat')
open(unit=22,file='out2.dat')
open(unit=3,file='densite.dat')
open(unit=4,file='natom.dat')
open(unit=5,file='spin.dat')
write(5,*)  'time    Nneg      N0      N1      N2      N3'
write(2,*) 'tau=',tau
write(2,*) 'B0=',B0,'  Bprime=',Bprime,'  Bseconde=',Bsec
write(2,*) 'calcul fait avec nat =',nat,' macroatomes'
write(21,*) 'tau=',tau
write(21,*) 'B0=',B0,'  Bprime=',Bprime,'  Bseconde=',Bsec
write(21,*) 'calcul fait avec nat =',nat,' macroatomes'
write(22,*) 'tau=',tau
write(22,*) 'B0=',B0,'  Bprime=',Bprime,'  Bseconde=',Bsec
write(22,*) 'calcul fait avec nat =',nat,' macroatomes'
do isegment=1,nsegment
   write(2,*) Bevap_i(isegment),Bevap_f(isegment),duree(isegment)
   write(21,*) Bevap_i(isegment),Bevap_f(isegment),duree(isegment)
   write(22,*) Bevap_i(isegment),Bevap_f(isegment),duree(isegment)
end do
write(2,*) '-----------------------------------------------'
write(21,*) '-----------------------------------------------'
write(22,*) '-----------------------------------------------'
!
write(*,*) 'tau=',tau
write(*,*) 'B0=',B0,'  Bprime=',Bprime,'  Bseconde=',Bsec
write(*,*) 'calcul fait avec nat =',nat,' macroatomes'
do isegment=1,nsegment
```

```
   write(*,*) Bevap_i(isegment),Bevap_f(isegment),duree(isegment)
end do
write(*,*) '-----------------------------------------------'
!
write(*,*) '   t xrms  yrms zrms   temp   N_atomes taux_c', &
     ' densit ', ' phase   Pcol Pocc Bevap'
write(2,*) '   t xrms  yrms zrms   temp   N_atomes taux_c', &
     ' densit ', ' phase   Pcol Pocc Bevap'
write(21,*) '   t xrms  yrms zrms   temp   N_atomes taux_c', &
     ' densit ', ' phase   Pcol Pocc Bevap'
write(22,*) '   t xrms  yrms zrms   temp   N_atomes taux_c', &
     ' densit ', ' phase   Pcol Pocc Bevap'
write(4,*)' t    N_atomes    N_Rb      N_Li'
macro=macro0
!
!initialization of spin states
do iat=1,natini
   ! initialization of F state
   a(iat)%F  = nivInit(atype(iat))
   ! initialization of spin state depending on gF
   if (gF(atype(iat),a(iat)%F) .gt. 0.0d0) then
      a(iat)%mF = nivInit(atype(iat))
   else
      a(iat)%mF = nivInit(atype(iat)) *(-1.d0)
   endif
end do
!initialization of positions and velocities
do i = 1 , ntype
   do idim=1,3
      rayon0(i,idim)=sqrt(kB*tempini(i)/mass(i))/omega(i,idim)
      velesse0(i,idim)=sqrt(kB*tempini(i)/mass(i))
   end do
end do
do iat=1,natini
   indic(iat)=.true.
   do idim=1,3
      alea1=1.0-ran3(idum)
      alea2=1.0-ran3(idum)
  pos(iat,idim)=dep_g(atype(iat),idim,3)+ rayon0(atype(iat),idim)* &
           sqrt(-2.*log(alea1))*cos(2*pi*alea2)
      vel(iat,idim)=velesse0(atype(iat),idim)* &
           sqrt(-2.*log(alea1))*sin(2*pi*alea2)
   end do
end do
! output the initial positions of all atoms
do iat=1,natini
   if (atype(iat) .eq. 1) write(111,*)'Rb', pos(iat,:)
   if (atype(iat) .eq. 2) write(111,*)'Li', pos(iat,:)
end do
do iat=natini+1,nat
   indic(iat)=.false.
end do
!
