N-Body Simulation of High-Density Magneto-Optical Traps (MOT)

Documentation & Algorithm Analysis

November 26, 2025

Abstract

This document provides a comprehensive description of the physics, numerical methods, and usage of the `mot_simulation.py` suite. The simulation models the dynamics of large ^{87}Rb atom clouds ($N \sim 10^7 - 10^9$) in the high-density regime. It specifically addresses the transition from the temperature-limited regime to the density-limited regime dominated by collective effects: **Shadowing** (attenuation) and **Radiative Rescattering**. To make these $O(N^2)$ interactions computationally feasible, the code employs a **Superparticle** approximation combined with **Particle-Mesh (FFT)** algorithms.

Contents

1 Physics of the Magneto-Optical Trap

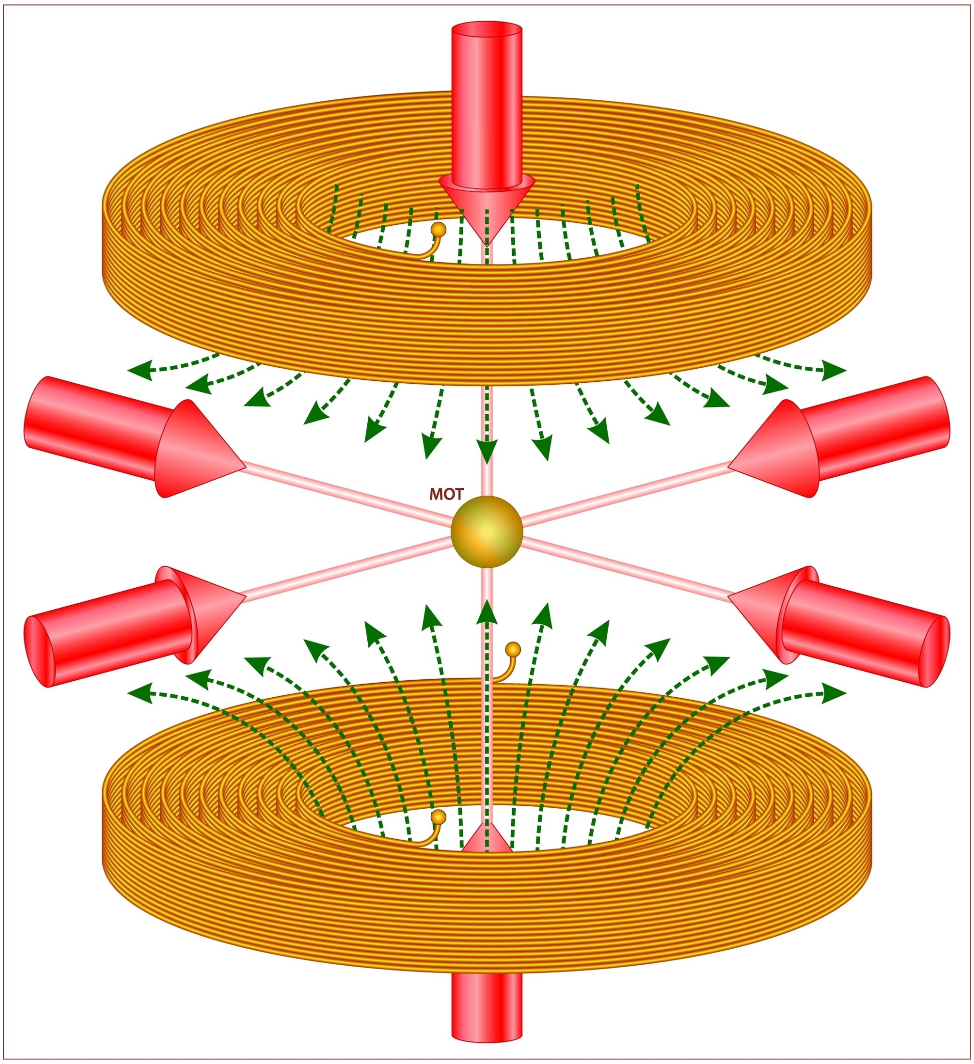

Figure 1: Visualization of a magneto-optical trap setup.