! beginning of the loop over time
ncoll=0
t=0.
tmes=0. !interval of time between two measurements
dt=0.   ! dt est  ajuste a chaque pas pour un bon taux de coll. !
do while (t.lt.tmax)
   ! re-index of left atoms and Duplication around z axis
   nrest(:) = 0 ; natrest = 0
   !Finding number of each kind of atom.
   do iat=1,nat
      if (indic(iat)) then
         nrest(atype(iat)) = nrest(atype(iat)) + 1
         natrest = natrest + 1
      end if
   end do
   !if ((natrest.le.nat/2).and.(macro.eq.1)) goto 777
```

```
  if ((natrest.le.nat/2).and.(macro.gt.1)) then
     ! manufacture of the auxiliary file
     iataux=1
     do iat=1,nat
        if (indic(iat)) then
           do idim=1,3
              posaux(iataux,idim)=pos(iat,idim)
              velaux(iataux,idim)=vel(iat,idim)
           end do
           !duplication of spin and F, and species of atoms
           nivaux(iataux)  = a(iat)
           atypeaux(iataux)= atype(iat)
           iataux=iataux+1
        endif
     end do
     !transfer of the auxiliary file into the main file
     do iat=1,natrest
        do idim=1,3
           pos(iat,idim)=posaux(iat,idim)
           vel(iat,idim)=velaux(iat,idim)
        end do
        a(iat)     =  nivaux(iat)
        atype(iat) =  atypeaux(iat)
     end do
     !duplication with symmetry compared to axis Z
     do iat=natrest+1,2*natrest
do idim=1,2
           pos(iat,idim) = -posaux(iat-natrest,idim)
           vel(iat,idim) = -velaux(iat-natrest,idim)
        end do
        pos(iat,3) = posaux(iat-natrest,3)
        vel(iat,3) = velaux(iat-natrest,3)
        a(iat)     = a(iat-natrest)
        atype(iat) = atype(iat-natrest)
     end do
     ! updating indic()
     do iat=1,2*natrest
        indic(iat)=.true.
     end do
     do iat=2*natrest+1,nat
        indic(iat)=.false.
     end do
     ! reduction in the atomic size
     macro=macro/2
     natrest=natrest*2
     nrest(:)= nrest(:)*2
  endif                  ! end of duplication
  !
  ! choosing of the time step
  !Average position and standard deviation for each atom species.
  posAverr(:,:)=0 ; posdevv(:,:)=0
  !Average velocity and standard deviation for each atom species.
  velAverr(:,:)=0 ; veldevv(:,:)=0
  !Number of atoms in each spin state.
  N0=0; N1=0; N2=0; N3=0; Nneg=0
  !
  do iat=1,nat
     if (indic(iat)) then
        posAverr(atype(iat),:)= posAverr(atype(iat),:)+pos(iat,:)
        posdevv(atype(iat),:) = posdevv(atype(iat),:) +pos(iat,:)**2
        velAverr(atype(iat),:)= velAverr(atype(iat),:)+vel(iat,:)
        veldevv(atype(iat),:) = veldevv(atype(iat),:) +vel(iat,:)**2
        if (a(iat)%mF.eq.-3.d0) N3=N3+1
        if (a(iat)%mF.eq.-2.d0) N2=N2+1
        if (a(iat)%mF.eq.-1.d0) N1=N1+1
        if (a(iat)%mF.eq.0.d0) N0=N0+1
        if (a(iat)%mF.gt.0.d0) Nneg=Nneg+1
     endif
  enddo
```

```
        vquadd(:) = 0.d0; vquad = 0.d0;
        posdev(:) = 0.d0; veldev(:) = 0.d0; posAver(:) = 0.d0
        do idim=1,3
           posAverr(:,idim) = posAverr(:,idim)/nrest(:)
           posdevv(:,idim)  = posdevv(:,idim) /nrest(:)
           posdevv(:,idim)  = sqrt(posdevv(:,idim)-posAverr(:,idim)**2)
           velAverr(:,idim) = velAverr(:,idim)/nrest(:)
           veldevv(:,idim)  = veldevv(:,idim)/nrest(:)
           veldevv(:,idim)  = sqrt(veldevv(:,idim)-velAverr(:,idim)**2)
           vquadd(:)=vquadd(:)+veldevv(:,idim)**2
        end do
        do i = 1, ntype
           vquadd(i)=sqrt(vquadd(i))
        end do
        do i=1,ntype
           vquad     = vquadd(i)     + vquad
           posdev(:) = posdevv(i,:)  + posdev(:)
           veldev(:) = veldevv(i,:)  + veldev(:)
           posAver(:)= posAverr(i,:) + posAver(:)
        end do
        posdev(:) = posdev(:)  / dble(ntype) ! dble() : convert to real*8
        veldev(:) = veldev(:)  / dble(ntype)
        posAver(:)= posAver(:) / dble(ntype)
        vquad = vquad / dble(ntype)
        vbar(:)=4./sqrt(pi) * vquadd(:)/sqrt(3.)
        densmoy(:)=(nrest(:)*macro)/(2.*sqrt(2.)*(2.*pi)*sqrt(2.*pi))/ &
             (posdevv(:,1)*posdevv(:,2)*posdevv(:,3))
        gammac1=minval(sigma_e)*minval(densmoy)*minval(vbar)
        gammac2=128*(minval(densmoy)/maxval(vbar))*(hbar/maxval(mass))**2
        gammac=max(gammac1,gammac2)
        dt=min(0.005d0/gammac,1.d-4)
        !
        ! Mesurement
      if ((t.eq.0.).or. &
           (int(t*nmes/tmax).ne.int((t+dt)*nmes/tmax))) then
           write(2621,*) t, vquadd(1),vquadd(2)
           temperature(:)=mass(:)*vquadd(:)**2/(3.*kB)
           !calculating overall temperature-----------------------------------------
           temperatureAll = 0.0d0
           do i = 1 , ntype
              temperatureAll = temperatureAll + nrest(i)*mass(i)*vquadd(i)**2
           end do
           temperatureAll = temperatureAll/(3.d0*kB*natrest)
           !-------------------------------------------------------------------------
           density(:)=densmoy(:)*2.*sqrt(2.)  ! densite centrale
           densite = sum(density) / dble(ntype)
           if (tmes.gt.0.) tauxcoll=2.*ncoll/(natrest*tmes)
           phase(:)=density(:)*(hplanck/(sqrt(2*pi)*mass(:)))**3 *2.d0*sqrt(2.d0)/ &
                  (veldevv(:,1)*veldevv(:,2)*veldevv(:,3))
           !