The Magneto-Optical Trap (MOT) relies on the interplay between dissipative optical forces (Doppler cooling) and position-dependent restoring forces arising from an inhomogeneous magnetic field (Zeeman effect).

1.1 Doppler Cooling (The Friction Force)

The simulation assumes a simplified two-level atomic structure (ground state $J = 0$ to excited state $J' = 1$). The atoms are illuminated by six counter-propagating laser beams along the Cartesian axes. The laser frequency ω_L is detuned below the atomic resonance ω_0 (red detuning, $\delta = \omega_L - \omega_0 < 0$).

For an atom moving with velocity $\vec{v}$, the laser frequency seen by the atom is shifted by the Doppler effect: $\omega' = \omega_L - \vec{k} \cdot \vec{v}$.

- If the atom moves *towards* a laser beam, the light is Doppler-shifted closer to resonance, increasing the absorption rate.
- If the atom moves *away*, the light is shifted further from resonance, decreasing absorption.

The net result is a radiation pressure force that opposes velocity, acting as a viscous drag: $\vec{F}_{cool} \approx -\beta\vec{v}$.

1.2 Spatial Confinement (The Restoring Force)

To trap atoms, the force must be position-dependent. This is achieved using a quadrupole magnetic field generated by anti-Helmholtz coils:

$$\vec{B}(x, y, z) = B'(x\hat{x} + y\hat{y} - 2z\hat{z}) \tag{1}$$

where B' is the magnetic field gradient.

This field causes a spatial variation in the atomic energy levels due to the **Zeeman Effect**. The detuning becomes position-dependent:

$$\delta_{\text{eff}}(\vec{r}) = \delta - \vec{k} \cdot \vec{v} - \frac{\mu_B g_F}{\hbar}|\vec{B}(\vec{r})| \tag{2}$$

By choosing correct circular polarizations (σ^+, σ^-) for the laser beams, atoms further from the center are brought closer to resonance with the beam pushing them back towards the center. This creates a harmonic restoring force: $\vec{F}_{trap} \approx -\kappa\vec{r}$.

2 Collective Effects in High-Density Regimes

Standard MOT theory assumes atoms are independent. However, when the number of atoms N exceeds $\sim 10^6$ or density n exceeds 10^{10} cm^{-3}, the cloud becomes optically thick. Two competing collective effects emerge.

2.1 The Shadow Effect (Attenuation)

As laser beams propagate through the cloud, photons are absorbed by the outer layers of atoms. This creates a ”shadow” in the center of the trap.

- **Mechanism:** The intensity of each beam I_α decays according to the Beer-Lambert law:
$$\frac{dI_\alpha}{dz} = -n(\vec{r})\sigma_{sc}I_\alpha \tag{3}$$
- **Consequence:** The restoring force from the ”far side” beam is weaker than the ”near side” beam. This imbalance creates a net compressive force that squeezes the cloud, potentially leading to instabilities.

2.2 Radiative Rescattering (The Repulsive Force)

The most critical effect for the density limit is **Rescattering**. Photons absorbed by one atom are not destroyed; they are re-emitted via spontaneous emission. Ideally, these photons escape the cloud. In a dense cloud, however, a re-emitted photon has a high probability of being re-absorbed by a neighboring atom.