           if (ncollpos.ne.0) probacollmoy=probacollmoy/ncollpos
           ! output of the whole system
           write(2,999) t,1E6*posdev(1),1E6*posdev(2),1E6*posdev(3),   &
                1E6*temperatureAll,natrest*macro,tauxcoll,densite*1E-16,  &
                sum(phase)/dble(ntype), probacollmoy,densite*(dx**3)/macro,Bevap*1E4
           ! output of different species separately
           write(21,999) t,1E6*posdevv(1,1),1E6*posdevv(1,2),1E6*posdevv(1,3),   &
                1E6*temperature(1),nrest(1)*macro,tauxcoll,density(1)*1E-16,  &
                phase(1), probacollmoy,density(2)*(dx**3)/macro,Bevap*1E4
           write(22,999) t,1E6*posdevv(2,1),1E6*posdevv(2,2),1E6*posdevv(2,3),   &
                1E6*temperature(2),nrest(2)*macro,tauxcoll,density(2)*1E-16,  &
                phase(2), probacollmoy,density(2)*(dx**3)/macro,Bevap*1E4
           write(*,999) t,1E6*posdev(1),1E6*posdev(2),1E6*posdev(3),   &
                1E6*sum(temperature)/dble(ntype), natrest*macro,tauxcoll,densite*1E-16, &
                sum(phase)/dble(ntype),probacollmoy,densite*(dx**3)/macro,Bevap*1E4
999        format(f5.2,' ',f4.0,' ',f5.0,' ',f4.0,' ',e8.3,'
',i9,' ', &
                f5.1,' ',f6.1,' ',e8.2,' ',f4.2,' ',f4.2,f6.2)
           write(3,*) t,densite*1E-16
```

```
            write(4,9) t,natrest*macro, nrest(1)*macro, nrest(2)*macro
            write(5,99) t, Nneg, N0, N1, N2, N3
9           format(f5.2,' ',i9,' ',i9,' ',i9) 99
format(f5.2,'   ',i5,'   ',i5, '   ',i5,'   ',i5,'   ',i5,'   ')
            ncoll=0
            ncollpos=0
            probacollmoy=0.
            tmes=0.
         endif ! End of measurement
         !
         ! collisions
      dx=4.*max(posdev(1)/nmaxx,posdev(2)/nmaxy,posdev(3)/nmaxz)
      do iat=1,nat
            if (indic(iat)) then
               ix=nint((pos(iat,1)-posAver(1))/dx)
               iy=nint((pos(iat,2)-posAver(2))/dx)
               iz=nint((pos(iat,3)-posAver(3))/dx)
               if ((abs(ix).le.nmaxx).and.(abs(iy).le.nmaxy).and.(abs(iz).le.nmaxz)) then
                  ! calculate the number of atoms in the grid
                  if (table(ix,iy,iz).eq.0) then  !There is not already an atom present
                     table(ix,iy,iz)=iat
                  else                ! the atom is already present
                     iat2=table(ix,iy,iz)
                     m1=mass(atype(iat)) ; m2=mass(atype(iat2))
                     vcoll = 0.d0
                     do idim=1,3
                        vcdm(idim)=(m1*vel(iat,idim)+m2*vel(iat2,idim)) / (m1+m2)
                        vrel(idim)=vel(iat,idim)-vel(iat2,idim)
                        vcoll=vcoll+vrel(idim)**2
                     end do
                     vcoll=sqrt(vcoll)
                     !find the scattering length between iat and iat2 fist.
                     !a0 = G_elastic(iat,iat2), temporarily not considered here.
                     if (atype(iat).eq.1 .and. atype(iat2).eq.1) then
                        a0 = 102.d0 * 5.291772083d-11
                        sigma = 8.d0 * pi * a0 **2
                     else if (atype(iat).eq.2 .and. atype(iat2).eq.2) then
                        a0 = 0.d0 * 5.291772083d-11 ! no collision between Li6-Li6
                      sigma = 4.d0 * pi * a0 **2
                     else
                        a0 = 20.d0 * 5.291772083d-11
                        sigma = 8.d0 * pi * a0 **2
                     end if
                     !elastic collision
                     !first found out the elastic collision cross section between
                     !iat and iat2, G_elastic(iat,iat2) gives the scattering length.
                     sigmaloc1=sigma/(1.d0+(m1*vcoll*a0/(2.*hbar))**2)
                     sigmaloc2=sigma/(1.d0+(m2*vcoll*a0/(2.*hbar))**2)
                     sigmaloc =sqrt(sigmaloc1*sigmaloc2)
                     !
                     probacoll=macro*sigmaloc*vcoll*dt/(dx**3)
                     if (a(iat)%mF.ne.a(iat2)%mF) probacoll = probacoll/2.d0
                     probacollmoy = probacollmoy + probacoll
                     ncollpos = ncollpos + 1
         !if(abs(probacoll).gt.1) write(*,*) dt, "probacoll=",probacoll, " is > 1"
                     if (probacoll.gt.ran3(idum)) then ! elastic collision
                        ncoll = ncoll + 1
                        !write(*,*) ncoll
                        ctheta= 2.*ran3(idum)-1.