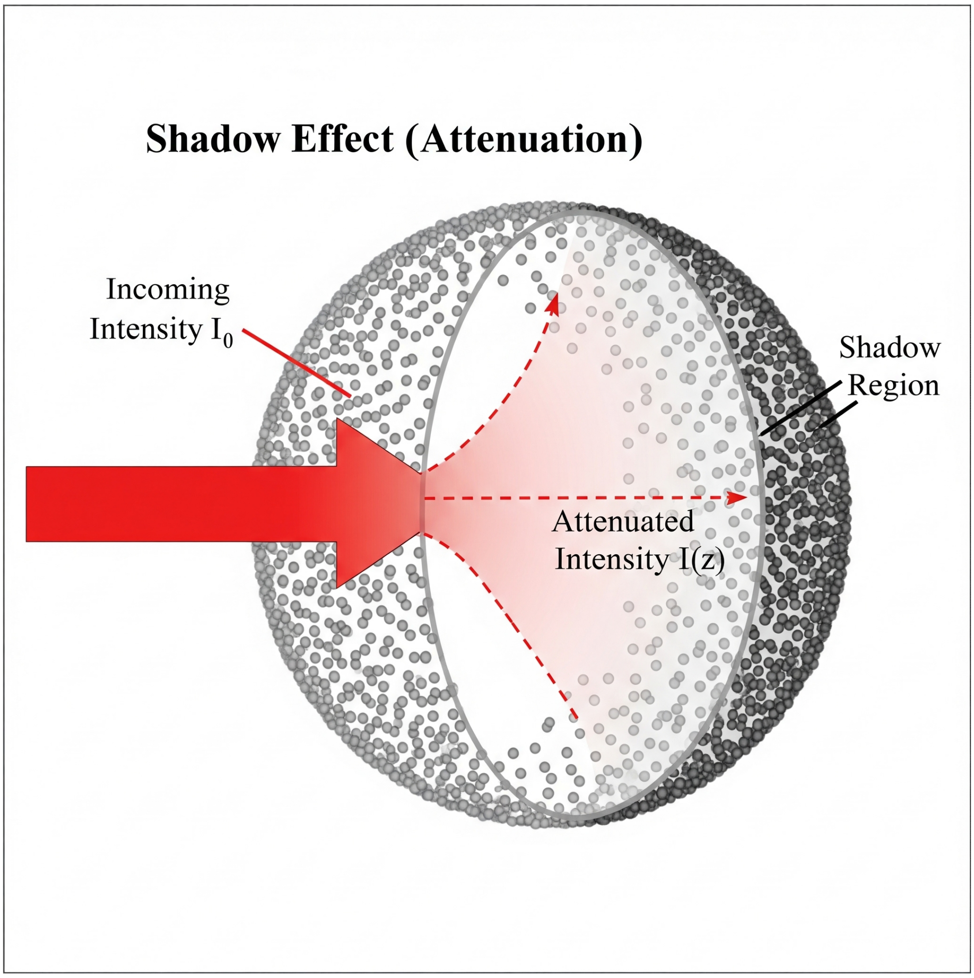

Figure 2: Visualization of shadowing.

2.2.1 Derivation of the Force

Consider an atom at position $\vec{r}_j$ with a total scattering rate Γ_{sc}. This atom acts as a point source of radiation, re-emitting power P_j:

$$P_j = \hbar\omega_L \times \Gamma_{sc}(\vec{r}_j) \tag{4}$$

This power propagates outwards as a spherical wave. The intensity I_{resc} at a distance r from the source atom is given by the inverse-square law:

$$I_{resc}(r) = \frac{P_j}{4\pi r^2} \tag{5}$$

A second atom at position $\vec{r}_i$ will absorb this scattered radiation. The force exerted on atom i is the momentum transfer rate from this rescattered field:

$$\vec{F}_{ij} = \sigma_R \frac{I_{resc}(|\vec{r}_i - \vec{r}_j|)}{c} \hat{n}_{ij} \tag{6}$$

where σ_R is the *rescattering cross-section* (which may differ from the laser cross-section due to spectral broadening) and c is the speed of light.

Combining these terms yields the full pairwise force equation:

$$\vec{F}_{ij} = \frac{\sigma_R P_j}{4\pi c} \frac{\vec{r}_i - \vec{r}_j}{|\vec{r}_i - \vec{r}_j|^3} \tag{7}$$

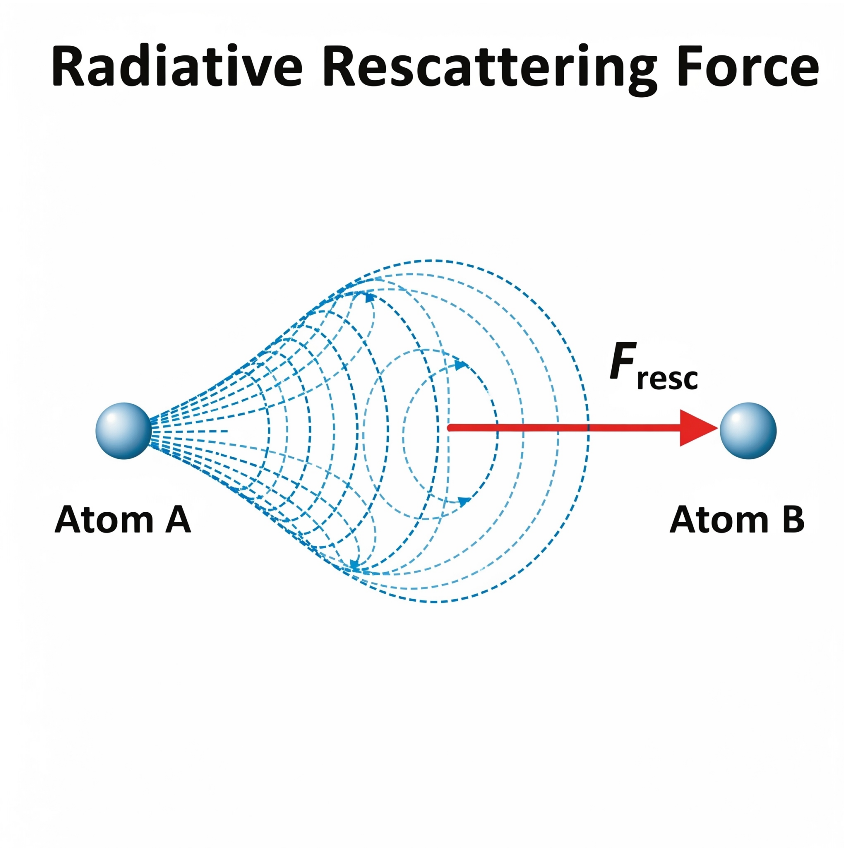

Figure 3: Visualization of re-scattering force.

2.2.2 Electrostatic Analogy

This equation is mathematically isomorphic to **Coulomb's Law** for electrostatics:

$$\vec{F}_{el} = \frac{1}{4\pi\epsilon_0}\frac{q_i q_j}{r^2}\hat{r} \tag{8}$$

By comparing the constants, we can define an "effective charge" for the atoms. If we assume the rescattering cross-section σ_R is uniform, the effective charge q_{eff} of an atom is proportional to the square root of its scattered power:

$$q_{\text{eff}} \sim \sqrt{\frac{\sigma_R P}{c}} \tag{9}$$

Consequently, a MOT in the multiple-scattering regime behaves like a uniformly charged gas of ions (a non-neutral plasma), subject to an external trapping potential.

2.2.3 The Density Limit

This repulsive force scales as $1/r^2$, effectively creating an outward "radiation pressure." The MOT cloud stops shrinking when this outward pressure balances the inward magneto-optical trapping pressure.

- **Low N:** Size is constant (Temperature limited). Density increases with N.
- **High N:** Density is constant (Density limited). The cloud size R expands as $N^{1/3}$ to maintain constant density.

3 Numerical Algorithms

3.1 1. Time Integration: Velocity Verlet

Complexity: $O(N)$
The simulation integrates the equations of motion using the Velocity Verlet algorithm, a symplectic integrator known for stability in conservative systems.

3.2 2. Beam Attenuation: Vectorized Sorting (Grid/Tube)

Complexity: $O(N \log N)$
Calculating the shadow for every atom is naively $O(N^2)$. We approximate the cloud as a bundle of parallel 1D tubes aligned with the laser axes.

1. **Binning:** Atoms are binned into transverse grid cells (u, v).
2. **Sorting:** Within each tube, atoms are sorted by their longitudinal position (w).
3. **CumSum:** The optical depth is computed via a cumulative sum along the sorted array, reducing the complexity to that of the sorting algorithm.