                        stheta=sqrt(1.-ctheta**2)
                        phi=2.*pi*ran3(idum)
                        vel(iat ,1) = vcdm(1) + m2/(m1+m2)*vcoll*stheta*cos(phi)
                        vel(iat2,1) = vcdm(1) - m1/(m1+m2)*vcoll*stheta*cos(phi)
                        vel(iat ,2) = vcdm(2) + m2/(m1+m2)*vcoll*stheta*sin(phi)
                        vel(iat2,2) = vcdm(2) - m1/(m1+m2)*vcoll*stheta*sin(phi)
                        vel(iat ,3) = vcdm(3) + m2/(m1+m2)*vcoll*ctheta
                        vel(iat2,3) = vcdm(3) - m1/(m1+m2)*vcoll*ctheta
                        table(ix,iy,iz) = 0
                     endif
```

```
                endif
              endif
            endif
          end do
          do iat=1,nat
            ix=nint((pos(iat,1)-posAver(1))/dx)
            iy=nint((pos(iat,2)-posAver(2))/dx)
            iz=nint((pos(iat,3)-posAver(3))/dx)
    if ((abs(ix).le.nmaxx).and.(abs(iy).le.nmaxy).and.(abs(iz).le.nmaxz)) table(ix,iy,iz)=0.
          end do
        !
        ! move the atoms using Runge Kutta method and calculate any losses.
        do iat=1,nat
          !if (gF(atype(iat),a(iat)%F)*a(iat)%mF .lt. 0.d0) indic(iat) = .false.
          if (indic(iat).and.(ran3(idum)*tau.lt.dt)) indic(iat)=.false. !Backgroud loss
          if (indic(iat)) then
          !finding the g factor for atom "iat" first, to save time for the time evolution.
            gFtemp(iat) = gF(atype(iat),a(iat)%F)
            do idim=1,3
              y(idim)=pos(iat,idim)
              y(idim+3)=vel(iat,idim)
            enddo
            call rk4(t,y,dt)
            do idim=1,3
              pos(iat,idim)=y(idim)
              vel(iat,idim)=y(idim+3)
            enddo
          endif
          !get rid of Li atoms outside the boundary
          distance = sqrt(pos(iat,1)**2 + pos(iat,2)**2 + pos(iat,3)**2)
          if (distance .gt. bound .and. atype(iat).eq.2) indic(iat) = .false.
        end do
        !
        !rf spin flipping and evaporative cooling
        isegment=1
        trampe=duree(isegment)
        do while ((t.gt.trampe).and.(isegment.lt.nsegment))
          isegment=isegment+1
          trampe=trampe+duree(isegment)
        end do
      Bevap=Bevap_f(isegment)+(Bevap_i(isegment)-Bevap_f(isegment))*(trampe-t)/duree(isegment)
        !Bevap=Bevap_f(isegment)+(Bevap_i(isegment)-Bevap_f(isegment))*exp(-t/10.d0)
        do iat=1,nat
          if (indic(iat) .and. atype(iat).eq.1) then
            Bloc=sqrt((Bprime*pos(iat,1)-0.5d0*Bsec*pos(iat,1)*pos(iat,2))**2+&
              (B0+0.5d0*Bsec*pos(iat,2)**2-0.25d0*Bsec*(pos(iat,1)**2+pos(iat,3)**2))**2+&
                (pos(iat,3)*Bprime+0.5d0*pos(iat,2)*pos(iat,3)*Bsec)**2)
            !dB = abs( hbar * rabi(atype(iat)) / (a(iat)%mF*gF(atype(iat),a(iat)%F)*uB) )
            !if((abs(Bloc-Bevap).lt.dB).and.(ran3(idum).lt.(rabi(atype(iat))*dt)**2)) then
            if (Bloc .gt. Bevap) then
          !if(a(iat)%mF*gF(atype(iat),a(iat)%F)/abs(gF(atype(iat),a(iat)%F)).eq.-a(iat)%F) &
              !              a(iat)%F = max(a(iat)%F-1.d0,Fmin(atype(iat)))
              !a(iat)%mF = a(iat)%mF - gF(atype(iat),a(iat)%F)/abs(gF(atype(iat),a(iat)%F))
              indic(iat) = .false.
            end if
          endif
        end do
        !
        t = t + dt
        tmes = tmes + dt
      end do
      goto 9000
777   continue
      !       write(*,*) 'macro=',macro
      !       write(*,*) 'program EV is terminated!'
9000  continue
      close(2)
      close(3)
      close(4)
```

```
   close(5)
  end program ev
!-----------------------------------------------------------------------
     !Derivative subroutine for Runge-Kutta method.
     subroutine derivs(t,y,y_t)
!-----------------------------------------------------------------------
       use parameters
       implicit none
       real*8 y(1:6),y_t(1:6)
       real*8 t,deno,gF
       real*8 tmp, tmp1, tmp2, tmp3, y12, y22, y32, tmp4
       !
       y_t(1)=y(4)
       y_t(2)=y(5)
       y_t(3)=y(6)
       y12  =  y(1)**2
       y22  =  y(2)**2
       y32  =  y(3)**2
       tmp4 = Bprime*Bsec
       tmp1 = uB*gFtemp(iat)*a(iat)%mF/mass(atype(iat))
       tmp2 = Bprime**2
       tmp3 = 0.5d0*B0*Bsec
       tmp  = 0.125d0*Bsec**2
       deno=sqrt((B0+Bsec/2.*y22-Bsec*(y12+y32)/4.0d0)**2 + &
            (Bprime*y(1)-Bsec*y(1)*y(3)/0.5d0)**2 + (Bprime*y(3)+Bsec*y(2)*y(3)/0.5d0)**2)
       y_t(4)=-tmp1*(tmp2*y(1)-tmp4*y(1)*y(2)-tmp3*y(1)+tmp*y(1)*(y12+y32))/deno
       y_t(5)=-tmp1*(-0.5d0*tmp4*y12+B0*Bsec*y(2)+2*tmp*y22*y(2)+0.5d0*y32*tmp4)/deno
       y_t(6)=-tmp1*(tmp2*y(3)+tmp4*y(3)*y(2)-tmp3*y(3)+tmp*y(3)*(y12+y32))/deno
     end subroutine derivs
!-----------------------------------------------------------------------
     !4th order Runge-Kutta method
     subroutine rk4(t,y,h)
!-----------------------------------------------------------------------
       implicit none
       real*8 y(1:6),dydt(1:6),dym(1:6),dyt(1:6),yt(1:6)
       real*8 t,h,hh,h6,th
       external derivs
       integer i
       !
       hh=h*0.5d0
       h6=h/6.d0
       th=t+hh
       !
       call derivs(t,y,dydt)
       do i=1,6
          yt(i)=y(i)+hh*dydt(i)
       enddo
       !
       call derivs(th,yt,dyt)
       do i=1,6
          yt(i)=y(i)+hh*dyt(i)
       enddo
       call derivs(th,yt,dym)
       do i=1,6
          yt(i)=y(i)+h*dym(i)
          dym(i)=dyt(i)+dym(i)
       enddo
       call derivs(t+h,yt,dyt)
       do i=1,6
          y(i)=y(i)+h6*(dydt(i)+dyt(i)+2.d0*dym(i))
       enddo
     end subroutine rk4
!-----------------------------------------------------------------------
     !function that produces Random numbers with equal probability.