3.3 3. Rescattering: Particle-Mesh (FFT)

Complexity: $O(N + M \log M)$
Directly summing the pairwise Coulomb-like rescattering force is $O(N^2)$. For $N = 10^5$, this requires 10^{10} operations per step, which is computationally prohibitive. We exploit the electrostatic analogy to solve the problem using a grid-based field solver (Particle-Mesh method).

3.3.1 Theoretical Basis: The Poisson Equation

Since the force follows an inverse-square law, the divergence of the force field is proportional to the source density (Gauss's Law). We can describe the rescattering field via a scalar potential Φ:

$$\vec{F}_{\text{resc}}(\vec{r}) = -\sigma_R \nabla \Phi(\vec{r}) \tag{10}$$

The potential Φ satisfies the **Poisson Equation**:

$$\nabla^2 \Phi(\vec{r}) = -\frac{S(\vec{r})}{c} \tag{11}$$

where $S(\vec{r})$ is the continuous density of scattered power (Watts per m^3).

3.3.2 Algorithm Implementation Steps

The Particle-Mesh (PM) algorithm solves this differential equation in four steps:

Step 1: Charge Assignment (Binning)
The continuous atom positions are discretized onto a regular 3D grid of size $M \times M \times M$. The scattered power P_j of each superparticle is distributed to the nearest grid nodes using a weighting scheme (e.g., Nearest Grid Point or Cloud-in-Cell). This yields the source term grid S_{ijk}.

Step 2: Poisson Solver in Fourier Space
The Poisson equation is a convolution, which becomes a simple multiplication in Fourier space. We apply the Fast Fourier Transform (FFT) to the source grid:

$$\hat{S}(\vec{k}) = \text{FFT}(S_{ijk}) \tag{12}$$

In Fourier space, the Laplacian operator ∇^2 becomes $-k^2$. The potential is found by multiplying with the Green's function:

$$\hat{\Phi}(\vec{k}) = \frac{\hat{S}(\vec{k})}{k^2} \quad \text{(with } \hat{\Phi}(0) = 0\text{)} \tag{13}$$

Step 3: Field Calculation
The force is the gradient of the potential. In Fourier space, the gradient operator ∇ corresponds to multiplication by $i\vec{k}$. We compute the three components of the force field vector $\vec{E}$ directly in k-space:

$$\hat{\vec{E}}(\vec{k}) = -i\vec{k}\,\hat{\Phi}(\vec{k}) \tag{14}$$

We then apply the Inverse FFT (IFFT) to transform these field components back to real space, obtaining the force field on the grid $\vec{F}_{grid}$.

Step 4: Force Interpolation
The force acting on a specific atom at position $\vec{r}_i$ is computed by interpolating the values from the surrounding grid nodes in $\vec{F}_{grid}$ (e.g., Trilinear Interpolation).

3.3.3 Complexity Analysis

The complexity is dominated by the FFT step, which scales as $O(M^3 \log M)$, where M is the grid resolution. Crucially, this is independent of the number of atoms N (once N is large enough).

- **Pairwise Summation:** $O(N^2)$
- **Particle-Mesh:** $O(N) + O(M^3 \log M)$

For a typical simulation with $N = 10^5$ atoms and a 32^3 grid, the Particle-Mesh method is several orders of magnitude faster.

4 Usage Guide

4.1 Running the Simulation

```
python mot_simulation.py [N_atoms] [B_gradient] [T_init_uK] [R_init_m]
```

Example: `python mot_simulation.py 10000 0.1 1000 5e-4`

4.2 Post-Processing

Use the included `mot_postprocessing.py` script to generate graphs and animations.

```
python mot_postprocessing.py [N_atoms] [B_gradient] [T_init_uK] [R_init_m]
```

It generates:

- **mot_evolution.gif:** Animation of cloud dynamics and temperature.
- **Virial.png:** Stability analysis comparing Kinetic Energy vs. Trapping Potential.