     FUNCTION ran3(idum)
!-----------------------------------------------------------------------
       INTEGER idum
       REAL ran3
       integer, parameter:: MBIG=1000000000
```

```
      integer, parameter:: MSEED=161803398
      integer, parameter:: MZ=0
      real,    parameter:: FAC=1./MBIG
      INTEGER i,iff,ii,inext,inextp,k
      INTEGER mj,mk,ma(55)
      SAVE iff,inext,inextp,ma
      DATA iff /0/
      if(idum.lt.0.or.iff.eq.0)then
         iff=1
         mj=MSEED-iabs(idum)
         mj=mod(mj,MBIG)
         ma(55)=mj
         mk=1
         do i=1,54
            ii=mod(21*i,55)
            ma(ii)=mk
            mk=mj-mk
            if(mk.lt.MZ)mk=mk+MBIG
            mj=ma(ii)
         end do
         do k=1,4
            do i=1,55
               ma(i)=ma(i)-ma(1+mod(i+30,55))
               if(ma(i).lt.MZ)ma(i)=ma(i)+MBIG
            end do
         end do
         inext=0
         inextp=31
         idum=1
      endif
      inext=inext+1
      if(inext.eq.56)inext=1
      inextp=inextp+1
      if(inextp.eq.56)inextp=1
      mj=ma(inext)-ma(inextp)
      if(mj.lt.MZ)mj=mj+MBIG
      ma(inext)=mj
      ran3=mj*FAC
      return
    END FUNCTION ran3
!-------------------------------------------------------------------------------
    !Finding inelastic collisional cross section between different spin states
    real*8 Function G_inelastic(i1,i2)
      use parameters
      implicit none
      integer m , n, i1, i2,isign
      real*8 gF
      isign = gF(atype(i1),a(i1)%F)/abs(gF(atype(i1),a(i1)%F))
      m=(Fmax(atype(i1))-a(i1)%F)*(Fmax(atype(i1))+a(i1)%F+2)+a(i1)%F-isign*a(i1)%mF+1
      if(atype(i1) .eq. 2) m=m+(Fmax(1)-Fmin(1)+1)*(2*Fmax(1)+2*Fmin(1)+2)/2.d0
      isign = gF(atype(i2),a(i2)%F)/abs(gF(atype(i2),a(i2)%F))
      n=(Fmax(atype(i2))-a(i2)%F)*(Fmax(atype(i2))+a(i2)%F+2)+a(i2)%F-isign*a(i2)%mF+1
      if(atype(i2) .eq. 2) m=m+(Fmax(1)-Fmin(1)+1)*(2*Fmax(1)+2*Fmin(1)+2)/2.d0
      G_inelastic = sigma_in(m,n)
      return
    end Function G_inelastic
!-------------------------------------------------------------------------------
    !finding elastic scattering length between different spin states
    real*8 Function G_elastic(i1,i2)
      use parameters
      implicit none
      integer m , n, i1, i2,isign
      real*8 gF
      isign = gF(atype(i1),a(i1)%F)/abs(gF(atype(i1),a(i1)%F))
      m=(Fmax(atype(i1))-a(i1)%F)*(Fmax(atype(i1))+a(i1)%F+2)+a(i1)%F-isign*a(i1)%mF+1
      if(atype(i1) .eq. 2) m=m+(Fmax(1)-Fmin(1)+1)*(2*Fmax(1)+2*Fmin(1)+2)/2.d0
      isign = gF(atype(i2),a(i2)%F)/abs(gF(atype(i2),a(i2)%F))
      n=(Fmax(atype(i2))-a(i2)%F)*(Fmax(atype(i2))+a(i2)%F+2)+a(i2)%F-isign*a(i2)%mF+1
      if(atype(i2) .eq. 2) m=m+(Fmax(1)-Fmin(1)+1)*(2*Fmax(1)+2*Fmin(1)+2)/2.d0
```

```
      G_elastic = sl(m,n) ! sl(:,:) is the scattering length matrix
      return
    end Function G_elastic
!-------------------------------------------------------------------------------
```